МАЗМҰНЫ

БІРІНШІ БАСЫЛЫМҒА АЛҒЫ СӨЗ

Қоғамды ақпараттандыру заманында Информатика - Computer Sc*i*ence (Есептеу ғылымы) - ең қажетті және ең маңызды ғылым болды. Онымен адам қоғамының тек материалды өндірісіне ғана емес, интеллектуалды және рухани тірлігіне де әсер етіп оларды түбегейлі түрлендіретін зат пен энергияны меңгерген революциядан кейінгі ақпаратты жинақтау, өңдеу және жіберу саласындағы революцияны байланыстырады.

Информатика адамның интеллектуалды өміріндегі әртүрлі есептерді шығару және жеке үдерістерді автоматтандыру үшін жасанды тілдерде жазылған программалар арқылы компьютерде жүзеге асырылатын ақпараттық технологияларды жасау мен зерттеу проблемаларымен айналысады.

Информатика *Техникалық қамтым – Hardware* (*Қатты жабдықтар*), *Программалық қамтым – Software* (*Жұмсақ жабдықтар*) және *Интеллектуалдық қамтым – Brainware* (*Ақылды жабдықтар*) деген бөлімдерден тұрады. Бұл бөліктер теңмәнді және адамзат қоғамының дамуының заманауи талаптарына ыңғайланып оңалуға қабілетті.

Заманауи ақпараттық технологияларда лингвистикалық (морфологиялық, синтаксистік, семантикалық және сөйлеу) анализ бен синтез әдістері маңызды рөл атқарады. Әртүрлі тілдік процессорлар (трансляторлар, компиляторлар, интерпретаторлар, конверторлар, редакторлар және т.б.) лингвистикалық әдістерді қолдануға негізделген. Тілдер мен автоматтар теориясы информатика ғылымының іргелі саласы бола тұра, осы әдістердің ғылыми негізін құрады және тілдерді тудыру мен тану механизмдерін зерттеумен шұғылданады.

Тудыру механизмдер (формалды грамматикалар) теориясының негізін *XX*-ғасырдың 40–50 жылдары табиғи тілдерге арналған лингвистикалық жұмыстарға байланысты Н.Хомский (N.Chomsky) қалады. Ал танушы механизмдер (ақырлы автоматтар) теориясының негізін сол жылдары М.О.Рабин (***M***.O.Rabin), Д.Скотт (D.Scott) және

В.М.Глушков жасады. Сол кезде тілдер мен автоматтар теориясы программалау тілдерін жасау және жүзеге асыру саласында кең практикалық қолданыс тапты.

Қазір лингвистикалық әдістерге негізделген технологиялық жабдықтардың қолдануы қомақты артты. Олар табиғи тілдерді өңдеу жүйелерін, соның ішінде орфографиялық түзеткіштерді, морфологиялық және синтаксистік талдау, мағыналы іздеу мен шешім қабылдау жүйелерін, сонымен қатар, сөйлеу технологияларын жасау кезінде қолданылады. Дегенмен оларды сауатты және тиімді қолдану пайдаланушыдан, ең кемінде, олар негізделетін математикалық теорияны білуді талап етеді.

Ұсынып отырған оқулық әр жылдары Информатика, Ақпараттық жүйелер, Есептеу техникасы және программалық қамтым мамандықтары бойынша Л.Н.Гумилев атындағы Еуразия ұлттық университетінің ақпараттық технологиялар факультетінің студенттері мен магистранттарына оқылған дәрістер негізінде жазылды.

Оқулықты дайындауға мазмұны информатика мамандығының жалпы білім беру мемлекеттік стандартына сәйкес болатындай оқу басылымдарының жоқтығы себеп болды. Сондықтан оқу процесін толыққанды қамту үшін авторға, ішінде бір ғана ұғым әртүрлі терминдер, анықтамалар және белгілеулерге ие болатын, түрлі әдебиеттер мен интернет ресурстарын пайдалануға тура келді. Бұл жағдай бір ұғымға тек бір ғана термин, анықтама және белгі пайдаланатын осы оқулықты жазу идеясына әкеліп соқтырды. Автор осы ұсынылған оқулықтың оқылатын пән бойынша оқу басылымдарының жетіспеушілігін жөндеуге және бірдей ұғымдардың терминдерін, анықталуларын және белгілеулерін бірыңғай етуге мүмкіндік береріне үміт артады.

Оқу материалын беру методикасы оқырманды тілдер мен автоматтардың классикалық теориясының іргелі фактілері туралы айтуға және оқылатын саладағы тұжырымдарды дәлелдеу әдістерімен таныстыруға, сонымен қатар, оны қандайда бір мысалдармен жабдықтауға негізделген. Теоремалар дәлелдеулерінің барлығы жуық конструктивті сипат алады, сондықтан олар практикаға пайдалы материал береді.

Оқулық орыс және қазақ тілдерінде бірдей мазмұнды жазылды және ол үш деңгейлі бес бөлімнен және әдебиеттер тізімінен тұрады.

Бірінші бөлімде оқулықта қолданылатын белгілеулер, ұғымдар, түсініктер, қысқартулар, математикалық негіздер және тілдерді анықтау механизмдері талқыланады.

Екінші бөлімде регулярлық тілдерді тудырушы механизмдер (регулярлық жиындар, регулярлық өрнектер, регулярлық алгебра, регулярлық теңдеулер мен олардың жүйесі, оң сызықты грамматика), регулярлық тілдерді танушы механизмдер (бейдетерминді және детерминді ақырлы автоматтар және олардың эквиваленттігі), регулярлық тілдердің қасиеттері (регулярлық өрнектер, оң сызықты грамматикалар және ақырлы автоматтардың эквиваленттілігі) және олардың алгоритмдік проблемалары қарастыралыды.

Үшінші бөлімде контекстісіз тілдерді тудырушы механизмдер (контексті-бос грамматикалар), контекстісіз тілдерді танушы механизмдер (бейдетерминді мен детерминді стекті автоматтар және олардың эквиваленттілігі), контекстісіз тілдердің қасиеттері (контексті-бос грамматикалармен стекті автоматтардың эквиваленттілігі) және олардың алгоритмдік проблемалары беріледі.

Төртінші бөлімде контексті тілдерді тудырушы механизмдер (контексті-тәуелді грамматикалар), контексті тілдерді танушы механизмдер (бейдетерминді мен детерминді сызықты шенеуленген автоматтар және олардың эквиваленттілігі), контексті тілдердің қасиеттері (контексті-тәуелді грамматикалар және сызықты шенеуленген автоматтардың эквиваленттілігі) және олардың алгоритмдік проблемалары жазылады.

Бесінші бөлімде шенеуленбеген тілдерді тудырушы механизмдер (шенеуленбеген грамматикалар), шенеуленбеген тілдерді танушы механизмдер (бейдетерминді және детерминді Тьюринг машиналары және олардың эквиваленттілігі), шенеуленбеген тілдердің қасиеттері (шенеуленбеген грамматикалар және сызықты шенеуленген автоматтардың эквиваленттілігі) және олардың алгоритмдік проблемалары ұсынылады.

Әдебиеттер тізімінде оқулықтың оқыту материалдарын құрған кезде пайдаланылған жалпы әдебиеттік көздер көрсетілген. Олардың ішінде кең таралған монографиялар, оқулықтар, оқу құралдары және басқалар бар. Сондықтан осы оқулық мәтінінде оларға тікелей сілтеме жасалынбаған.

Оқулық студенттерге, магистранттарға, докторанттарға, оқытушыларға, ғалымдарға және тілдер мен автоматтар теориясын өздігінен оқып білем деген барлық азаматтарға арналған.

Оқулыққа байланысты барлық ескертпелер мен ұсыныстарды мына e-mail әдіріске жіберуге болады: sharalt@ma*i*l.ru.

Автор:
ШӘРІПБАЙ Алтынбек Әмірұлы
техника ғылымдарының докторы, профессор,
Қазақстан Республикасының мемлекеттік сыйлығының лауреаты

ЕКІШІ БАСЫЛЫМҒА АЛҒЫ СӨЗ

Осы оқулықты қайта басуға оның көкейкестігі негіз болды, себебі оған сұраныс басылмақ түгіл, өсуде. Оқырмандардың бұл оқулықты кітапханаларда таба алатына қарамастан, олар оны дүкендерден сатып ала алмайды, себебі оның 1-ші басылымы аз болды. Бұл жағдай авторды осы оқулықты қайта басуға ой салды.

Осы басылымда менің алғашқы ойым жоғарыда көрсетілген 1-ші басылымның алғы сөзін өзгертпеу болды. Бірақ соған қарамастан, мен оқулықта заманауи тілдік процессорларды (конвертерлер, верификаторлар, редакторлар, трансляторлар, компиляторлар, интерпретаторлар және басқаларды) жасау үшін қажет тілдер мен автоматтардың іргелі және қолданбалы проблемалары қарастырылады дегенді қостым.

Әрбір параграф алдында оқу материалын құрастыруда пайдаланылған әдеби түпнұсқалар мен интернет ресурстарына сілтеме берілген.

Оқулық ең алдымен "Ақпараттық және коммуникациялық технологиялары" (АКТ) бағытындағы әртүрлі мамандықтардың студенттеріне, магистранттарына және докоранттарына ұсынылады. Сонымен қатар, ол АКТ-мамандарына (аналитиктер мен программалаушыларға), оқытушыларға және оқымыстыларға пайдалы бола алады. Соңында, оқулық формалды тілдер және автоматтармен жұмыс жасайтын және заманауи тілдік процессорларды жасау технологиялары туралы көбірек білгісі келетін барлық адамдарды қызықтыруы мүмкін.

Автор осы оқулықтың 2-ші басылымының алғашқы үлгісін елеулі жақсартуға бағалы ескертпелер мен ұсыныстар берген пікір берушілер Ә.Ә. Адамовқа, Ғ.З.Қазиевке, У.А. Тукеевке және оқулықты техникалық редакциялаған З.Қадеркееваға, А.Сағадиеваға өз алғысын білдіреді

Автор:

ШӘРІПБАЙ Алтынбек Әмірұлы

I. ТІЛДЕР МЕН АВТОМАТТАР НЕГІЗДЕРІ

I.1. Белгілеулер, ұғымдар және қысқартулар

Бұл тарауда пайдаланатын белгілеулер, ұғымдар және қысқартулар қарастырылады.

I.1.1. Белгілеулер

Бұл параграфта пайдаланатын белгілеулер, олардың атаулары және қолданылымы беріледі, олар I.1.1-кестеде көрсетілген.

I.1.1-кесте. Белгілеулер.

	Белгілеулер	**Атаулары және қолданылымы**
1.	$=$	Теңдік: сол жақтағы шаманың оң жақтағы шамаға тең екендігін көрсетеді
2.	$\neq$	Теңсіздік: сол жақтағы шаманың оң жақтағы шамаға тең емес екендігін көрсетеді
3.	$\in$	Тиістілік: сол жақтағы шаманың оң жақтағы шамаға тиісті екендігін көрсетеді
4.	$\notin$	Тиісті еместік: сол жақтағы шаманың оң жақтағы шамаға тиісті емес екендігін көрсетеді
5.	$\varnothing$	Бос жиын: жиында бір де бір элемент болмайтындығын көрсетеді
6.	ε	Бос тізбе: тізбеде бір де бір символ болмайтындығын көрсетеді
7.	$\{\varepsilon\}$	Бос тізбелер жиыны: жиынның элементі бір ғана бос тізбе болатындығын көрсетеді
8.	⌴	Кетік: тізбелер арасын ажыратуға қолданылатын және кодтау кестесінде коды бар, бірақ экран мен баспаға шыққанда көрінбейтін таңба
9.	$\subseteq$	Кіру: сол жақтағы шаманың оң жақтағы шамаға кіретіндігін көрсетеді
10.	$\supseteq$	Қамту: сол жақтағы шаманың оң жақтағы шаманы қамтитындығын көрсетеді

11.	⟃	Префикс: сол жақтағы тізбе оң жақтағы тізбенің алдыңғы іштізбесі болатындығын белгілейді
12.	⟄	Постфикс: оң жақтағы тізбе сол жақтағы тізбенің соңғы іштізбесі болатындығын белгілейді
13.	***U***	Универсум: белгілі типтегі барлық шамаларды қамтитын шама
14.	∩	Қиылысу амалы: сол жақтағы шамамен оң жақтағы шаманың қиылысуын көрсетеді
15.	∪	Бірігу амалы: сол жақтағы шамамен оң жақтағы шаманың бірігуін көрсетеді
16.	\	Айырым амалы: сол жақтағы шамадан оң жақтағы шаманы айырғанды (алғанды) көрсетеді
17.	×	Декарттық көбейтінді: сол жақтағы шамамен оң жақтағы шаманың тура көбейтіндісін көрсетеді
18.	·	Тіркесу - конкатенация амалы: екі немесе одан шамалардың тіркесін көрсетеді
19.	∨	Таңдау - дизъюнкция амалы: екі немесе одан көп шамалардың баламасын көрсетеді
20.	*	Итерация амалы: шаманың өзіне-өзі еселі тіркесулерінің бірігулерін көрсетеді
21.	+	Оң итерация амалы: бос емес шамалардың өзіне-өзі еселі тіркесулерінің бірігулерін көрсетеді.
22.	< , >	Бұрыштама жақша: ұғымдардың атаулары жақшаның ішінде екендігін көрсетеді
23.	{,}	Ирек жақша: міндетті элементтері жақшаның ішінде көрсетеді
24.	**[,]**	Тік жақша: жақша ішінде міндетті емес элементтерді көрсетеді
25.	→	Ереже айырымы: сол жағында анықталатын элемент, ал оң жағында анықтағыш элементтер орналасатындығын көрсетеді
26.	⇒	Шығарымдылық қатынасы: грамматика ережелерін қолдануымен алынатын сол жақтағы

		тізбемен оң жақтағы тізбе арасындағы қатынасты көрсетеді
27.	$\perp$	Стек түбін көрсеткіш: стекте ештеме жоқ екендігін көрсетеді
28.	$\vdash,\dashv$	Басы мен аяғын көрсеткіштер: таспаның сол жақ шетін және оң жақ шетін көрсетеді
29.	$\models$	Автоматтың кескіндері арасындағы қатынас
30.	$\boldsymbol{L}(\boldsymbol{G})$	$\boldsymbol{G}$ грамматикасымен туындайтын тіл
31.	$\boldsymbol{L}(\boldsymbol{M})$	$\boldsymbol{M}$ автоматымен танылған тіл
32.	$\rightleftharpoons$	Анықтама бойынша: сол жағындағы объект анықтама бойынша оң жағындағы өрнекке тең

I.1.1-мысалдар:

1. $\boldsymbol{A}=\boldsymbol{B}$ жазуы $\boldsymbol{A}$ шамасымен $\boldsymbol{B}$ шамасы тең екендігін білдіреді.
2. $a\in\boldsymbol{A}$ жазуы a шамасы $\boldsymbol{A}$ шамасына тиісті екендігін білдіреді.
3. $\boldsymbol{A}\subseteq\boldsymbol{B}$ жазуы $\boldsymbol{A}$ шамасы $\boldsymbol{B}$ шамасына кіретіндігін білдіреді.
4. $\boldsymbol{A}\cup\boldsymbol{B}$ жазуы $\boldsymbol{A}$ шамасымен $\boldsymbol{B}$ шамасы бірігуін білдіреді.
5. $\boldsymbol{A}\cdot\boldsymbol{B}$ жазуы $\boldsymbol{A}$ шамасымен $\boldsymbol{B}$ шамасы тіркесуін білдіреді.

I.1.1-тапсырмалар:

1. $\boldsymbol{A}$ шамасымен $\boldsymbol{B}$ шамасының теңсіздігін жазыңыз.
2. a шамасы $\boldsymbol{A}$ шамасына тиісті емес екендігін жазыңыз.
3. $\boldsymbol{A}$ шамасы $\boldsymbol{B}$ шамасын қамтитындығы жазыңыз
4. $\boldsymbol{A}$ шамасымен $\boldsymbol{B}$ шамасының қиылысуын жазыңыз.
5. $\boldsymbol{A}$ шамасымен $\boldsymbol{B}$ шамасының тік көбейтіндісін жазыңыз.
6. $\boldsymbol{A}$ шамасынының $\boldsymbol{B}$ шамасына кіретіндігін жазыңыз.

I.1.1-сұрақтар:

1. Бос жиын қалай белгіленеді?
2. Бос тізбелер жиыны қалай белгіленеді?
3. Міндетті элементтер қалай көрсетіледі?
4. Таспаның басы мен аяғы қалай көрсетіледі?
5. Шығарымдылық қатынасы қалай көрсетіледі?
6. Таңдау амалы қалай көрсетіледі?

I.1.2. Ұғымдар

Бұл параграфта қолданылатын ұғымдар және олардың түсініктемелері қарастырылған, олар I.1.2-кестеде келтірілген.

I.1.2-кесте. Ұғымдар.

№	Ұғымдар	Түсініктемелер
1.	*Абстрактылы автомат*	Тілді танитын немесе түрлендіретін алгоритмдік жүйе
2.	*Әліпби*	Бірнәрсені белгілейтін символдардың бос емес ақырлы жиыны
3.	*Бейтерминал*	Басқа тұрақты немесе айнымалы шамалар арқылы анықталатын айнымалы шама
4.	*Бос емес тізбе*	Бір немесе бірнеше әліпби символынан құралатын тізбе
5.	*Бос жиын*	Құрамында бір де бір элементі жоқ жиын
6.	*Бос тізбе*	Әліпбидің бір де символын қамтымайтын және ұзындығы нөлге тең тізбе
7.	*Кетік*	Тізбелерді бір-бірінен айыруға қолданылатын символ
8.	*Лингвистика*	Тілдердің топтарын және олардың арасындағы қасиеттерін зерттейтін ғылым
9.	*Мәтін*	Символдар тізбесі
10.	*Семантика*	Мәтін мен мағына (мән) арасындағы сәйкестіктің сипаттамалары
11.	*Семиотика*	Мағыналы символов жүйесі жөніндегі ғылым
12.	*Символ*	Тілдің ең кіші бөлінбейтін синтаксистік бірлігі
13.	*Синтаксис*	Тілдің мақсаты мен міндетіне тәуелсіз мағыналы мәтіндерді анықтайтын ережелер
14.	*Таныгыш*	Қайсы бір әліпбиде берілген тілді - тізбелер жиынын танитын абстрактылы автомат
15.	*Терминал*	Өздігінен анықталатын символ (әріп, цифра, арнаулы таңба)
16.	*Тізбе*	Берілген әліпбидегі символдардың ақырлы тізбегі

17.	*Тізбе ұзындығы*	Тізбедегі символдар санын көрсететін функция
18.	*Тіл*	Белгілі құрылымдары және мағыналары бар ақырлы тізбелер жиыны
19.	*Формалды грамматика*	Формалды тілді тударатын математикалық жүйе

I.1.2-мысалдар:

1. Машиналық тілдің әліпбиі 0 мен 1 жиынынан тұрады.
2. Классикалық латын әліпбиі 26 әріптен тұрады.
3. Орыс әліпбиінің барлық әріптері дыбыстарды белгілемейді.
4. Бейтерминалдарға «Сөйлем», «Сөз», «Әріп» ұғымдары жатады.
5. Терминалдарға 0,1,3,4,5,6,7,8,9 жатады.

I.1.2-тапсырмалар:

1. Тізбе ұғымын беріңіз.
2. Лингвистика ұғымын беріңіз.
3. Бейтерминал ұғымын беріңіз.
4. Танығыш ұғымын беріңіз.
5. Формалды грамматика ұғымын беріңіз.
6. Әліпби ұғымын беріңіз.
7. Абстрактылы автомат ұғымын беріңіз.
8. Бос жиын ұғымын беріңіз.
9. Кетік ұғымын беріңіз.
10. Семантика ұғымын беріңіз.

I.1.2-сұрақтар:

1. Тізбе ұзындығы деген не?
2. Бос тізбе деген не?
3. Бос жиын деген не?
4. Кетік деген не?
5. Тіл деген не?
6. Формалды грамматика деген не?
7. Танушы деген не?

I.1.3. Қысқартулар

Бұл параграфта қолданылатын қысқартулар мен олардың ашылымдары қарастырылған, олар I.1.3-кестеде берілген.

I.1.3-кесте. Қысқартулар.

	Қысқартулар	**Ашылымдар**
1.	*АА*	Ақырлы автомат
2.	*БАА*	Бейдетерминді ақырлы автомат
3.	*БСА*	Бейдетерминді стекті автомат
4.	*БСША*	Бейдетерминді сызықты шенеуленген автомат
5.	*БТМ*	Бейдетерминді Тьюринг машинасы
6.	*ДАА*	Детерминді ақырлы автомат
7.	*ДСА*	Детерминді стекті автомат
8.	*ДСША*	Детерминді сызықты шенеуленген автомат
9.	*ДТМ*	Детерминді Тьюринг машинасы
10.	*КБГ*	Контексті-бос грамматика
11.	*КГ*	Контексті грамматика
12.	*КСГ*	Контектісіз грамматика
13.	*КТГ*	Контексті-тәуелді грамматика
14.	*ОСГ*	Оң сызықты грамматика
15.	*РЖ*	Регулярлық жиын
16.	*РӨ*	Регулярлық өрнек
17.	*СА*	Стекті автомат
18.	*ССГ*	Сол сызықты грамматика
19.	*США*	Сызықты шенеуленген автомат
20.	*ТМ*	Тьюринг машинасы
21.	*ШГ*	Шенеуленбеген грамматика

І.1.3-мысалдар:

1. ФГ – формалды грамматика.
2. КЛ – компьютерлік лингвистика.
3. МЛ – математикалық лингвистика.
4. ТАТ – тілдер мен автоматтар теориясы.
5. РА – регулярлық алгебра.
6. КБГ – контексті-бос грамматика.
7. БАА – бейдетерминді ақырлы автомат.
8. ОСГ – оң сызықты грамматика.
9. ШГ – шенеуленбеген грамматика.
10. РӨ – регулярлық өрнек.

І.1.3-тапсырмалар:

1. Бейдетерминді ақырлы автоматты қысқартыңыз.
2. Детерминді Тьюринг машинасын қысқартыңыз.
3. Сол сызықты грамматиканы қысқартыңыз.
4. Сызықты шенеуленген автоматты қысқартыңыз.
5. Шенеуленбеген грамматиканы қысқартыңыз.
6. Стекті автоматты қысқартыңыз.
7. Бейдетерминді Тьюринг машинасын қысқартыңыз.
8. Контекстісіз грамматиканы қысқартыңыз.
9. Сызықты шенеуленген автомат қысқартыңыз.
10. Оң сызықты грамматиканы қысқартыңыз.

І.1.3-сұрақтар:

1. ДСША нені білдіреді?
2. КСГ нені білдіреді?
3. ДАА нені білдіреді?
4. БСША нені білдіреді?
5. КБГ нені білдіреді?
6. РӨ нені білдіреді?
7. БТМ нені білдіреді?
8. Регулярлық алгебра нені білдіреді?
9. Регулярлық жиын нені білдіреді?

I.2. Математикалық негіздер

I.2.1. Жиындар және олардағы амалдар

Бұл параграфта жиын ұғымы қарастырылады, олардағы амалдар және олардың қасиеттері ашылады, мысалдар ұсынылады, тапсырмалар беріледі, сұрақтар қойылады [14,17,22,27,28,33].

I.2.1.1-анықтама. *Жиын* деп қайсы бір типпен (белгімен) ажыралатын элементтер жинағы.

Жиын анықтамасы екі түрде беріледі: барлық элементтерін көрсету немесе барлық элементтерін сипаттау арқылы. Жиын элементтері ирек {және} жақшаның ішінде көрсетіледі.

Жиын индекстері бар немесе жоқ үлкен латын әріптермен, элементтер кіші латын әріптермен немесе араб цифрларымен белгіленеді. Мысалы, натурал сандарының жиыны $\boldsymbol{N}$ әріпімен, оның элементтері 1, 2, 3,… араб цифрларымен белгіленеді.

Мына $a\in\boldsymbol{A}$ ($a\notin\boldsymbol{A}$) жазбасы a элементі $\boldsymbol{A}$ жиынына кіреді (кірмейді) дегенді білдіреді.

I.2.1.1-мысалдар:

1. Барлық теріс емес бүтін сандар натурал сандар жиыны болады.
2. $\boldsymbol{D} = \{0, 1\}$, мұнда $\boldsymbol{D}$ жиынының элементтері тек қана көрсетілген 0, 1 тұрақты шамалары болады.
3. $\boldsymbol{X} = \{x\colon x>0\}$, мұнда $\boldsymbol{X}$ жиынының элементтері тек оң айнымалы шамалар x болады.

Барлық жиындар ішінде екі айырықша жиындар ерекшеленеді:

1. $\varnothing$ – *бос жиын* бір де бір элемент қамтымайды.
2. $\boldsymbol{U}$ – *универсал жиын* (*универсум*) қарастыралатын типтегі (пәндік облыстағы) барлық элементтерді қамтыйды.

Теорияға қатысты универсум деген осы теорияда қарастырылатын барлық объектілерді қамтитын жиын.

Мысалы, универсум болып:

1) сандар теориясында – барлық бүтін сандар жиыны;

2) тілдер теориясында – берілген әліпбидегі барлық сөздер жиыны;

3) геометрияда – n-өлшемді геометриялық кеңістікте барлық нүктелер жиыны.

I.2.1.2-анықтама. $\boldsymbol{A}$ жиынының қуаты оның элементтерінің санына тең болады және ол $|\boldsymbol{A}|$ арқылы белгіленіледі.

I.2.1.2-мысалдар:

1. $|\varnothing| = 0$.
2. Егер $\boldsymbol{A} = \{a, b, \text{c}, d, e\}$, онда $|\boldsymbol{A}| = 5$.

I.2.1.3-анықтама. Мейлі екі $\boldsymbol{A}$ және $\boldsymbol{B}$ жиындары берілсін. Сонда оларда келесі амалдарды анықтауға болады:

(1) $\boldsymbol{A}$ және $\boldsymbol{B}$ жиындарының *бірігуі* $\boldsymbol{A}$ немесе $\boldsymbol{B}$ элементтерінен тұрады: $\boldsymbol{A}\cup\boldsymbol{B} = \{x : x\in\boldsymbol{A} \vee x \in \boldsymbol{B}\}$.

(2) $\boldsymbol{A}$ және $\boldsymbol{B}$ жиындарының *қиылысуы* $\boldsymbol{A}$ және $\boldsymbol{B}$ элементтерінен тұрады: $\boldsymbol{A}\cap\boldsymbol{B} = \{x : x\in\boldsymbol{A}\ \&\ x\in \boldsymbol{B}\}$.

(3) $\boldsymbol{A}$ жиынының *толықтыруы* $\boldsymbol{U}$ универсумының элементтерінен тұрады және $\boldsymbol{A}$ жиынының элементтерін қамтымайды:

$$\overline{\boldsymbol{A}} = \{\ \boldsymbol{x} \mid \quad \boldsymbol{x} \in \boldsymbol{U}\ \ \&\ \ \boldsymbol{x} \notin\ \boldsymbol{A}\ \}.$$

(4) $\boldsymbol{A}$ және $\boldsymbol{B}$ жиындарының *айырмасы* $\boldsymbol{A}$ жиынының элементтерінен тұрады және $\boldsymbol{B}$ жиынының элементтерін қамтымайды:

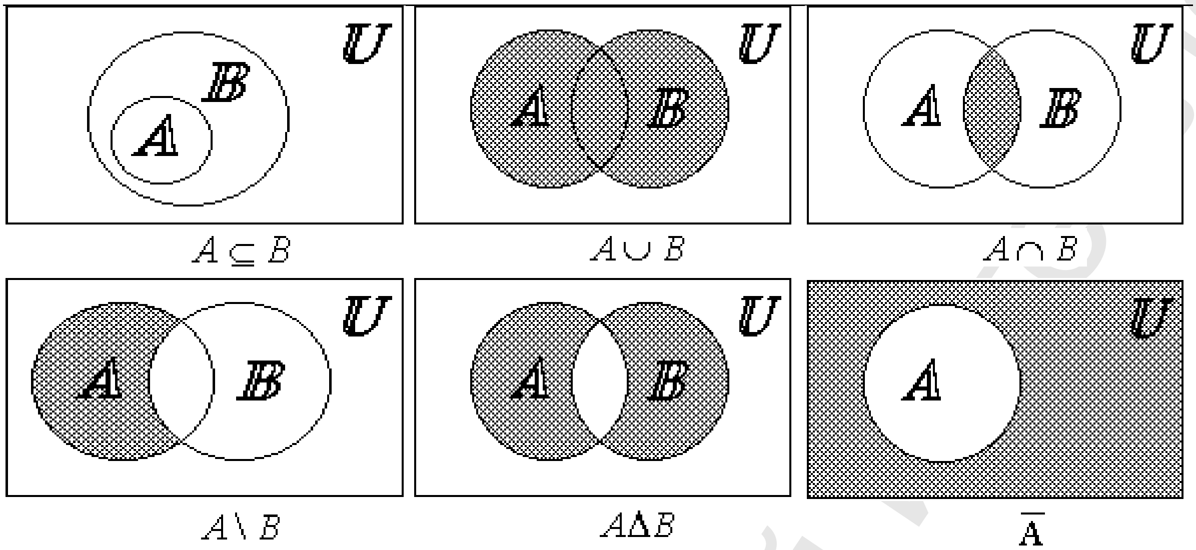
$$\boldsymbol{A} \setminus \boldsymbol{B} = \{x : x\in\boldsymbol{A}\ \&\ x\notin\boldsymbol{B}\}.$$

(5) $\boldsymbol{A}$ және $\boldsymbol{B}$ жиындарының *симметриялық айырмасы* тек қана $\boldsymbol{A}$ жиынының элементтерінен немесе $\boldsymbol{B}$ жиынының элементтерінен тұрады: $\boldsymbol{A}\triangle\boldsymbol{B}=\{x\text{: } (x\in\boldsymbol{A}\&x\notin\boldsymbol{B})\vee(x\in\boldsymbol{B}\&x\notin\boldsymbol{A})\}$.

(6) $\boldsymbol{A}$ және $\boldsymbol{B}$ жиындарының *тік көбейтуі* $\boldsymbol{A}$ және $\boldsymbol{B}$ жиындарының элементтер сыңарынан тұрады: $\boldsymbol{A}\times\boldsymbol{B} = \{(a , b) : a\in\boldsymbol{A}\ \&\ b\in\boldsymbol{B}\}$.

(1) – (3) амалдарын Эйлер-Венн диаграммасы (I.2.1-сурет) арқылы бнйнелеуге болады. Онда универсум $\boldsymbol{U}$ тік төотбұрыш арқылы, ал $\boldsymbol{A}$

және $\boldsymbol{B}$ жиындары шеңберлер арқылы бейнеленген. Нәтижелерді ерекшелеу үшін штрихтеу қолданылады.



І.2.1-сурет. Эйлер-Венн диаграммалары.

Мұнда $\boldsymbol{A}$ мен $\boldsymbol{B}$ жиындары $\boldsymbol{U}$ универсумының ішжиындары болады және ол $\boldsymbol{A} \subseteq \boldsymbol{U}$, $\boldsymbol{B} \subseteq \boldsymbol{U}$ деп жазылады (І.2.2 қараңыз).

(1) – (3) амалдарын тек екі жиында ғана емес, саны n болатын $\boldsymbol{A}_1$, $\boldsymbol{A}_2,\ldots, \boldsymbol{A}_n$ жиындарында да анықтауға болады, мұнда $n \in \boldsymbol{N}$ & $n > 2$.

$\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_n$ жиындарының бірігуі былай анықталады:

$$\boldsymbol{A}_1 \cup \boldsymbol{A}_2 \cup \ldots \cup \boldsymbol{A}_n = \bigcup_{i=1}^{n} \boldsymbol{A}_i .$$

$\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_n$ жиындарының қиылысуы былай анықталады:

$$\boldsymbol{A}_1 \cap \boldsymbol{A}_2 \cap \ldots \cap \boldsymbol{A}_n = \bigcap_{i=1}^{n} \boldsymbol{A}_i .$$

$\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_n$ жиындарының тік көбейтіндісі $(a_1, a_2, \ldots, a_n)$, $a_1 \in \boldsymbol{A}_1, a_2 \in \boldsymbol{A}_2, \ldots, a_n \in \boldsymbol{A}_n$ түрінде болатын кортеждер жиыны ретінде анықталады, яғни

$$\boldsymbol{A}_1 \times \boldsymbol{A}_2 \times \ldots \times \boldsymbol{A}_n = \{(a_1, a_2, \ldots, a_n): a_1 \in \boldsymbol{A}_1 \,\&\, a_2 \in \boldsymbol{A}_2 \,\&\ldots\&\, a_n \in \boldsymbol{A}_n\}$$

Мұнда, егер $\boldsymbol{A}_1 = \boldsymbol{A}_2 = \ldots = \boldsymbol{A}_n = \boldsymbol{A}$, то $\underbrace{\boldsymbol{A} \times \boldsymbol{A} \times \ldots \times \boldsymbol{A}}_{n} = \boldsymbol{A}^n$ – дәреже.

І.2.13-мысалдар: Мейлі $\boldsymbol{A} = \{a, b, c, d, e, f\}$, $\boldsymbol{B} = \{c, d\}$, сонда:

1. $\boldsymbol{A} \cup \boldsymbol{B} = \{a, b, c, d, e, f\}$;
2. $\boldsymbol{A} \cap \boldsymbol{B} = \{c, d\}$;
3. $\boldsymbol{B} \times \boldsymbol{B} = \{(c, c), (c, d), (d, c), (d, d)\}$;
4. $\boldsymbol{A} \setminus \boldsymbol{B} = \{a, b, e, f\}$;

5. $\boldsymbol{A}\triangle\boldsymbol{B} = \{a, b, e, f\}$;

6. $\overline{A}$ универсум $\boldsymbol{U}$ қалай болатындығына тәуелді. Айталық, егер $\boldsymbol{U} = \{a, b, c, d, e, f, h\}$, онда $\overline{A} =\{h\}$.

Енді жиындардың кестелік берілу әдісін және олардағы амалдарды көрсетуге болады. Мейлі $\boldsymbol{U}$, $\boldsymbol{A} \subseteq \boldsymbol{U}$ және $x\in\boldsymbol{U}$ берілсін.

I.2.1.4-анықтама. $\boldsymbol{A}$ жиыны үшін *индикаторлық функция* деп $\boldsymbol{I_A}(x)$ функциясын айтады, ол былай анықталады:

$$\boldsymbol{I}_A(\boldsymbol{x})=\begin{cases}1, & \textbf{\textit{если}} \quad \boldsymbol{x}\in\boldsymbol{A}\\ 0, & \textbf{\textit{если}} \quad \boldsymbol{x}\notin\boldsymbol{A}\end{cases}$$

Сонымен: $\boldsymbol{I}_A\colon\ \boldsymbol{U}\to\{0,1\}$.

$\boldsymbol{A} \subseteq \boldsymbol{U}$ және $\boldsymbol{B} \subseteq \boldsymbol{U}$ үшін мына қасиеттер бар:

$$\boldsymbol{I}_A(\boldsymbol{x})=\boldsymbol{I}_B(\boldsymbol{x})\Leftrightarrow \boldsymbol{A}=\boldsymbol{B};$$

$$\boldsymbol{I}_A(\boldsymbol{x})\le\boldsymbol{I}_B(\boldsymbol{x})\Leftrightarrow \boldsymbol{A}\subseteq\boldsymbol{B};$$

$$\boldsymbol{I}_{\overline{A}}(\boldsymbol{x})=1-\boldsymbol{I}_A(\boldsymbol{x});$$

$$\boldsymbol{I}_{A\cup B}(\boldsymbol{x})=\boldsymbol{I}_A(\boldsymbol{x})+\boldsymbol{I}_B(\boldsymbol{x})-\boldsymbol{I}_A(\boldsymbol{x})\cdot\boldsymbol{I}_B(\boldsymbol{x});$$

$$\boldsymbol{I}_{A\cap B}(\boldsymbol{x})=\boldsymbol{I}_A(\boldsymbol{x})\cdot\boldsymbol{I}_B(\boldsymbol{x});$$

$$\boldsymbol{I}_{A\setminus B}(\boldsymbol{x})=\boldsymbol{I}_A(\boldsymbol{x})-\boldsymbol{I}_A(\boldsymbol{x})\cdot\boldsymbol{I}_B(\boldsymbol{x});$$

$$\boldsymbol{I}_{A\Delta B}(\boldsymbol{x})=\boldsymbol{I}_A(\boldsymbol{x})+\boldsymbol{I}_B(\boldsymbol{x})-2\boldsymbol{I}_A(\boldsymbol{x})\cdot\boldsymbol{I}_B(\boldsymbol{x}).$$

Индикаторлар I.2.1-кесте арқылы берілген.

I.2.1-кесте. Индикаторлар.

$\boldsymbol{x}\in\boldsymbol{A}$	$\boldsymbol{x}\in\boldsymbol{B}$	$\boldsymbol{x}\in\boldsymbol{A}\cup\boldsymbol{B}$	$\boldsymbol{x}\in\boldsymbol{A}\cap\boldsymbol{B}$	$x\in A\setminus B$	$\boldsymbol{x}\notin\boldsymbol{A}$	$\boldsymbol{x}\in\boldsymbol{A}\Delta\boldsymbol{B}$
0	0	0	0	0	1	0
0	1	1	0	0	1	1
1	0	1	0	1	0	1
1	1	1	1	0	0	0

Жиындардағы амалдар келесі қасиеттерге ие болады:

I. Бірігу, қиылысу және айырма:

1) $\boldsymbol{A} \cup \varnothing = \boldsymbol{A}$ – нөл қасиеті;

2) $\boldsymbol{A} \cup \boldsymbol{A} = \boldsymbol{A}$ – идемпотенттік;

3) $\boldsymbol{A} \cup \boldsymbol{B} = \boldsymbol{B}$, егер $\boldsymbol{A}$-ның барлық элементтері $\boldsymbol{B}$-мен қамтылса;

4) $\boldsymbol{A} \cup \boldsymbol{B} = \boldsymbol{B} \cup \boldsymbol{A}$ – коммутативтік;

5) $(\boldsymbol{A} \cup \boldsymbol{B}) \cup \boldsymbol{C} = \boldsymbol{A} \cup (\boldsymbol{B} \cup \boldsymbol{C}) = \boldsymbol{A} \cup \boldsymbol{B} \cup \boldsymbol{C}$ – ассоциативтік;

6) $\boldsymbol{A} \cap \varnothing = \boldsymbol{A}$ – нөлдің қасиеті;

7) $\boldsymbol{A} \cap \boldsymbol{A} = \boldsymbol{A}$ – идемпотенттік;

8) $\boldsymbol{A} \cap \boldsymbol{B} = \boldsymbol{A}$, егер $\boldsymbol{A}$-ның барлық элементтері $\boldsymbol{B}$-мен қамтылса;

9) $\boldsymbol{A} \cap \boldsymbol{B} = \boldsymbol{B} \cap \boldsymbol{A}$– коммутативтік;

10) $(\boldsymbol{A} \cap \boldsymbol{B}) \cap \boldsymbol{C} = \boldsymbol{A} \cap (\boldsymbol{B} \cap \boldsymbol{C}) = \boldsymbol{A} \cap \boldsymbol{B} \cap \boldsymbol{C}$ – ассоциативтік;

11) $\boldsymbol{A} \cup (\boldsymbol{B} \cap \boldsymbol{C}) = (\boldsymbol{A} \cup \boldsymbol{B}) \cap (\boldsymbol{A} \cup \boldsymbol{C})$ – дистрибутивтік;

12) $\boldsymbol{A} \cap (\boldsymbol{B} \cup \boldsymbol{C}) = (\boldsymbol{A} \cap \boldsymbol{B}) \cup (\boldsymbol{A} \cap \boldsymbol{C})$ – дистрибутивтік;

13) $\boldsymbol{A} \cap (\boldsymbol{B} \setminus \boldsymbol{C}) = (\boldsymbol{A} \cap \boldsymbol{B}) \setminus (\boldsymbol{A} \cap \boldsymbol{C})$ – дистрибутивтік;

14) $\boldsymbol{A} \cup \overline{A} = \boldsymbol{U}$ – толықтыру қасиеті;

15) $\boldsymbol{A} \cap \overline{A} = \varnothing$ – толықтыру қасиеті;

16) $\overline{A \cup B} = \overline{A} \cap \overline{B}$ – де Морган заңы;

17) $\overline{A \cap B} = \overline{A} \cup \overline{B}$ – де Морган заңы;

18) $\overline{\overline{\mathbf{A}}} = \mathbf{A}$ – инволютивтік;

19) $\boldsymbol{A} \setminus \varnothing = \boldsymbol{A}$ – айырма қасиеті;

20) $\boldsymbol{A} \setminus \boldsymbol{A} = \varnothing$ – айырма қасиеті;

21) $\boldsymbol{A} \setminus \boldsymbol{B} = \boldsymbol{A} \cap \overline{A} = \varnothing$ – айырма қасиеті;

22) $\boldsymbol{B} \setminus \boldsymbol{A} = \boldsymbol{B} \cap \overline{A} = \boldsymbol{B} \setminus (\boldsymbol{B} \cap \boldsymbol{A})$– айырма қасиеті;

II. Симметриялық айырма және тік көбейту:

1) $\boldsymbol{A} \triangle \varnothing = \boldsymbol{A}$ – нөл қасиеті;

2) $\boldsymbol{A} \triangle \boldsymbol{A} = \varnothing$ – идемпотенттік;

3) $\boldsymbol{A} \triangle \boldsymbol{B} = (\boldsymbol{A} \cup \boldsymbol{B}) \setminus (\boldsymbol{A} \cap \boldsymbol{B})$ – симметриялық айырма қасиеті;

4) $\boldsymbol{A} \triangle \boldsymbol{B} = \boldsymbol{B} \triangle \boldsymbol{A}$ – коммутативтік;

5) $(\boldsymbol{A} \triangle \boldsymbol{B}) \triangle \boldsymbol{C} = \boldsymbol{A} \triangle (\boldsymbol{B} \triangle \boldsymbol{C}) = \boldsymbol{A} \triangle \boldsymbol{B} \triangle \boldsymbol{C}$ – ассоциативтік;

6) $(\boldsymbol{A} \cup \boldsymbol{B}) \times \boldsymbol{C} = (\boldsymbol{A} \times \boldsymbol{C}) \cup (\boldsymbol{B} \times \boldsymbol{C})$ – дистрибутивтік;

7) $\boldsymbol{A}\times(\boldsymbol{B}\cup\boldsymbol{C}) = (\boldsymbol{A}\times\boldsymbol{B}) \cup (\boldsymbol{A}\times\boldsymbol{C})$ – дистрибутивтік;

8) $(\boldsymbol{A}\cap\boldsymbol{B})\times\boldsymbol{C} = (\boldsymbol{A}\times\boldsymbol{C}) \cap (\boldsymbol{B}\times\boldsymbol{C})$ – дистрибутивтік;

9) $\boldsymbol{A}\times(\boldsymbol{B}\cap\boldsymbol{C}) = (\boldsymbol{A}\times\boldsymbol{B}) \cap (\boldsymbol{A}\times\boldsymbol{C})$ – дистрибутивтік;

10) $(\boldsymbol{A}\setminus\boldsymbol{B})\times\boldsymbol{C} = (\boldsymbol{A}\times\boldsymbol{C}) \setminus (\boldsymbol{B}\times\boldsymbol{C})$ – дистрибутивтік;

11) $\boldsymbol{A}\times(\boldsymbol{B}\setminus\boldsymbol{C}) = (\boldsymbol{A}\times\boldsymbol{B}) \setminus (\boldsymbol{A}\times\boldsymbol{C})$ – дистрибутивтік;

І.2.1.4-мысалдар: Мейлі $\boldsymbol{A}=\{1, 2\}$, $\boldsymbol{B}=\{a,b\}$, $\boldsymbol{C}=\{+, -\}$. Сонда дистрибутивтік $(\boldsymbol{A}\cup\boldsymbol{B})\cup\boldsymbol{C} = \boldsymbol{A}\cup(\boldsymbol{B}\cup\boldsymbol{C}) = \boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C}$ былай болады $(\{1,2\}\cup\{a,b\}) \cup \{+,-\} = \{1,2\}\cup(\{a,b\})\cup\{+,-\}) = \{1,2\}\cup\{a,b\}\cup\{+,-\}$.

І.2.1-жаттығу. Мейлі $\boldsymbol{A}=\{1,2,4\}$, $\boldsymbol{B}=\{3,4,5,6\}$. Амалды орында:

1) $\boldsymbol{A} \cap \boldsymbol{B}$;

2) $\boldsymbol{A}\times\boldsymbol{B}$;

3) $\boldsymbol{A} \setminus \boldsymbol{B}$;

4) $\boldsymbol{A}\triangle\boldsymbol{B}$.

І.2.1-сұрақтар: Келесі сұрақтарға жауап беріңіз:

1. Жиын деген не?
2. Жиындар қалай беріледі?
3. Әмбебап жиын деген не?
4. Ішжиын қалай анықталады?
5. Эйлер-Венн диаграммасы деген не?
6. Бірігу үшін Эйлер-Венн диаграммасы қалай көрінеді?
7. Жиындардың тік көбейтіндісі қалай анықталады?
8. Жиындардың айырмасы қалай анықталады?
9. Жиындардың симметриялық айырмасы қалай анықталады?
10. Жиындардың индикаторлық функциясы қалай анықталады?
11. Де Морган заңы не көрсетеді?
12. Егер $\boldsymbol{A}=\{1,2,3\}$, $\boldsymbol{B}=\{3,4,5\}$, онда $\{1,2,3,4,5\}$ қалай шығады?
13. Егер $\boldsymbol{A}=\{1,2,4\}$, $\boldsymbol{B}=\{4,3,2\}$, онда $\{2,4\}$ қалай шығады?
14. Егер $\boldsymbol{A}=\{1,2,3,4\}$, $\boldsymbol{B}=\{3,4,5,6\}$, онда $\{2,4\}$ қалай шығады?
15. Дистрибутивтік деген не?
16. Ассоциативтік деген не?
17. Коммутативтік деген не?

I.2.2. *Қатынастар және оларды бейнелеу әдістері*

Бұл параграфта қатынас ұғымы және олардың бейнелеу әдістері қарастырылады, мысалдар ұсынылады, тапсырмалар беріледі, сұрақтар қойылады [14,17,22,27,28,33].

I.2.2.1-анықтама. Егер $\boldsymbol{A}$ және $\boldsymbol{B}$ бір ғана типті жиындар болса, онда келесі қатынастарды енгізуге болады:

1) $\boldsymbol{A} = \boldsymbol{B}$: ***A*** *тең* ***B***, егер ***A*** мен ***B*** бірдей элементтерден тұрса, бұл ***A*** жиыны ***B*** жиынының ішжиыны болатындығын білдіреді;

2) $\boldsymbol{A} \subseteq \boldsymbol{B}$: ***A*** *қамтылады* ***B***, егер ***A***-ның элементтері ***B***-ға кіретін болса, бұл ***A*** жиыны ***B***-ның *ішжиыны* болатындығын білдіреді;

3) $\boldsymbol{A} \subset \boldsymbol{B}$: ***A*** *қатал қамтылады* ***B***, егер ***A***-ның барлық элементтері ***B***-ға кіретін болса және ***A*** мен ***B*** тең болмаса, яғни ***B***-ның кейбір элементері ***A***-ға кірмейді, бұл ***A*** жиыны ***B***-ның *меншікті ішжиыны* болатындығын білдіреді.

Дәл осылай *қамтиды* $\boldsymbol{A} \supseteq \boldsymbol{B}$ және *қатал қамтиды* $\boldsymbol{A} \supset \boldsymbol{B}$ қатынастарын анықтауға болады.

Жоғарыда ендірілген $=$, $\subseteq$ және $\subset$ қатынастары тік көбейтінді $\boldsymbol{A}\times\boldsymbol{B}$-ның ішжиындары екендігін оңай байқауға болады.

Сонымен, кез келген қатынас белгілі заңмен ерекшелінген тік көбейтіндінің қандай-да бір ішжиыны болады деп есептеуге.

I.2.2.1-ескертпе. Бос жиын $\varnothing$ кез келген ақырлы жиынның меншікті ішжиыны болады.

I.2.2.1-мысалдар:

1) егер $\boldsymbol{A}=\{a,b,c\}$, $\boldsymbol{B}=\{b,a,c\}$, онда $\boldsymbol{A}=\boldsymbol{B}$;
2) егер $\boldsymbol{A}=\{1,2,3,4\}$, $\boldsymbol{B}=\{3,1,4,2\}$, онда $\boldsymbol{A} \subseteq \boldsymbol{B}$;
3) егер $\boldsymbol{A}=\{1,2,3\}$, $\boldsymbol{B}=\{3,1,4,2\}$, онда $\boldsymbol{A} \subset \boldsymbol{B}$.

I.2.2.2-анықтама. Мейлі $\boldsymbol{A_1}$, $\boldsymbol{A_2}$, …, $\boldsymbol{A_n}$ – бір-бірінен өзгешелігі міндетті емес қандай-да бір жиындар, $n\geq1$. Сонда $\boldsymbol{A_1}$, $\boldsymbol{A_2}$, …, $\boldsymbol{A_n}$ жиындарындағы *n өлшемді қатынас* деп мына ішжиынды айтады

$$\boldsymbol{R^n} \subseteq \boldsymbol{A_1} \times \boldsymbol{A_2} \times \ldots \times \boldsymbol{A_n},$$

мұнда $\boldsymbol{R}_1$– $\boldsymbol{A}_1$-дегі унарлы қатынас, $\boldsymbol{R}^2$ – $\boldsymbol{A}_1$, $\boldsymbol{A}_2$ -дегі бинарлы қатынас, $\boldsymbol{R}^3$ – $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$ үштік қатынас на $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$ және т.с.с.

А жиынындағы әрбір *бірлік қатынас* оның ішжиынының сипаттық қасиеті болады. ***А*** жиынындағы барлық бірлік қатынастардың жиыны ***А***-дағы барлық ішжиындардың жиынымен сәйкес келеді.

I.2.2.2-мысалдар:

1. $R_1^1=\{n|n\in N \& n<100\}$ бірлік қатынасы 100-ден кіші натурал сандардың жиынын анықтайды;

2. $R_2^1=\{n|\forall k\in N(n=2*k)\}$ бірлік қатынасы жұп натурал сандардың жиынын анықтайды;

3. $R_3^1=\{n|\forall k\in Z(n=2*k+1)\}$ бірлік қатынасы тақ бүтін сандардың жиынын анықтайды.

I.2.2.3-анықтама. *Екілік қатынас* жиындар сыңарында анықталады және үш әдістің біреуімен бейнеленді:

1) *префикстік жазба* – қатынас таңбасы екілік қатынастың алдында жазылады;

2) *инфикстік жазба* – қатынас таңбасы екілік қатынастың ортасында жазылады;

3) *постфикстік жазба* – қатынас таңбасы екілік қатынастың соңында жазылады;

Элементтер сыңарында анықталған екілік қатынасты кестемен бейнелеуге болады: жолдар сыңардың бірінші элементіне сәйкес, бағандар сыңардың екінші элементіне сәйкес, ал нақты элементтер арасында қатынастың болуы арнаулы символ «1» немесе басқамен белгіленеді.

I.2.2.3-мысалдар:

1. Егер $a\in \boldsymbol{A}$ мен $b\in \boldsymbol{B}$ екілік қатынас ***R***-де болса, онда оны былай жазуға болады:

$\boldsymbol{R}ab$ – префикстік жазба;

$a\boldsymbol{R}b$ – инфикстік жазба;

$ab\boldsymbol{R}$ – постфикстік жазба.

2. Егер $\boldsymbol{A} = \{a_1, \ldots, a_r\}$ мен $\boldsymbol{B} = \{b_1, \ldots, b_s\}$ жиындары екілік қатынас $\boldsymbol{R}$-де болса, онда оны кесте (I.2.2-кесте) арқылы бейнелеуге болады. Мұнда a_i (i=1,2,…,r) жолдармен, b_j (j=1,2,…, s) бағандармен, ал $a_i\boldsymbol{R}b_j$ қатынасының болуы 1-мен белгіленеді:

I.2.2-кесте. Екілік қатынас.

$\boldsymbol{R}$	b_1	b_2	…	b_{s-1}	b_s
a_1	1		…	1	
a_2		1	…		
…	…	…	…	…	…
a_{r-1}	1		…	1	
a_r			…		1

I.2.2.4-анықтама. $\boldsymbol{S}$ жиынындағы $\boldsymbol{R}$ екілік қатынасы:

1) *рефлексивті*, егер әрбір $s\in\boldsymbol{S}$ үшін $s\boldsymbol{R}s$ орындалса;

2) *транзитивті*, егер кез келген s, t, $u\in\mathbf{S}$ үшін $s\boldsymbol{R}t$ мен $t\boldsymbol{R}u$-дан $s\boldsymbol{R}u$ шықса;

3) *симметриялы*, егер кез келген s, $t\in\boldsymbol{S}$ үшін $s\boldsymbol{R}t$-дан $t\boldsymbol{R}s$ шықса;

антисимметриялы, егер $a\boldsymbol{R}b$ және $b\boldsymbol{R}a$-дан $a=b$ шықса.

I.2.2.4-мысалдар:

1. Сандарда анықталған > қатынасы рефлексивті емес.
2. Сандарда анықталған =, ≥, > қатынастары транзитивті болады.
3. Сандарда анықталған = қатынасы симметриялық болады.

I.2.2.5-анықтама. Екілік қатынас $\boldsymbol{R}$ *эквиваленттік қатынас* деп аталады, егер ол рефлексивтік, транзитивтік және симметриялық қасиеттерді қанағаттардырса.

$\boldsymbol{S}$ жиынындағы әрбір эквиваленттік қатынасқа осы жиынды сыбайлас топтарға бөлудің біреуі ғана сәйкес келеді.

I.2.2.5-мысалдар: «=» қатынасы барлық сандық жиындарында эквиваленттік қатынас болады, мысалы, кез келген k, m, $n \in \boldsymbol{N}$ үшін:

– рефлексивтік: $n = n$;

– транзитивтік: $k < m$ мен $m < n$ қатынасынан $k < n$ шығады;

– симметриялық: из $k = m$ следует $m = k$.

I.2.2.6-анықтама. Қандай да $\boldsymbol{S}$ жиынындағы $\boldsymbol{R}$ екілік қатынасы *аракідік реттеу* қатынасы деп аталады, егер ол рефлексивтік, транзитивтік және антисимметриялық қасиеттерді қанағаттандыратын болса.

I.2.2.6-мысалдар: Аракідік реттеу қатынастары келесілер:

1) $\subseteq$ қатынасы кейбір жиындардың ішжиындары үшін;

2) $\leq$ қатынасы натурал сандар жиынында;

3) $\geq$ қатынасы бүтін сандар жиынында.

I.2.2.7-анықтама.

1. Аракідік реттеу қатынасы бар $\boldsymbol{A}$ жиыны *аракідік реттелген жиын* деп аталады. Дәлірек айтсақ, аракідік реттелген жиын деп $<\boldsymbol{A}, \boldsymbol{R}>$ қосағын айтады, мұнда $\boldsymbol{A}$ – жиын, $\boldsymbol{R}$ – $\boldsymbol{A}$ жиынындағы аракідік реттеу қатынасы.
2. Егер аракідік реттелген $\boldsymbol{A}$ жиынындағы $\boldsymbol{a}$ және $\boldsymbol{b}$ элементтері арасында $\boldsymbol{R}$ аракідік реттеу қатынасына қатысты $\boldsymbol{aRb}$ немесе $\boldsymbol{bRa}$ орын алса, онда *салыстырымды* деп аталады.
3. Егер $\boldsymbol{A}$ жиынының кез келген екі $\boldsymbol{a}$ және $\boldsymbol{b}$ элементі аракідік реттеу $\boldsymbol{R}$ қатынасына қатысты салыстырымды болса, онда $\boldsymbol{A}$ жиынындағы аракідік реттеу сызықтық рет деп аталады.
4. Аракідік реттелген жиындағы әрбір элементтер қосағы салыстырымды болса, онда ол *сызықты реттелген жиын* немесе *тізбе* деп аталады.
5. Егер аракідік реттелген $\boldsymbol{A}$ жиынының $\boldsymbol{a}$ және $\boldsymbol{b}$ элементтері арасында ешқандай аракідік реттеу қатынасы орындалмаса, онда олар *салыстырымсыз* деп аталады. Салыстырылымсыз элементтерінің бар болуы *«аракідік реттелген жиын»* терминінің мағынасын түсіндіреді.

I.2.2.7-мысалдар:

1. $\leq$ қатынасы бар натурал сандардың жиыны ***N*** аракідік реттелген жиын, барлық натурал сандар $\leq$ қатынасына қатысты салыстырылымды және ***N*** тізбе болады.

2. = қатынасы бар барлық нақты сандардың жиыны сызықты реттелген жиын болады, барлық нақты сандар = қатынасына қатысты салыстырымды болады.

3. Мейлі ***A*** – аракідік реттеу қатынастары <, =, > анықталған [0,1] кесіндісіндегі нақтымәнді функциялар жиыны болсын, сонда $\boldsymbol{f(x)=x}$ және $\boldsymbol{g(x)=1-x}$ элементтері салыстырымсыз болады.

I.2.2.8-анықтама. Мейлі ***A*** – аракідік реттелген жиын, ***B*** – оның ішжиыны, яғни $\boldsymbol{A \supseteq B}$. Сонда:

1) ***A*** жиынындағы ***B*** жиынының *төменгі* (*жоғарғы*) *қыры* деп кез келген $b\in \boldsymbol{B}$ үшін $a\leq b$ ($b \leq a$) орындалатын $a\in \boldsymbol{A}$ элементі аталады;

2) $a\in \boldsymbol{A}$ элементі ***A*** жиынындағы *ең кіші* (*ең үлкен*) деп аталады, егер a осы ***A*** жиынының өзінде *төменгі* (*жоғарғы*) *қыр* болса.

3) $a\in \boldsymbol{A}$ элементі ***A*** жиынында *минималды* (*максималды*) деп аталады, егер $b < a$ ($a < b$) орындалатындай $b\in \boldsymbol{A}$ болмаса

A жиынының ең кіші (ең үлкен) элементі осы жиындағы жалғыз минималды (максималды) болады.

I.2.2.8-мысалдар:

1. ***A*** жиынының барлық ішжиындарының жиынында жалғыз ең кіші элемент $\varnothing$ және ең үлкен элемент ***A***-ның өзі болады.

2. Натурал сандар жиынында ең кіші элемент 1 бар, ал ең үлкен элемент жоқ.

3. Бүтін сандар жиынында ең кіші, ең үлкен, минималды және максималды элемент болмайды.

A және ***B*** жиындарының элементтері *өзара бірмәнді сәйкестікте* болады, егер әрбір элемент $a\in \boldsymbol{A}$ үшін белгілі бір заңдылықпен жалғыз ғана элемент $b\in \boldsymbol{B}$ сәйкес қойылса және әрбір $b\in \boldsymbol{B}$ тек бір ғана $a\in \boldsymbol{A}$ элементімен сәйкестікте болса. Егер ***A*** және ***B*** жиындары элементтерінің арасында өзара бірмәнді сәйкестік орнатылатын болса, онда олар *эквивалентті* (*тең қуатты*) деп аталады.

I.2.2.2-ескертпе. Екілік қатынас $<\boldsymbol{R}, \boldsymbol{A}, \mathbf{B}>$ жиындар үштігімен берілуі мүмкін, мұнда $\boldsymbol{R} \subseteq \boldsymbol{A} \times \boldsymbol{B}$ – қатынас графигі және ол былай жазылады $(a, \underline{b}) \in \boldsymbol{R}$ немесе $a\boldsymbol{R}b$. Сонда мыналар анықталады:

Анықталу облысы: $\mathbf{Dom}\ \boldsymbol{R} = \{x \in \boldsymbol{A}: \exists y \in \boldsymbol{B}(x, y) \in \boldsymbol{R}\}$;

Өзгеру облысы: $\mathbf{Ru}\boldsymbol{n}\ \boldsymbol{R} = \{y \in \boldsymbol{B}: \exists x \in \boldsymbol{A}(x, y) \in \boldsymbol{R}\}$;

Кері қатынас: $\boldsymbol{R}^{-1} = \{(y, x) \in \boldsymbol{B} \times \boldsymbol{A}: (x, \underline{y}) \in \boldsymbol{R}\}$;

Композиция қатынасы: $\boldsymbol{R} \subseteq \boldsymbol{A} \times \boldsymbol{B}$, $\boldsymbol{S} \subseteq \boldsymbol{B} \times \boldsymbol{C}$,

$$\boldsymbol{R} \cdot \boldsymbol{S} = \{(x, z) \in \boldsymbol{A} \times \boldsymbol{C}: \exists y \in \boldsymbol{B}\ [(x\boldsymbol{R}y)\ \&(y\boldsymbol{S}z)]\}.$$

I.2.2.9-анықтама. $\boldsymbol{f} \subseteq \boldsymbol{X} \times \boldsymbol{Y}$ екілік қатынасы $\boldsymbol{X}$-тен $\boldsymbol{Y}$-ке *функция*, деп аталады, егер $\mathbf{Dom}\ \boldsymbol{R} = \boldsymbol{X}$ және $(x, y) \in \boldsymbol{f}$, $(x, z) \in \boldsymbol{f} \Rightarrow y = z$.

Функция $\boldsymbol{f}: \boldsymbol{X} \to \boldsymbol{Y}$:

1) *сюръективтік* деп аталады, егер кез келген $y \in \boldsymbol{Y}$ үшін $y = \boldsymbol{f}(x)$ болатындай $x \in \boldsymbol{X}$ бар болса, яғни $\forall y \in \boldsymbol{Y}\ \exists x \in \boldsymbol{X}(y = \boldsymbol{f}(x))$;

2) *инъективтік* деп аталады, x_1, $x_2 \in \boldsymbol{X}$ үшін $x_1 \neq x_2$ болғандықтан $\boldsymbol{f}(x_1) \neq \boldsymbol{f}(x_2)$ шықса, яғни $\forall x_1 \in \boldsymbol{X}\ \forall x_2 \in \boldsymbol{X}\ (x_1 \neq x_2 \Rightarrow \boldsymbol{f}(x_1) \neq \boldsymbol{f}(x_2))$;

3) *биективтік* деп аталады, егер ол сюръективтік және инъективтік болса.

Кез келген екілік функциямен үштік қатынасты байланыстыруға болады, мысалы, егер екілік функция $\boldsymbol{f}(x,y)$ берілсе, онда оны $z = \boldsymbol{f}(x,y)$ болатындай үштік қатынас $\boldsymbol{R}^3(x, y, z)$-пен байланыстуға болады.

I.2.2.9-мысалдар: Мейлі x, y, $z \in \boldsymbol{N}$ – натурал сандар және $\boldsymbol{f}(x,y) = z$ екілік функция берілсін, сонда:

1) егер x тең 3, y тең 5, z тең 8, ал $\boldsymbol{f}$ қосу амалы+болса, онда $\boldsymbol{f}(x,y) = z$ орнына 3+5=8 деп жазады және онымен 8 = 3+5 болатындай ***Add***(3, 5, 8) үштік қатынасын байланыстыруға болады.

2) егер x тең 8, y тең 2, z тең 4, ал $\boldsymbol{f}$ бөлу амалы: болса, онда $\boldsymbol{f}(x,y) = z$ орнына 8:2=4 деп жазады және онымен 4 = 8:2 болатындай ***Dev***(8, 2, 4) үштік қатынасын байланыстыруға болады.

Кейбір x, $y \in \boldsymbol{N}$ үшін бөлу амалының нәтижесі бүтін сан емес екендігі және сондықтан олар үшін үш орынды ***Dev***(x, y, z) қатынасы орындалмайтындығы белгілі. Сондықтан мұндай сандар үшін нәтижелер жайында қосымша шарттар анықталуы қажет.

I.2.2.10-анықтама. $\boldsymbol{A}$ жиынындағы $\boldsymbol{R}$ қатынасының *транзитивтік тұйықталуы* деп $\boldsymbol{R}$-ді ішжиын ретінде қамтитын барлық транзитивті қатынастардың қиылысуын айтады (немесе $\boldsymbol{R}$-ді ішжиын ретінде қамтитын минималды транзитивті қатынас).

Транзитивті қатынас кез келген қатынас үшін бар болады. Ол үшін кез келген транзитвті қатынастар жиындарының қиылысуы транзитивті екенін ескерейік. Сондай-ақ Более того, $\boldsymbol{R}$ қатынсын қамтитын транзитивті қатынас міндетті түрде бар болады.

Транзитивті тұйықталы келесі қасиеттерге ие болады:

1)Рефлексивті қатынастың транзитивті тұйықталуы рефлексивті болады, себебі транзитивті қатынас алғашықы қатынасты қамтиды;

2)Симметриялы қатынастың транзитивті тұйықталуы симметрия болады. Дегендей, мейлі транзитивті қатынас $a\boldsymbol{R}b$ бар болсын, демек $a\boldsymbol{R}x_1$, $x_1\boldsymbol{R}x_2$, …, $x_n\boldsymbol{R}b$ болатындай x_1, x_2, **…**, x_n бар болады. БІрақ $\boldsymbol{R}$ қатынасының симметриялығынан $b\boldsymbol{R}x_n$, $x_n\boldsymbol{R}x_{n-1}$, …, $x_1\boldsymbol{R}a$ шығады, сондықтан $b\boldsymbol{R}a$.

3)Транзитивті тұйықталу антисимметриялықты сақтамайды, мысалы, $\{a, b, c\}$ жиынындағы қатынастар $\{(a, b), (b, c), (c, a)\}$ үшін.

4)Транзитивті қатынастың транзитивті тұйықталуы – оның өзі.

$\boldsymbol{R^*}=\boldsymbol{R}^+\cup\boldsymbol{R}^0$ қатынасын, мұнда $\boldsymbol{R}^0=\{(\varepsilon, \varepsilon): \varepsilon\in\boldsymbol{A}\}$ кейде рефлексивті-транзитивті дейді, дегенмен де "транзитивті тұйықталу" деп $\boldsymbol{R^*}$ ұйғарылған. Әдетте бұл қатынастардың арасында айырмашылық мардымды болмайды.

I.2.2.10-мысалдар:

1. Кез келген $n\in\boldsymbol{N}$ үшін $n = n$ орындалады.
2. Кез келген a, b, $c\in\boldsymbol{N}$ үшін $a<b$ мен $b<c$ болса $a<c$ шығады.
3. Кез келген x, $y\in\boldsymbol{N}$ үшін $x=y$ болса $y= x$ шығады.
4. Егер $\boldsymbol{A}-$ қалалар жиын болса және оларда " x-тен y-ке автобус маршруты бар" деген $x\boldsymbol{R}y$ қатынасы берілсе, онда бұл қатынастың транзитивті тұйықталуы "x-тен y-ке автобуспен жетуге мүмкіншілік бар" дегенді білдіреді.
5. Егер $\boldsymbol{R}^4 = \boldsymbol{R}$

I.2.2-жаттығу. $\boldsymbol{A}=\{a,b\}$, $\boldsymbol{B}=\{0,1\}$, $\boldsymbol{C}=\{c,2\}$ болса мынаны анықта:

1) $\boldsymbol{A}=\boldsymbol{B}$;
2) $\boldsymbol{A} \subseteq \boldsymbol{B}$;
3) $\boldsymbol{A} \subset \boldsymbol{B}$;
4) $\boldsymbol{A} \setminus \boldsymbol{B}$;
5) $\boldsymbol{A} \times \boldsymbol{B}$;
6) $\boldsymbol{A} \triangle \boldsymbol{B}$;
7) $\boldsymbol{A} \cup \boldsymbol{B}) \times \boldsymbol{C}$;
8) $\boldsymbol{A} \times (\boldsymbol{B} \cup \boldsymbol{C})$;
9) $(\boldsymbol{A} \cap \boldsymbol{B}) \times \boldsymbol{C}$;
10) $\boldsymbol{A} \times (\boldsymbol{B} \cap \boldsymbol{C})$;
11) $\boldsymbol{A} \cap (\boldsymbol{B} \cup \boldsymbol{C})$;
12) $\boldsymbol{A} \cup (\boldsymbol{B} \cap \boldsymbol{C})$;
13) $\boldsymbol{A} \setminus (\boldsymbol{B} \cup \boldsymbol{C})$;
14) $(\boldsymbol{A} \setminus \boldsymbol{B}) \cap \boldsymbol{C}$.

I.2.2-сұрақтар:

1. Қатынас деген не?
2. Бірлік қатынас қалай анықталады?
3. Екілік қатынаста қанша жиын қатысады?
4. Екілік қатынас қандай қасиеттерге ие?
5. Екілік қатынас қалай беріледі?
6. Аракідік реттік деген не?
7. Анықталу облысы деген не?
8. Өзгеру облысы деген не?
9. Функция қалай анықталады?
10. Қатынастар композициясы қалай анықталады?
11. Керілеті шарты неден тұрады?
12. Жиынның қуаты қалай анықталады?
13. Жиындар бірігулерінің қуаты неден тұрады?
14. Жиындар қиылысуының қуаты неден тұрады?
15. Жиындар тік көбейтулерінің қуаты неден тұрады?
16. Екі жиынның эквиваленттігі қалай анықталады?
17. Қандай жиындар тең қуатты болады?
18. Қандай элементтер салыстырымсыз болады?
19. Сызықтық рет деген не?

І.2.3. Символдар, тізбелер және олардағы амалдар

Бұл параграфта символдар мен тізбелер қарастырылады, олардағы амалдар анықталады және осы амалдардың қасиеттері талқыланады, мысалдар ұсынылады, тапсырмалар мен сұрақтар беріледі [1-9,11-18,21,25,27-32].

І.2.3.1-анықтама:

1. Символ деп кез келген тілді анықтағандағы минималды бірлікті айтады.
2. Символдар ретінде терминалдар – өздігінен анықталған тұрақты шамалар немесе бейтерминалдар басқа символдар арқылы анықталған айнымалы шамалар бола алады.
3. Терминал деп әріптер, цифрлар және арнаулы таңбалар (амалдар, жақшалар, тыныс белгілері және т.б.) аталады.
4. Бейтерминалдар деп қандай да бір ұғымды белгілейтін морфемдер, лексемалар, сөздер, сөйлемдер және т.б. аталады.

І.2.3.1-келісім.

1. Терминал атауы өзінің мәнімен беріледі.
2. Бейтерминал атауы кіші грек әріптерімен беріледі, оның индексі болады немесе болмайды, индекстер ретінде цифрлар немесе кіші латын әріптері алынады.

Берілген символдармен жұмыс істеу үшін осы символдарда анықталған амалдар қолдану керек, ал кез келген амалды қолдану үшін оның *анықтамасын*, *белгілеуін* және *қасиетін* білу қажет.

Символдарда анықталған амалдар арқылы символдық өрнектер құрылады. Символдық өрнектерді өңдеу үшін символдық амалдардың қасиеттері пайдаланылады. Бұл қасиеттер оларлардың орындалуын жеңілдетіп, қарапайымдайды.

Символдарда *құрастыру* амалдары анықталған, олар екі символдан бір символдық шама алуға мүмкіндік береді.

Құрастыру амалдарының ішінде ең қарапайымы *конкатенация* (*тіркесу*). Әдетте бұл амал ‘$\cdot$’ таңбасымен таңбасымен белгіленеді және оны екі сиимволдық $\boldsymbol{\alpha}$ мен $\boldsymbol{\beta}$ шамаларында анықтайды. Ал $\boldsymbol{\alpha}$

символы мен β символының тіркесу нәтижесі γ болады және мәні α мен β мәндерінің тіркесі болады және ол былай жазылады $\alpha\cdot\beta = \gamma$. Мұнда α мен β мәндері бір терминал немесе терминалдар тізбесі болатындығын ескерткен жөн. Сонымен, символдардан конкатенация амалы арқылы әртүрлі *тізбелер* (*сөздер*, *жолдар*) құрастырылады. Конкатенация амалы тізбелерде анықталуы мүмкін, оның нәтижесі тағы да тізбе болады.

Толықтылық үшін символдық шамалар жиынында «*бос тізбе*» деп аталатын абстрактылы шама ендіріледі. Бос тізбе бір де бір символды қамтымайды және ‘ε’ арқылы белгіленеді.

I.2.3.2-ескертпелер:

1. Символдық өрнектерде ‘·’ амалының таңбасы жазылмайды, ол тек амалдың қасиетін анықтағанда, басқа амалмен шатастырмау үшін жазылады.

2. Бос тізбе символдық шамалар жиынында сандар жиынындағы нөлдің рөлін орындайды.

Келесі құрастыру амалына *дизъюнкция* (*таңдау*) амалы жатады, ол ‘∨’ арқылы белгіленеді. Екі берілген α және β символдарында анықталған таңдау амалының нәтижесі, мәні α немесе β мәнінен құралған, γ символы болады, ол $\alpha \vee \beta = \gamma$ деп жазылады. Мысалы, егер α тізбесінің мәні ‘робот’, ал β тізбесінің мәні ‘компьютер’ болса, онда γ тізбесінің мәні ‘робот’ немесе ‘компьютер’ болады.

Символда және бос емес тізбеде *жою* амалын $\alpha^0=\varepsilon$ анықтауға болады, ол бос тізбеде анықталмаған.

Құрастыру амалдарына *итерация* амалы да жатады, ол ‘*’ арқылы белгіленеді.

Кез келген символдық шама α үшін итерация ‘*’ амалы былай анықталады $\alpha^* = \alpha^0 \vee \alpha^1 \vee \alpha^2 \vee \alpha^3 \vee \ldots \vee \alpha^k \vee \ldots$,

мұнда $\alpha^0=\varepsilon,\ \alpha^1=\alpha,\ \ \alpha^k=\underbrace{\alpha\cdot\alpha\cdot\ldots\cdot\alpha}_{k}$, $k=2,3,\ldots$

Жоғарыда анықталлған амалдар арқылы символдық өрнекті дұрыс құру үшін олардың орындалу ретін білу қажет: * – бірінші,

^{0} және $\cdot$ – екінші, $\vee$ – үшінші. Басқа орындалу ретін орнату үшін жақшалар қолданылады.

Жоғарыда қарастырылған амалдарды *базалық символдық амалдар* деп атауға болады. Олардың көмегімен символдық өрнектерді өңдеуге қажет болатын кез келген күрделі символдық амадық амалдарды анықтауға болады. Бірақ ол үшін базалық символдық амалдардың алгебралық қасиеттерін білу қажет.

Мейлі келген бос емес тізбелері α, β, γ және бос тізбе ε берілсін. Сонда символдық шамалардағы амалдарының алгебралық қасиеттері төмендегі тепе-теңдіктермен беріледі:

$\alpha\cdot\varepsilon=\varepsilon\cdot\alpha=\alpha$

$\alpha\cdot\beta\neq\beta\cdot\alpha$ - кейбір α, β үшін;

$\alpha\cdot(\beta\cdot\gamma)=(\alpha\cdot\beta)\cdot\gamma=\alpha\cdot\beta\cdot\gamma$

$\alpha^0=\varepsilon$

$\alpha\alpha^0=\alpha^0\alpha=\alpha$

$\alpha\vee\beta=\beta\vee\alpha$

$\alpha\vee(\beta\vee\gamma)=(\alpha\vee\beta)\vee\gamma=\alpha\vee\beta\vee\gamma$

$\alpha\cdot(\beta\vee\gamma)=\alpha\cdot\beta\vee\alpha\cdot\gamma$

$(\alpha\vee\beta)\cdot\gamma=\alpha\cdot\gamma\vee\beta\cdot\gamma$

$\alpha\vee\alpha=\alpha$

$\varepsilon^*=\varepsilon$

$(\alpha^*)^*=\alpha^*$

$\alpha^*=\alpha\vee\alpha^*$

Базалық символдық амалдардың осы қасиеттерін пайдалана отырып, күрделі символдық амалдарды анақтауға болады.

I.2.3.2-анықтама.

1. Берілген ω тізбесіндегі символдар саны ω *тізбесінің ұзындығы* деп аталады, мұнда әрбір символ сонша рет саналады, қанша ол ω тізбесінде кездессе, ол |ω| арқылы белгіленеді. Бос тізбенің ұзындығы нөлге тең, яғни $|\varepsilon| = 0$.

2. a символы ω тізбесіне кіреді, егер қандай да бір η және ξ тізбелері бар болып, $\omega = \eta a \xi$ орындалса және a символының ω тізбесіне кірілім саны $|\omega|_a$ деп таңбаланады.

3. τ тізбесі ω тізбесінің *іштізбесі* деп аталады және ол былай $\tau \subseteq \omega$ белгіленеді, егер $\omega = \eta\tau\xi$ болатындай қандай да бір η және ξ тізбелері бар болса яғни $\tau \subseteq \omega \rightleftharpoons \exists\eta\exists\xi(\omega = \eta\alpha\xi)$. τ тізбесінің ω тізбесіне кірілім саны $|\omega|_\tau$ деп таңбаланады.

4. Берілген ω тізбесіне *кері тізбе* деп ω тізбесінің символдарынан кері ретпен құралған тізбені айтады. Кері тізбе алғашқы тізбе ω-ға *керілеу* амалын қолданумен алынады және ол және ол ω^R арқылы таңбаланады.

I.2.3-мысалдар:

1. Егер $\boldsymbol{\alpha}$-ның мәні ‘о’ символы және $\boldsymbol{\beta}$-ның мәні ‘л’ символы болса, онда $\boldsymbol{\gamma}$-ның мәні ‘ол’ болады.

2. Егер $\boldsymbol{\alpha}$ тізбесінің мәні ‘қазақ’, ал $\boldsymbol{\beta}$ тізбесінің мәні ‘стан’ болса, онда $\boldsymbol{\gamma}$ тізбесінің мәні ‘қазақстан’ болады.

3. Мейлі $\boldsymbol{\alpha}$ = ‘менің отаным’ және $\boldsymbol{\beta}$ = ‘қазахстан’ болса. Сонда $\boldsymbol{\gamma}$ тізбесінің мәні ‘менің отаным қазахстан’.

4. $abbac$ тізбесі үшін мыналарды $|abbac| = 4$, $|abbac|_a=2$, $|abbac|_b= 2$, $|abbac|_c = 1$ жазуға болады.

5. Мыналар ақиқат: $ab^3 = abbb$; $a^3b^3 = aaabbb$; $(ab)^3 = ababab$.

6. Егер aaa тізбесі берілсе, онда $aaa°=aa°a=a°aa=aa$.

7. Егер $\omega = abcd$, онда $\omega^R = dcba$.

8. Егер $\alpha = aa$, $\beta = abaacd$, $\xi = cd$, $\zeta = ab$, онда $\beta = \zeta\alpha\,\xi$, т.е. $\alpha \subseteq \beta$.

9. Егер $\omega = abcada$, онда $|\omega|_a = 3$.

10. Егер $\omega = abcabcdabda$ және $\sigma =ab$, онда $|\omega|\sigma = 2$.

11. Мейлі $\boldsymbol{a{\cdot}b{\cdot}c}$ тізбесі берілсін. Осы тізбедегі символдарды кері ретпен алмастыру керек, яғни $(\boldsymbol{a{\cdot}b{\cdot}c})^R = \boldsymbol{c{\cdot}b{\cdot}a}$ алу керек. Ол үшін базалық амалдар жою мен тіркесудің қасиеттерін пайдалана отырып, күрделі символдық *керілеу* амалын келесі амалдар тізбегін орындау керек

$$(\boldsymbol{a{\cdot}b{\cdot}c})^{\boldsymbol{R}}=\boldsymbol{a}^0{\cdot}\boldsymbol{b{\cdot}c}=\varepsilon{\cdot}\boldsymbol{b}^0{\cdot}\boldsymbol{c}=\varepsilon{\cdot}\varepsilon{\cdot}\boldsymbol{c}=\boldsymbol{c{\cdot}b}=\boldsymbol{c{\cdot}b{\cdot}a}\ \cdot$$

I.2.3-жаттығулар:

1. Мейлі $\alpha = abcd$, $\boldsymbol{\beta} = ecbaf$, $\boldsymbol{\gamma} = def$ болса мыналарды анықтаңыз:

1) β тізбесінің барлық іштізбесін;

2) $\alpha\beta$ тізбесінің барлық іштізбесін;

3) $\beta\gamma$ тізбесінің барлық іштізбесін;

4) α тізбесінің барлық іштізбесін;

5) $\alpha\boldsymbol{\gamma}$ тізбесінің барлық іштізбесін;

6) $\alpha \cdot \beta$;

7) $\alpha \vee \beta$;

8) $\alpha \cdot (\beta \vee \gamma)$;

9) $(\alpha \vee \beta) \cdot \gamma$;

2. Берілген α және β тізбелерден γ тізбесіне тең болатын өрнекті олардың келесі мәндері үшін тұрғызыңдар:

$\alpha = abcd$, $\beta = ecbaf$, $\gamma = def$;

$\alpha = 12345$, $\beta = 6789$, $\gamma = 28f$;

$\alpha = abcd$, $\beta = 123456$, $\gamma = c2d5$.

3. Берілген α=‘астана’ тізбесінің барлық іштізбелерін табыңдар

I.2.3-сұрақтар:

1. Символ деген не?
2. Терминал мен бейтерминалдың айырмашылығы неде?
3. Бейтерминалдардың атаулары қалай беріледі?
4. Тізбенің ұзындығы неге тең?
5. Бос тізбе деген не?
6. Символдарда конкатенация амалы қалай анықталады?
7. Символдарда дизъюнкция амалы қалай анықталады?
8. Символдарда итерация амалы қалай анықталады?
9. Символдың тізбеге кіретіндігі қалай анықталады?
10. Тізбенің басқа тізбеге кіретіндігі қалай анықталады?
11. Тізбенің ұзындығы деген не?
12. Кері тізбе қалай тұрғызылады?
13. Іштізбенің ұзындығы қандай болуы керек?

I.2.4. Әліпби, тіл және тілдердегі амалдар

Бұл параграфта әліпби мен тіл ұғымдары қарастырылады, олардағы амалдар анықталған және осы амалдардың қасиеттері талқыланған, мысалдар ұсынылады, тапсырмалар беріледі, сұрақтар қойылады [1-9,11-18,21,25,27-32].

I.2.4.1-анықтама.

1. *Әліпби* деген символдар (әріптер) деп аталатын элементтердің бос емес ақырлы жиыны. Ары қарай, әліпби үлкен латын әріптерімен, ал элементтер кіші латын әріптерімен таңбаланады.

2. Берілген әліпбидегі *сөз* (тізбе, жол) деп осы әліпбидегі элементтердің ақырлы тізбегін айтады. Ары қарай, сөздерді гректің кіші әріптерімен таңбалайды.

3. Бір де бір символды қамтымайтын сөзді *бос сөз* деп атайды және ол ε әріпімен таңбаланады.

4. ω сөзінің ұзындығы оның символдар санымен беріледі және ол |ω| деп таңбаланады, мұнда әрбір символ қанша рет кездеседі, сонша рет саналады. Бос сөздің ұзындығы нөлге тең болады, яғни |ε|=0.

5. Егер *α* және *β* ***T*** әліпбиіндегі сөздер болса, онда *α·β* сөзі *α* сөзінің соңына *β* сөзін тіркеп жазғаннан алынған *α* және *β* сөздерінің конкатенациясы (тіркесі) деп аталады. Әдетте жазбада · таңбасын қоспай *αβ* түрінде жазады.

6. Егер *α* – сөз және $n \geq 0$ – оң бүтін сан болса, онда $\alpha^n \rightleftharpoons \underbrace{\alpha\cdot\alpha\cdot\ldots\cdot\alpha}_{n}$ и $\alpha^0 \rightleftharpoons \varepsilon$ (⇌ таңбасы “анықталу бойынша тең ”).

7. Егер τ және ω ***T*** әліпбиінде сөздер болса және қандай да бір сөз ξ үшін ω = τξ орындалса, онда τ сөзі ω сөзінің *префиксі* (*басы*) деп аталады және ол $\alpha \Subset \beta$ деп таңбаланады, яғни $\tau \Subset \omega \rightleftharpoons \exists\xi(\omega = \tau\xi)$.

8. Егер τ және ω ***T*** әліпбиінде сөздер болса және қандай да бір сөз ξ үшін ω = ζτ орындалса, онда τ сөзі ω сөзінің *постфиксі* (*соңы*) деп аталады және ол $\tau \Supset \omega$ деп таңбаланады, яғни $\tau \Supset \omega \rightleftharpoons \exists\zeta(\omega = \zeta\tau)$.

9. Егер τ және ω ***T*** әліпбиінде сөздер болса және қайсы бір сөздер ξ және ζ үшін ω = ζτξ орындалса, онда τ сөзі ω сөзінің *ішсөзі* (*іштізбесі*)

деп аталады және ол $\tau \subseteq \omega$ деп таңбаланады, яғни $\tau \subseteq \omega$, т.е. $\tau \subseteq \omega \rightleftharpoons \exists\zeta\ \exists\xi(\omega = \zeta\tau\xi)$

10. Егер τ және ω ***T*** әліпбиінде сөздер болса және τ сөзі ω сөзінің ішсөзі болса, онда $|\omega|_\tau$ арқылы τ сөзінің ω сөзіне кіру саны таңбаланады.

I.2.4.1-ескертпе. ***T*** әліпбиіндегі кез келген бос емес сөз ω және бос сөз ε үшін $\varepsilon \sqsubset$ және и $\varepsilon \sqsupset \omega$ қатынастар орындалады.

***I.2.4.1-мысалдар*:** Мейлі $\boldsymbol{T}=\{a,b,c,d\}$ әліпбиі берілсін. Сонда:

1. a, bb, ccc, $abcd$, $dcba$ және басқа тізбектер a,b,c,d әріптерінен тұратын ***T*** әліпбиіндегі сөздер болады.
2. $|a|=1$, $|bb|=2$, $|ccc|=3$, $|abcd|=4$.
3. Если $\alpha = ccc$ және $\beta = dddd$, онда $\alpha\beta = cccdddd$.
4. $ab^2 = abb$, $(ab)^3 = ababab$.
5. Если $\alpha = ab$, $\beta = abcd$, $\xi = cd$, онда $\beta = \alpha\xi$, т.е. $\alpha \Subset \beta$.
6. Если $\alpha = cd$, $\beta = abcd$, $\zeta = ab$, онда $\beta = \zeta\alpha$, т.е. $\alpha \Supset \beta$.
7. $\varepsilon \sqsubset abcd$, $a \sqsubset abcd$, $ab \sqsubset abcd$, $abc \sqsubset abcd$, $abcd \sqsubset abcd$.
8. $abcd \sqsupset \varepsilon$, $abcd \sqsupset d$, $abcd \sqsupset cd$, $abcd \sqsupset bcd$, $abcd \sqsupset abcd$.
9. $\varepsilon \Subset abcd$, $a \Subset abcd$, $ab \Subset abcd$, $abc \Subset abcd$, $abcd \Subset abcd$.
10. $abcd \sqsupset \varepsilon$, $abcd \sqsupset d$, $abcd \sqsupset cd$, $abcd \sqsupset bcd$, $abcd \sqsupset abcd$.
11. Егер $\tau = ca$ және $\xi = da$, онда $\tau \sqsubset \omega$ және $\omega = cada$.
12. Егер $\tau = 12$ және $\zeta = 11$, онда $\tau \sqsupset \omega$ және $\omega = 1112$.
13. Егер $\tau=bbb$, $\zeta=aaa$ және $\xi=ccc$, онда $\tau \subseteq \omega$ және $\omega = aaabbbccc$.
14. Егер $\omega = abcdabcd$ және $\tau = cd$, онда $\tau \subseteq \omega$ және $|\omega|_\tau = 2$.

Егер ***T*** – қандай да бір тілдің әліпбиі болса, онда осы әліпбидегі ұзындығы ақырлы барлық тізбелер жиыны былай анықталады:

$$\boldsymbol{T^*} \rightleftharpoons \bigcup_{k=0}^{\infty} T^k,$$

мұнда $\boldsymbol{T}^0 \rightleftharpoons \{\varepsilon\}$ – ұзындығы 0 тізбелер жиыны, $\boldsymbol{T}^k \rightleftharpoons \underbrace{T \cdot T \cdot \ldots \cdot T}_{k}$ – ұзындығы k тізбелер жиыны, $k \geq 1$ – натурал сандар.

$\boldsymbol{T}^{+}$ арқылы барлық бос емес тізбелердің жиыны белгіленеді, яғни $\boldsymbol{T}^{+} = \boldsymbol{T}^{*} \setminus \{\varepsilon\}$. Мысалы, егер $\boldsymbol{T} = \{a\}$, онда $\boldsymbol{T}^{+}$ жиыны былай анықталады $\boldsymbol{T}^{+} = \{a, aa, , aaa, \ldots\}$.

$\boldsymbol{T}^{+}$ жиынындағы барлық тізбелер мағыналы бірлік (сөздер, сөзтіркестері, фразалар, сөйлемдер және мәтіндер) болмайтындығы белгілі. Тілдің грамматикалық ережелерін қанағаттандыратын және семантикалық мағыналары бар бірліктері ғана мағыналы бола алатындығы ескертіледі. Сондықтан, кез келген нақты $\boldsymbol{L}$ тілі $\boldsymbol{T}^{+}$ жиынының меншікті ішжиыны болады, яғни $\boldsymbol{L} \subset \boldsymbol{T}^{+}$. Мысалы, егер $\boldsymbol{T}$ әліпбиінің элементтері ретінде қазақ әріптерін және әртүрлі арнаулы символдарды алсақ, онда казақ тілі $\boldsymbol{T}^{+}$ жиынының меншікті ішжиыны болады және онда қазақ тілінде мағынасы бар сөздер, сөзтіркестері, фразалар, сөйлемдер және мәтіндер ғана қамтылады.

Дегенмен, әрбір тіл берілген әліпбидегі ұзындығы ақырлы тізбелер жиыны болғандықтан, сонда бір ғана әліпбиде берілген тілдердің *бірігуін*, *қиылысуын*, *айырмасын*, *толықтыруын*, *тік көбейтуін*, *симметриялық айырылымын*, *конкатенациясын* және *итерациясын* қарастыруға болады.

I.2.4.2-анықтама:

Мейлі $\boldsymbol{T}$ әліпбиінде екі тіл $\boldsymbol{L_1}$ және $\boldsymbol{L_2}$ берілсін, яғни $\boldsymbol{L_1} \subseteq \boldsymbol{T}^{+}$ және $\boldsymbol{L_2} \subseteq \boldsymbol{T}^{+}$, ал $\boldsymbol{U}$ – универсум, онда мыналарды анықтауға болады:

1. Бірігу: $\boldsymbol{L_1} \cup \boldsymbol{L_2} \rightleftharpoons \{x : x \in \boldsymbol{L_1} \vee x \in \boldsymbol{L_2}\}$;
2. Қиылысу: $\boldsymbol{L_1} \cap \boldsymbol{L_2} \rightleftharpoons \{x : x \in \boldsymbol{L_1} \ \& \ x \in \boldsymbol{L_2}\}$;
3. Тік көбейту: $\boldsymbol{L_1} \times \boldsymbol{L_2} \rightleftharpoons \{(a, b) : a \in \boldsymbol{L_1} \& \ b \in \boldsymbol{L_2}\}$;
4. Толықтыру: $\overline{L} \rightleftharpoons \{x : x \in \boldsymbol{U} \ \& \ x \notin \boldsymbol{L}\}$;
5. Айырма: $\boldsymbol{L_1} \setminus \boldsymbol{L_2} \rightleftharpoons \{x : x \in \boldsymbol{L_1} \ \& \ x \notin \boldsymbol{L_2}\}$;
6. Симметриялық айырма: $\boldsymbol{L_1} \triangle \boldsymbol{L_2} \rightleftharpoons \{x : x \in (\boldsymbol{L_1} \setminus \boldsymbol{L_2}) \vee x \in (\boldsymbol{L_2} \setminus \boldsymbol{L_1})\}$;
7. Конкатенация: $\boldsymbol{L_1} \cdot \boldsymbol{L_2} \rightleftharpoons \{a \cdot b : a \in \boldsymbol{L_1} \& \ b \in \boldsymbol{L_2}\}$;
8. Итерация –Клини жұлдызы (Klee*n*e star): $\boldsymbol{L}^{*} \rightleftharpoons \bigcup_{k=0}^{\infty} L^{k}$, мұнда

$$\boldsymbol{L}^{0} \rightleftharpoons \{\varepsilon\}, \boldsymbol{L}^{\boldsymbol{k}} \rightleftharpoons \underbrace{L \cdot L \cdot \ldots \cdot L}_{k}, \ k \geq 1.$$

I.2.4.2-мысалдар:

1. Егер $\boldsymbol{L_1}= \{aa, bb\}$, $\boldsymbol{L_2}=\{cc, dd\}$, онда

$$\boldsymbol{L_1}\cdot\boldsymbol{L_2} = \{aacc, aadd, bbcc, bbdd\}.$$

2. Егер $\boldsymbol{L}= \{a^k ba^l: 0 < k < l\}$, онда

$$\boldsymbol{L}^2 = \{a^k ba^l ba^m: 0 < k < l\text{-}1,\ m > 1\}.$$

3. Егер $\boldsymbol{T}=\{a, b\}$, $\boldsymbol{L}=\{aa, ab, ba, bb\}$, онда

$$\boldsymbol{L^*}= \{\tau\in \boldsymbol{T^*}: |\tau| = 2\},\ \boldsymbol{L^*}= \{\tau\in \boldsymbol{T^*}: |\tau|_a = 1\ \&\ |\tau|_b = 1\}.$$

I.2.4.2-ескертпелер:

1. $\boldsymbol{T}$ әліпбиіндегі ұзындығы ақырлы сөздердің $\boldsymbol{L}$ жиыны $\leq$ ($\geq$) қатынасы бойынша аракідік реттелген жиын болады және оның барлық сөздері $\leq$ ($\geq$) бойынша салыстырымды.

2. $\boldsymbol{T}$ әліпбиіндегі ұзындығы ақырлы сөздердің $\boldsymbol{L}$ жиыны $\subseteq$ ($\supseteq$) қатынасы бойынша аракідік реттелген жиын болады және $\boldsymbol{T}$ әліпбиіндегі сөздердің барлық ішжиындары $\subseteq$ ($\supseteq$) бойынша салыстырымды.

I.2.4.3-анықтама. Мейлі $\boldsymbol{L}\subseteq \boldsymbol{T^*}$. Сонда:

1. $\boldsymbol{L}^{\mathrm{R}}$ – $\boldsymbol{L}$ тілінде анықталған *керілеу* амалы, ол былай анықталады $\boldsymbol{L}^{\mathrm{R}} = \{\tau^{\mathrm{R}}: \tau\in\boldsymbol{L}\}$.

2. Pref($\boldsymbol{L}$) – $\boldsymbol{L}$ тілінің *барлық префикстер жиыны*, ол былай анықталады Pref($\boldsymbol{L}$) $\rightleftharpoons \{\tau: \exists\xi(\xi\in\boldsymbol{L}\&\ \tau\sqsubset\xi)\}$, $\sqsubset$ - префикс қатынасы.

3. Suf($\boldsymbol{L}$) – $\boldsymbol{L}$ тілінің *барлық постфикстер жиыны*, ол былай анықталады Suf($\boldsymbol{L}$) $\rightleftharpoons \{\zeta: \exists\xi\in\boldsymbol{L}(\zeta\sqsupset\xi)\}$, $\sqsupset$ - постфикс қатынасы..

4. Substr($\boldsymbol{L}$) – $\boldsymbol{L}$тілінің *барлық іштізбелер жиыны*, ол былай анықталады Substr($\boldsymbol{L}$) $\rightleftharpoons \{\tau: \forall\xi(\xi\in\boldsymbol{L}\&\ \xi\neq\varepsilon \Rightarrow \tau\subseteq\xi)\}$.

5. Егер $\boldsymbol{f}$: $\boldsymbol{K}\rightarrow\boldsymbol{L}$ функциясына қатысты $\boldsymbol{L}$ жиынының әрбір элементі $\boldsymbol{K}$ жиынының бір ғана элементінің образы болса, $\boldsymbol{f}$ функциясы *биекция* деп аталады,.

I.2.4.3-мысалдар: Мейлі $\boldsymbol{L}$ тілі берілсін:

1. $\{a, b\}$ әліпбиінде $\boldsymbol{L} = \{a^m ba^n: m\leq n\}$, онда оның барлық іштізбелер жиыны мына тізбелерді *b*, *ba*, *aba*, *baa*, *abaa*, *baaa*, *aabaa*, *abaaa* және т.б. қамтиды.

2. $\{a, b\}$ әліпбиінде $\boldsymbol{L} =\{ ab^n: n\geq 0\}$, оның барлық іштізбелер жиыны мыналарды *a*, *ab*, *abb*, *abbb*, *abbbb*, *abbbbb* және т.б. қамтиды.

3. $\{a, b, c\}$ әліпбиінде $\boldsymbol{L} = \{abc, a\}$, онда оның барлық іштізбелер жиыны былай анықталады Substr($\boldsymbol{L}$) $\rightleftharpoons \{\varepsilon, a, b, c, ab, ac, bc, abc\}$.

I.2.4-тапсырмалар:

1. Мейлі $\boldsymbol{T}=\{a, b\}$ және $\boldsymbol{L} = \{aa, ab\}$ берілсін. $\boldsymbol{L}^3$ табыңыз.
2. $\boldsymbol{L}^* \neq \{x^n: x\in \boldsymbol{L}, n\geq 0\}$ орындалатын $\boldsymbol{L}$ тілін табыңыз.
3. $\boldsymbol{L}^{\boldsymbol{R}})^* \neq (\boldsymbol{L}^*)^{\boldsymbol{R}}$ орындалатын $\boldsymbol{L}$ тілін табыңыз.
4. Мейлі $\boldsymbol{L_1}=\{(ab)^n: n\geq 0\}$ және $\boldsymbol{L_2}=\{a^m b^m: m\geq 0\}$. Сонда $\boldsymbol{L_1}\cap\boldsymbol{L_2}$ тілінің барлық тізбелерін көрсетіңіз
5. Мейлі $\boldsymbol{T}=\{a, b, c\}$. Сонда $\boldsymbol{L_1}=\{(abc)^n a: n\geq 2\}$ тілі мен $\boldsymbol{L_2}=\{ab(cab)^n ca: n\geq 1\}$ тең екендігін көрсетіңіз.
6. Мейлі $\boldsymbol{T}=\{a, b, c\}$. $\boldsymbol{L_1}=\{\tau\in \boldsymbol{T}^*: |\tau| = 4\}$ және $\boldsymbol{L_2}=\{\tau\in \boldsymbol{T}^*: |\tau|_c = 1\}$. Сонда $\boldsymbol{L}_1\setminus \boldsymbol{L}_2$ тілінің элементтерінің санаңыз.
7. Мына тілдердің $\boldsymbol{L}_1\cdot(\boldsymbol{L}_2^{\boldsymbol{R}})$, $\boldsymbol{L}_1^{\boldsymbol{R}}\cdot\boldsymbol{L}_2$ тең қуаттылығын анықтаңыз.
8. Мына тілдер үшін $\boldsymbol{L}_1= \{d, de, dee\}$, $\boldsymbol{L}_2= \{\varepsilon, d, e, de, d\}$ олардың конкатенациясы мен бірігуінің нәтижесін табыңдар.
9. Егер $\boldsymbol{L}=\{dcc, dcd, ddc, ddd\}$ болса, онда $\{\gamma: \gamma = \mathrm{d}\xi$ және $|\gamma| \geq\} \cup\{\varepsilon\}$ итрация $\boldsymbol{L}^*$ арқылы беріле ме?
10. Мейлі $\{a, b, c, d\}$ әліпбиінде $\boldsymbol{L} = \{abcd, ad\}$ тілі берілсін. Осы тілдің барлық ішсөздерінің жиыны Substr($\boldsymbol{L}$) табыңдар.
11. Мейлі $\{a,b,c\}$ әліпбиінде. $\boldsymbol{L}= \{a^k b^m c^n: k\leq m\leq n\}$ тілі берілсін. Осы тілдің барлық ішсөздерінің жиыны Substr($\boldsymbol{L}$) табыңдар.

I.2.4-сұрақтар:

1. Барлық тізбелердің жиыны қалай анықталады?
2. Бос емес тізбелердің жиыны қалай анықталады?
3. Берілген әліпбидегі тіл дегент не?
4. $\boldsymbol{L_1}\times\boldsymbol{L_2}$ қалай анықталады?
5. $\overline{L}$ қалай анықталады?
6. $\boldsymbol{L_1} \setminus \boldsymbol{L_2}$ қалай анықталады?
7. $\boldsymbol{L_1}\triangle\boldsymbol{L_2}$ қалай анықталады?
8. $\boldsymbol{L}^*$ қалай анықталады?
9. $\boldsymbol{L}^{\mathrm{R}}$ қалай тұрғызылады?
10. Pref($\boldsymbol{L}$) қалай алынады?
11. Suf($\boldsymbol{L}$) қалай алынады?

I.3. Тілдерді анықтау механизмдері

I.3.1. Тілдерді тудырушы механизмдер

Бұл параграфта формалды граматика – тілдерді тудырушы механизмдер, шығарымдылық қатынасы және формалды грамматикамен тудырылатын тілдердің анықтамалары қарастырылады, мысалдар ұсынылады, тапсырмалар мен сұрақтар қойылады [1-9,11-18,21,25,27-32].

Тілді тудырушы механизмдердің маңызды тобын алғаш 1959 жылы американдық лингвист Н.Хомский (N.Chomsky) ендірген формалды грамматикалар құрайды [24].

L тілін тудыратын формалды грамматикада қиылыспайтын екі символдар жиыны пайдаланылады:

1) ***L*** тілінің тізбелері құрастыралатын терминал символдарының (тұрақты шамалардың) ақырлы жиыны ***T***;

2) ***T*** жиынымен қиылыспайтын және ***L*** тілінің грамматикалық түсініктерін, категорияларын және т.б. белгілейтін бейтерминал (айнымалы шамалардың) ақырлы жиыны ***N***.

3) ***L*** тілінің тізбелерінің туындау үдерісі әрқайсысы (α,β) тізбелер жұбы болатын шығарым ережелерінің ақырлы жиыны ***P*** арқылы сипатталады. Мұндай жұптағы бірінші компонента α тізбесінің құрамында ең болмағанда бір бейтерминал бар кез келген тізбе, ал екінші компонента β терминал және/немесе бейтерминал символдарынан құралған кез келген тізбе. Ол бос тізбе де бола алады.

I.31-келісім. Формалды грамматикаға байланысты әртүрлі символдар мен тізбелер туралы келесі келісімдерді қабылданады:

(1) курсив кіші латын әріптері a, b,…, z мен араб цифрлары 0, 1, ..., 9 арқылы терминалдарды белгілейді;

(2) курсив үлкен латын әріптері A, B, C, …, X, Y, Z бейтерминалдарды белгілейді, мұнда S – бастапқы бейтерминал символ;

(3) кіші грек әріптері α, β,..., ω тізбелерді белгілейді, мұнда ε – бос тізбе;

(4) **P** жиынындағы (α, β) жұбы болатын шығарым ережесі мына түрде $\alpha \rightarrow \beta$ жазылады;

(5) бұл келісімдер төменгі және жоғарғы индекстері бар әріптерге де таратылады;

(6) $\alpha_1|\alpha_2|\ldots|\alpha_m \rightarrow \beta$ түріндегі шығарым ережесі келесі m шығарым ережелерінің қысқаша түрі:

$$\begin{gathered} \alpha_1 \rightarrow \beta \\ \alpha_2 \rightarrow \beta \\ \ldots \\ \alpha_m \rightarrow \beta \end{gathered}$$

(7) $\alpha \rightarrow \beta_1|\beta_2|\ldots|\beta_n$ түріндегі шығарым ережесі келесі $\boldsymbol{n}$ шығарым ережелерінің қысқаша түрі:

$$\begin{gathered} \alpha \rightarrow \beta_1 \\ \alpha \rightarrow \beta_2 \\ \ldots \\ \alpha \rightarrow \beta_n \end{gathered}$$

(8) $\alpha_1|\alpha_2|\ldots|\alpha_{\mathrm{m}} \rightarrow \beta_1|\beta_2|\ldots|\beta_n$ түріндегі шығарым ережесі (6) және (7) келісімдерінен алынған $\boldsymbol{m} \times \boldsymbol{n}$ шығарым ережелерінің қысқаша түрі.

I.3.1.1-анықтама. Формалды грамматика деп мынадай төрттікті $\boldsymbol{G} = <\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{P}, S>$ айтады, мұнда:

$\boldsymbol{T}$ – терминалдардың бос емес ақырлы жиыны;

$\boldsymbol{N}$ – бейтерминалдардың бос емес ақырлы жиыны және $\boldsymbol{T} \cap \boldsymbol{N} = \boldsymbol{\emptyset}$;

$\boldsymbol{P}$ – $\alpha \rightarrow \beta$ түріндегі шығарым ережелерінің бос емес ақырлы жиыны, мұнда $\alpha \in (\boldsymbol{T} \cup \boldsymbol{N})^* \times \boldsymbol{N} \times (\boldsymbol{T} \cup \boldsymbol{N})^*$, $\beta \in (\boldsymbol{T} \cup \boldsymbol{N})^*$, яғни $\boldsymbol{P} \subseteq \{(\alpha, \beta): \alpha \in (\boldsymbol{T} \cup \boldsymbol{N})^* \times \boldsymbol{N} \times (\boldsymbol{T} \cup \boldsymbol{N})^* \& \beta \in (\boldsymbol{T} \cup \boldsymbol{N})^*\}$;

S – бастапқы бейтерминалы, $S \in \boldsymbol{N}$.

Грамматиканың шығарым ережелерін элементарлық амал деп қарастыруға болады, оларды алғашқы тізбеге белгілі бір ретпен қолданған кезде тек дұрыс тізбелер туындайды. Қандай да бір тізбені тудыру үдерісінде қолданатын шығарым ережелерінің тізбегі осы тізбенің шығарымы деп аталады.

I.3.1.1-мысалдар: Егер $\boldsymbol{G} = <\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{P}, S>$ грамматикасы мыналармен $\boldsymbol{N} = \{S, E, V, F\}$, $\boldsymbol{T} = \{+, -, /, *, (,), a\}$ және ережелер $\boldsymbol{P} = \{S \to E, E \to E+V \mid E-V \mid V, V \to V*F \mid V/F \mid F, F \to a \mid (E)\}$ берілсе, онда ол жақшалы алгебралық өрнекті тудырады.

Грамматика арқылы тілді анықтау үшін *шығарымды тізбе* ұғымын және *тікелей шығарымдылық қатынасын* пайдаланады.

I.3.1.2-анықтама:

1. $\boldsymbol{G} = <\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{P}, S>$ грамматикасының *шығарымды тізбелері* рекурсивті былай анықталады:

1) S – $\boldsymbol{G}$ грамматикасының *шығарымды тізбесі*;

2) Егер $\alpha\beta\gamma$ – $\boldsymbol{G}$ грамматикасының *шығарымды тізбесі* және $\boldsymbol{P}$ құрамында $\beta \to \sigma$ шығарым ережесі болса, онда $\alpha\sigma\gamma$ – $\boldsymbol{G}$ грамматикасының *шығарымды тізбесі*;

2. Құрамында $\boldsymbol{N}$ жиынының бейтерминал символдары жоқ $\boldsymbol{G}$ грамматикасының шығарымды тізбесі $\boldsymbol{G}$ грамматикасымен туындаған *терминалдық тізбе* деп аталады.

3. Егер $\alpha = \gamma\xi\delta$, $\beta = \gamma\eta\delta$ және $\alpha \to \beta$, $\xi \to \eta$ $\boldsymbol{G}$ грамматикасының шығарым ережелері болса, онда α мен β тізбелері арасында *тікелей шығарымдылық қатынасы* орнатылды деп айтады және ол $\boldsymbol{G}$ грамматикасында β тізбесі α тізбесінен ξ-ді η-ге ауыстырғаннан тікелей шығатындығын білдіреді де, ол қатынас $\alpha \Rightarrow_G \beta$ түрінде белгіленеді. Егер грамматика алдын ала белгілі болса, онда тікелей шығу қатынасындағы көрсеткіш $\boldsymbol{G}$ көрсетілмейді және ол былай жазылады $\alpha \Rightarrow \beta$.

Мына $\Rightarrow^k$ белгілеу арқылы $\Rightarrow$ тікелей шығарымдылық қатынасының k-шы *дәрежесін* белгілейміз, егер $1 \le i \le k$ және $\alpha = \alpha_0$, $\alpha_k = \beta$ үшін $\alpha_{i-1} \Rightarrow \alpha_i$ $(1 \le i \le k)$ шарттары орындалатын $k+1$ тізбелері $\alpha_0, \alpha_1, ..., \alpha_k$, бар болса, онда $\alpha \Rightarrow^k \beta$ қатынасы орындалады. Бұл тізбелер қатары $\boldsymbol{G}$ грамматикасындағы α тізбесінен *ұзындығы* k болатын β тізбесінің *шығуы* деп аталады.

Егер кез келген $i \ge 1$ (немесе $i \ge 0$) үшін $\alpha \Rightarrow^i \beta$ қатынасы орындалса, онда оны $\alpha \Rightarrow^+ \beta$ (немесе $\alpha \Rightarrow^* \beta$) деп жазуға болады.

Мұнда $\Rightarrow^{+}$ арқылы $\Rightarrow$ қатынасының транзитивтік тұйықталуын, ал $\Rightarrow^{*}$ арқылы $\Rightarrow$ қатынасының рефлексивтік және транзитивтік тұйықталуын белгілейміз және мына жазба $\varphi\Rightarrow^{+}\psi$ ($\varphi\Rightarrow^{*}\psi$) былай оқылады: «$\psi$ *тізбесі* φ*-ден қарапайым емес жолмен шығады*» («ψ *тізбесі* φ*-ден шығады*»).

I.3.1-ескертпе. $\alpha\Rightarrow^{*}\beta$ тек қана және тек қана сонда, қашан қандай да бір $i\geq 0$ үшін $\alpha\Rightarrow^{i}\beta$ болса, және $\alpha\Rightarrow^{+}\beta$ тек қана және тек қана сонда, қашан қандай да бір $i\geq 1$ үшін $\alpha\Rightarrow^{i}\beta$ болса.

I.2.1.4-анықтамалар:

1. Грамматиканың алғашқы бейтерминалынан шығатын әрбір тізбе *сентенциалдық пішім* деп аталады.

2. Бейтерминал символдарды қамтымайтын шығарылатын тізбелер *терминалды тізбелер* деп аталады. Сондықтан $\boldsymbol{L}(\boldsymbol{G})$ тілін $\boldsymbol{G}$ грамматикасында шығарылған терминалды тізбелер жиыны деп анықтауға болады..

3. $\boldsymbol{P}$ жиынындағы шығарым ережелерін қолдану арқылы S бастапқы бейтерминалдан шығатын терминалдардық тізбелер жиынын $\boldsymbol{G}$ грамматикасымен туындаған $\boldsymbol{L}(\boldsymbol{G})$ *тілі* деп атайды,

$$\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{\tau\colon \tau \in \boldsymbol{T}^*,\ S\Rightarrow^*\tau\}.$$

Бұл $\boldsymbol{L}(\boldsymbol{G})$ тіліне кіретін әрбір тізбе сентенциалдық пішім болатындығын білдіреді.

I.3.1.2-мысалдар:

1.$\boldsymbol{G_1} = \langle \boldsymbol{T}, \boldsymbol{N}, \boldsymbol{P}, S\rangle$ грамматикасы берілсін, мұнда $\boldsymbol{T}=\{0,1\}$ – терминалдық жиын, $\boldsymbol{P} = \{S\rightarrow 0A1,\ 0A\rightarrow 00A1,\ A\rightarrow\varepsilon\}$ – шығарым ережелерінің жиыны, $\boldsymbol{N}=\{A,\ S\}$ – бейтерминалдық жиын. Егер $S\Rightarrow 0A1\Rightarrow 00A11\Rightarrow 0011$ шығарымын қарастырсақ, онда осы шығарымның бірінші қадамында S бастапқы бейтерминал $S\rightarrow 0A1$ ережесі бойынша $0A1$ тізбесіне, екінші қадамда $0A$ тізбесі $0A\rightarrow 00A1$ ережесі бойынша $00A1$ тізбесіне, ал үшінші қадамда A бейтерминалы ε бос тізбесіне ауысқандығы байқалады. Сонымен,

$S \Rightarrow^3 0011$, $S \Rightarrow^+ 0011$, $S \Rightarrow^* 0011$ және 0011 тізбесі $\boldsymbol{L}(\boldsymbol{G_1}) = \{0^n1^n: n>1\}$ тіліне жатады деп айтуға болады.

2. Ережелері $\boldsymbol{P_1} = \{S\to 01S,\ S\to 0\}$ болатын және ережелері $\boldsymbol{P_2} = \{S\to 0A,\ A\to 10A,\ A\to\varepsilon\}$ болатын екі грамматика эквивалентті.

3. Операндтары i, n және амалдары +, * болатын алгебралық өрнектерді тудыратын бірдей терминалдары $\boldsymbol{T} = \{i,\ n,\ (,\ ),\ +,\ *\}$, бірдей бейтерминалдары $\boldsymbol{N} = \{S, F, H\}$ және бөлек ережелері

$$\boldsymbol{P_1} = \{S\to S+F,\ S\to F,\ F\to F*H,\ F\to H, F\to H,\ H\to i,\ H\to n,\ H\to (S)\},$$

$$\boldsymbol{P_2} = \{S\to S+F|F,\ S\to S+F|S*F|F,\ F\to F*H|H,\ H\to i|n|(S)\}$$

бар екі грамматика эквивалентті болады.

I.3.1-тапсырмалар:

1. Ережелері мынадай

$$S\to A+B|B+A;\ A\to a;\ B\to b$$

грамматиканың барлық сентенциалдық пішімдерін табызыңыз.

2. Ережелері мынадай

$$S\to E|E+S|E\text{-}S;\ E\to F|F*E;\ F\to a|b$$

грамматика үшін берілген $a\text{-}b*a+b$ тізбесінің шығарымын тұрғызыңыз.

3. Ережелері мынадай

$$S\to aSBC|abC;\ CB\to BC;\ bB\to bb;\ bC\to bc;\ cC\to cc$$

грамматика үшін берілген $aaabbbccc$ тізбесінің шығарымын тұрғызыңыз.

4. Ережелері мынадай

$$S\to FF;\ F\to aFb;\ F\to ab$$

грамматикамен туатын тілді сипаттаңыз.

5. Ережелері мынадай

$$S\to Sc;\ S\to A;\ A\to aAb;\ A\to \varepsilon$$

грамматикамен туатын тілді сипаттаңыз.

6. Ережелері мынадай

$$S\to\varepsilon;\ S\to a;\ S\to b;\ S\to aSa;\ S\to bSb$$

грамматикамен туатын тілді сипаттаңыз.

7. Ережелері мынадай $S\to SA$; $SAA\to ASb$; $ASA\to b$; $A\to a$ грамматикамен туатын тілді сипаттаңыз.

8. Ережелері мынадай $S\to aSA$; $S\to abc$; $bA\to bbc$; $cA\to Aa$

грамматикамен туатын тілді сипаттаңыз

9. Ережелері мынадай $S \to aAS$; $S \to B$; $Aa \to aaA$; $AB \to B$; $B \to a$ грамматикамен туатын тілді сипаттаңыз

I.3.1-сұрақтар:

1. Ережелері $S \to ab$; $S \to aKSb$; $K \to bSb$; $KS \to b$; $K \to \varepsilon$

және

$$S \to aAb;\ A \to \varepsilon;\ A \to b;\ A \to S;\ A \to bSbS$$

екі грамматика эквивалентті ме?

2. Ережелері $S \to aD$; $D \to bba$; $D \to baDa$; $D \to aDaDa$

және

$S \to aaE$; $S \to abD$; $E \to bDD$; $D \to aaEa$; $D \to abDa$; $D \to ba$

екі грамматика эквивалентті ме?

3. Ережелері мынадай

$$S \to abba;\ S \to baa$$

грамматика қандай топқа жатады?

4. Ережелері мынадай

$$S \to AD;\ A \to aA;\ A \to \varepsilon;\ D \to bDc;\ D \to d$$

грамматика қандай топқа жатады?

5. Мынадай ережелермен

$$S \to AB,\ A \to a|Aa,\ A \to a|Aa$$

және

$$S \to AS|SB|AB,\ A \to a;\ B \to b?$$

берілген екі грамматика эквивалентті ме?

6. Мына ережелермен

$$S \to cE;\ E \to ddc;\ E \to dcEc;\ E \to cEcEc$$

және

$$S \to ccA;\ S \to cdB;\ A \to dBB;\ B \to ccAc;\ B \to cdBc;\ B \to dc?$$

берілген екі грамматика эквивалентті ме?

7. Көпмәнді грамматикамен туындыйтын $E \to E+E|E*E|(E)|i$ тілін бірмәнді грамматика көмегімен қалай сипаттауға болады?

I.3.2. Тілдер топтары және алгоритмдік проблемалар

Бұл параграфта тілдердің топтары мен типтері және олардың алгоритмдік проблемалары қарастырылады, мысалдар беріледі, тапсырмалар ұсынылады, сұрақтар қойылады [1-9,11-18,21,25-32].

I.3.2.1-анықтама. Формалды грамматикалардың шығарым ережелеріне шектеулер қойып әртүрлі тілдер тобын тудыруға болады:

1. Шығарым ережелеріне ешқандай шектеу қойылмайтын кез келген грамматика 0-ші топқа жатады және шектеусіз грамматика– unrestr*i*cted grammar (ШГ) деп аталады, ал одан туындаған тізбелер жиыны *рекурсивті-санаулы тіл* болады.
2. Барлық шығарым ережелері $\alpha\rightarrow\beta$ түрінде болған грамматикада мынадай шектеулік $\alpha = \xi H\zeta$, $\beta=\xi\eta\zeta$, $\xi\in(\boldsymbol{T}\cup\boldsymbol{N})^*$, $H\in\boldsymbol{N}$, $\eta\in(\boldsymbol{T}\cup\boldsymbol{N})^+$, $\zeta\in(\boldsymbol{T}\cup\boldsymbol{N})^*$ қойылса 1-ші топ болады және *контексті* (*контексті-тәуелді– context-sensitive*) *грамматика* (КГ) деп аталады, ал одан туындаған тізбелер жиыны *контексті- тәуелді тіл* болады.
3. Барлық шығарым ережелері $A\rightarrow\alpha$ түріндегі грамматикада $A\in\boldsymbol{N}$, $\alpha\in(\boldsymbol{T}\cup\boldsymbol{N})^*$ шектеулігі қойылса, ол 2-ші топқа жатады және *контексті-бос грамматика – contex-free grammar* деп аталады, ал одан туындаған тізбелер жиыны *контексті-бос тіл* болады.
4. Барлық шығарым ережелері $A\rightarrow\alpha B\beta$ немесе $A\rightarrow\alpha$, мұнда $A,B\in\boldsymbol{N}$, $\alpha,\beta\in\boldsymbol{T}^*$ түрінде болған грамматика 2-ші топқа жатады және ол *сызықты грамматика – linear grammar* (СГ) деп аталады. Сызықты грамматикада, егер $\beta=\varepsilon$, онда ол оң сызықты *грамматика – right-linear grammar*, ал егер $\alpha=\varepsilon$, онда ол сол сызықты *грамматика – left-linear grammar* болады. (*Оң/сол*) *сызықты грамматикамен* туындаған тізбелер жиыны – (*оң/сол*) *сызықты тіл*.

I.3.2-ескертпелер:

1. Кейбір әдебиеттерде контексті-тәуелді тілдер контексті грамматика (КГ) немесе қысқартылмаған грамматика немесе тікелей құрылғыштар грамматикасы деп аталады, ал контексті-бос грамматиканы контекстісіз грамматика (КГ) деп атайды.

2. Әрбір сызықтық грамматика контекстісіз грамматика болады.

3. Әрбір констекстісіз грамматика контексті грамматика болады.

4. Әрбір констексті грамматика шенеуленбеген грамматика болады.

5. Кез келген сызықтық тіл контекстісіз тілдің меншікті ішжиыны болады, бірақ контекстісіз тіл сызықтық болмауы мүмкін.

6. Бос тізбені қамтымайтын кез келген контекстісіз тіл контексті тілдің меншікті ішжиыны болады.

7. Кез келген констексті тіл рекурсивті-санаулы тілмен қамтылады.

8. Егер $\boldsymbol{L}_0$, $\boldsymbol{L_1}$, $\boldsymbol{L_2}$, $\boldsymbol{L_3}$ тілдері 0, 1, 2, 3 типті грамматикалармен сәйкес туындайтын болса, онда $\boldsymbol{L_3} \subseteq \boldsymbol{L_2} \subseteq \boldsymbol{L_1} \subseteq \boldsymbol{L}_0$ ақиқат.

I.3.2.1-мысалдар:

1. Мына $S\to aBa$; $B\to aBCa$; $B\to b$; $bC\to BB$; $aC\to Ca$ ережелері бар грамматикамен туындаған $a^n b^n c^n$ тілі рекурсивті-санаулы болады. Мұнда ережелерді кез-келген ретте орындауға болады, тек міндетті түрде $S \to aBC$ ережесін орындау керек (n-ді бекіту үшін), ал $bC \to bc$ ережесін C-ның оң жағында бір де бір B қалмағаннан кейін ғана қолдану керек (әйтпесе бұл B терминал b-ға ауыспайды және шешім терминал тізбемен аяқталды).

2. Мына a, b, c айнымалыларымен берілген булевтік формулалар жиыны контекстісіз тіл болып табылады, себебі ол мынадай грамматикамен туындайды $\boldsymbol{N}=\{S, F\}$, $\boldsymbol{T}=\{a, b, c, \neg, \vee, \wedge, (,)\}$ және $\boldsymbol{P} = \{S\to \neg S, S\to S \wedge F, S\to S \vee F, S\to F, F\to a|b|c, F\to (S)\}$.

3. Мейлі $\boldsymbol{G} = <\{S\}, \{a, b\}, \{S\to aSa, S\to b\}, S>$ болса. Сонда $aSa\Rightarrow^3 aaaaSaaaa$.

4. Егер $\boldsymbol{G} = <\{S\}, \{a,b\}, \{S\to aSa, S\to bb\}, S>$ болса, онда $\boldsymbol{L}(\boldsymbol{G})=\{a^n bba^n : n\geq 0\}$

5. Мына грамматикамен $\boldsymbol{N}=\{S\}$; $\boldsymbol{T}=\{a\}$; $\boldsymbol{P}=\{S\to a;\ S\to aaS\}$ туындаған $\{a^{2n\text{-}1}\}$ тілі оң сызықты тіл болады.

6. Мына грамматикамен $\boldsymbol{N}=\{S\}$; $\boldsymbol{T}=\{a\}$; $\boldsymbol{P}=\{S\to a;\ S\to Saa\}$ туындаған $\{a^{2n\text{-}1}\}$ тілі сол сызықты тіл болады.

7. Мына $S\to aSa$; $S\to Q$; $Q\to bQ$; $Q\to \varepsilon$ ережелері бар грамматика с сызықты, бірақ оң сызықты болмайды, себебі бірінші ереже қажетті түрде емес.

8. Мейлі $\boldsymbol{T} = \{t_1, t_2,\ldots, t_n\}$ әліпбиі берілсін. Сонда $\boldsymbol{T^*}$ оң сызықты болады, себебі ол мына $S \to \varepsilon$; $S \to t_1S$; $S \to t_2S$; ...; $S \to t_nS$ ережелері бар грамматикамен туындайды.

9. Мына $S \to QQ$; $Q \to cQQ$; $Q \to bQ$; $S \to a$ ережелері бар грамматика контекстісіз, бірақ сызықты болмайды, себебі бірінші және ереженің оң жағында артық бейтерминал бар, яғни олар қажетті түрде емес.

10. Мына $S \to QS$; $S \to US$; $S \to b$; $Qb \to Ab$; $A \to a$; $QA \to AAQ$; $UAb \to b$; $UAAA \to AAU$ ережелері бар грамматика контекстісіз, бірақ сызықты болмайды, себебі соңғы үш ереже қажетті түрде емес.

11. Мына $S \to ASQA$; $S \to AbA$; $A \to a$; $bQ \to bb$; $AQ \to UQ$; $UQ \to UV$; $UV \to QV$; $QV \to QA$ ережелері бар грамматика контексті, бірақ контекстісіз болмайды, себебі соңғы бес ережелердің сол жағында екі бейтерминал бар.

I.3.2.2-анықтамалар:

1. Егер $\boldsymbol{G}$ грамматикасымен туындайтын $\boldsymbol{L}(\boldsymbol{G})$ тілі терминал символдарынан тұратын ешқандай ақырлы тізбе (сөз) қамтымайтын болса, онда ол *бос тіл* деп аталады, яғни $\boldsymbol{L}(\boldsymbol{G}) = \varnothing$.

2. $\boldsymbol{L}(\boldsymbol{G})$ тілі бос емес болуы үшін ең кемінде $\xi \to \omega$ түрінде бір ереже және $S \Rightarrow^* \xi$ шешімі бар болуы керек, мұнда $S \in \boldsymbol{N}$ – бастапқы бейтерминал, $\xi \in (\boldsymbol{T} \cup \boldsymbol{N})^* \times \boldsymbol{N} \times (\boldsymbol{T} \cup \boldsymbol{N})^*$, $\boldsymbol{\omega \in T^*}$.

3. Егер берілген $\boldsymbol{G}$ грамматиканың шығарым ережелері тұйықталатын қайталаным құрса, онда мұндай грамматика *шексіз тіл* тудырады, яғни $\boldsymbol{L}(\boldsymbol{G}) = \infty$;

4. Егер кез келген ξ тізбесі және берілген $\boldsymbol{G}$ грамматикасы үшін $\xi \in \boldsymbol{L}(\boldsymbol{G})$ орындалса, онда ξ тізбесі $\boldsymbol{L}(\boldsymbol{G})$ тіліне тиісті болады;

5. Егер кез келген екі грамматика $\boldsymbol{G_i}$ және $\boldsymbol{G_j}$ ($\boldsymbol{i \neq j}$) үшін $\boldsymbol{L}(\boldsymbol{G_i}) = \boldsymbol{L}(\boldsymbol{G_j})$ болса, онда $\boldsymbol{G_i}$ және $\boldsymbol{G_j}$ грамматикалары *эквивалентті*. Басқаша айтқанда, егер екі грамматика $\boldsymbol{G_i}$ және $\boldsymbol{G_j}$ бір ғана тілді $\boldsymbol{L}(\boldsymbol{G_i}) = \boldsymbol{L}(\boldsymbol{G_j})$ тудырса, онда олар эквивалентті болады.

I.3.2.2-мысалдар:

1. $S \to Q$, $U \to abba$ ережелері бар грамматика $\varnothing$ тілін тудырады;

2. $S \to aS$ ережесі бар ∞ тілін тудырады;

3. $S\rightarrow aU$, $U\rightarrow baU$, $U\rightarrow\varepsilon$ ережесі бар грамматика мен $S\rightarrow abS$, $S\rightarrow a$ ережесі бар грамматика эквивалентті болады.

Жоғарыдағыларды ескере отырып *грамматикалардың* мынадай *алгоритмдік проблемаларын* қарастыруға болады:

1. *Бостық проблемасы* – берілген $\boldsymbol{G}$ грамматикасы үшін $\boldsymbol{L}(\boldsymbol{G})$ бос тіл бола ма, яғни $\boldsymbol{L}(\boldsymbol{G}) = \varnothing$?

2. *Тиістілік проблемасы* – кез келген ξ тізбе берілген $\boldsymbol{G}$ грамматикасынан туындайтын $\boldsymbol{L}(\boldsymbol{G})$ тіліне тиісті бола ма, яғни $\xi\in\boldsymbol{L}(\boldsymbol{G})$?

3. *Эквиваленттілік проблемасы* – кез келген екі $\boldsymbol{G_i}$ және $\boldsymbol{G_j}$ ($\boldsymbol{i\neq j}$) грамматикалар эквивалентті бола ма, яғни $\boldsymbol{L}(\boldsymbol{G_i})=\boldsymbol{L}(\boldsymbol{G_j})$?

4. *Тұйықтылық проблемасы* – белгілі бір типтегі кез келген екі тілге жиындық амалдарды қолданғанда нәтижесі сол топқа жататындығын анықтау.

5. *Шексіздік проблемасы* – берілген $\boldsymbol{G}$ грамматикасы үшін $\boldsymbol{L}(\boldsymbol{G})$ шексіз тіл бола ма, яғни $\boldsymbol{L}(\boldsymbol{G}) = \infty$?

І.3.2-тапсырмалар:

1. $\{a^m b^n c$: $\tau\in\{a,b\}^*$, $m\geq 0$, $|\tau|_b=2\}$ тілін тудыратын сызықты грамматиканы табыңыз.

2. $\{a^n\tau$: $n\geq 1$, $m\geq 1\}$ тілін тудыратын сызықты грамматиканы табыңыз.

3. $\{a,b\}^*$ - $a^n b^n c^n$: $n\geq 0\}$ тілін тудыратын сызықты грамматиканы табыңыз.

4. $\{\varphi a\psi\xi b$: $\varphi\in\{a,b\}^*$, $\psi\in\{a,b\}^*$, $|\varphi|=|\psi|\}$ тілін тудыратын сызықты грамматиканы табыңыз.

5. $\{a,b,c\}^*$ - $\{\tau c\tau$: $\tau\in\{a,b\}^*\}$ тілін тудыратын сызықты грамматиканы табыңыз.

6. Мына $S\rightarrow KbbaK$; $K\rightarrow Ka$; $K\rightarrow Kb$; $S\rightarrow\varepsilon$ оң сызықты грамматикаға
эквивалентті грамматиканы табыңыз.

7. Мына $S\rightarrow aSb$; $S\rightarrow K$; $S\rightarrow J$; $K\rightarrow aK$; $J\rightarrow Jb$; $K\rightarrow a$; $J\rightarrow\varepsilon$ оң сызықты грамматикаға эквивалентті грамматиканы табыңыз.

I.3.2-сұрақтар:

1. $L(G)^{R}$ тілі G грамматикасының ережелер саны қанша болса, сонша болатын бір де бір оң сызықты грамматикамен тумайтындай G грамматикасы бар ма?

2. Мынадай ережелері бар грамматика қандай топқа жатады:

a) $S \to a|Ba$, $B \to Bb|b$?

b) $S \to Ab$, $A \to Aa|ba$?

c) $S \to 0A1|01$, $0A \to 00A1$, $A \to 01$?

d) $S \to AB$, $AB \to BA$, $A \to a$, $B \to b$?

e) $S \to aC$; $S \to bD$; $C \to \varepsilon$; $D \to \varepsilon$?

3. Ережелері мынадай $S \to AS|SB|AB$, $A \to a$, $B \to b$ және $S \to AB$, $A \to a|Aa$, $B \to b|Bb$ болатын екі грамматика эквивалентті бола ма?

4. Мына ережелермен туындайтын тіл қай топқа жатады:

1) $S \to CD$, $C \to aCA$, $C \to bCB$, $AD \to aD$, $BD \to bD$, $Aa \to aA$, $Ab \to bA$, $Ba \to aB$, $Bb \to bB$, $C \to \varepsilon$, $D \to \varepsilon$?

2) $S \to AB$, $A \to Aa|bB$, $B \to a|Sb$?

3) $S \to aSBC|abC$, $CB \to BC$, $bB \to bb$, $bC \to bc$, $cC \to cc$?

4) $S \to aS$, $S \to aA$, $A \to bA$, $A \to bZ$, $Z \to \varepsilon$?

5) $S \to Ab$, $A \to Ab$, $A \to Za$, $Z \to \varepsilon$?

6) $S \to aB$, $B \to aB$, $B \to b$?

7) $S \to AB$, $A \to a$, $A \to ac$, $B \to b$, $B \to cb$?

8) $S \to Aa$, $A \to Bb$, $A \to \varepsilon$, $B \to \varepsilon$?

5. Ережелері мынадай грамматикамен туындаған тіл қалай сипатталады:

1) $S \to ASHA$, $S \to AbA$, $A \to a$, $bH \to bb$, $AH \to HA$?

2) $S \to aT\,bc$, $Tb \to bT$, $Tc \to Ubcc$, $bU \to Ub$, $aU \to aaT$?

3) $S \to aSH$, $S \to aR$, $RH \to bRG$, $GH \to HG$, $R \to bG$, $G \to c$?

4) $S \to aSH$, $S \to abc$, $bH \to bbc$, $cH \to Hc$?

5) $S \to aSA$, $S \to abcB$, $Ab \to abb$, $cB \to Ab$?

6) $S \to AD$, $A \to aA$, $A \to \varepsilon$, $D \to bDc$, $D \to \varepsilon$?

7) $S \to \varepsilon$, $S \to a$, $S \to b$, $S \to aSa$, $S \to bSb$?

8) $S \to SA$, $SAA \to ASb$, $ASA \to b$, $A \to a$?

9) $S \to aSb$, $S \to ab$?

І.3.3. Тілдерді танушы механизмдер

Бұл параграфта тілді танушы механизмдерінің (автоматтардың) формалды емес анықтамасы беріледі, олардың құрамы, кескіні және жұмысы сипатталады, сонымен қатар, мысалдар ұсынылады, тапсырмалар мен сұрақтар қойылады [1-13,15-21,23-32].

Әдетте «автомат» деген сөзбен іскеқосқаннан кейін бірқатар амалдарды өздігінен орындайтын құрылғыларды айтады. Біз барлық сигнал деңгей бойынша, ал амалдар уақыт бойынша кванттталған кез келген цифрлы (дискретті) құрылғының математикалық моделі ретінде қолданылатын абстрактылы автоматпен істі боламыз.

Абстрактылы автомат (ары қарай - автомат) қайсы бір жиынды танитын немесе оны басқа жиынға түрлендіреді және таспа, бастиек пен басқарушы құрылғыдан тұрады және жұмыс жадын да қамтиды.

Таспа – әрқайсысы қайсы бір кіріс (шығыс) әліпбиінің тек бір ғана символын сақтай алатын ұяшықтардың сызықты тізбегі.

Таспа шексіз, бірақ онда әрбір уақыт мезгілінде тек ақырлы сан ұяшығы ғана жүктелген болады. Жүктелген ұяшықтар аймағының сол және оң жағына шекаралас ұяшықтарда таспаның басын және соңын белгілейтін арнаулы маркерлер болуы мүмкін. Маркер таспаның тек сол жақ шетінде болуы немесе мүлде болмауы мүмкін.

Кіріс (*шығыс*) *бастиегі* әрбір уақыт мезгілінде таспаның тек бір ғана ұяшығын шолу жасай алады. Бастиек бір ұяшыққа солға жылжиды немесе бір ұяшыққа оңға жылжиды, иә болмаса қозғалмай орнында қалады. Бастиек әрі оқитын, әрі жазатын болады.

Жұмыс жады – деректерді оқу және жазу үшін арналған қосымша ақпарат қоймасы. Жұмыс жады динамикалық деректер құрылымы (кезек немесе стек) түрінде ұйымдастырылуы мүмкін.

Басқарушы құрылғы – күйлердің ақырлы жиынын сақтау үшін ішкі жады бар және автоматтың тәртібін басқаратын құрылғы. Ол ағымдағы күйге, бастиек оқитын ағымдағы кіріс символына және егер жұмыс жады бар болса, одан оқылатын ағымдағы ақпаратқа байланысты күйлердің қалай өзгеретіндігін сипаттайтын функция (қатынас) арқылы автоматтың жұмысын басқарады. Сондай-ақ, басқарушы құрылғы бастиектің қозғалу бағытын және жұмыс жадына қандай ақпарат жазылатындығын да анықтайды.

Автомат басқарушы құрылғы күйлерінің ақырлы жиынын, кіріс символдарын, бастапқы күй мен соңғы күйлер жиынын, өзінің аргументтері "ағымдағы" күй және "ағымдағы" кіріс символы болатын және мәндері барлық мүмкін келесі күйлер болатын ауысу функцияны беру арқылы анықталады.

Автоматтың жұмысын оның кескіні арқылы сипаттау ыңғайлы. Автомат кескіні мыналардан тұрады: басқарушы құрылғының күйі; бастиектің орнымен бірге кіріс таспасының қамтуы; жұмыс бастиегінің орнымен бірге жұмыс жадының қамтуы, егер ол бар болса; шығыс таспасының қамтуы, егер ол бар болса.

Автоматтың *алғашқы кескінінде* ішкі жад алдын ала жазылған басқарушы құрылғының алғашқы күйін белгілейтін символды қамтиды; басқарушы құрылғы алғашқы күйде болады; бастиек таспадағы ең сол кіріс символын оқиды; егер жұмыс жады болса, онда оның алдын ала орнатылған алғашқы қамтуы болады.

Автоматтың *ағымдағы кескінінде* ішкі жад алдын ала жазылған басқарушы құрылғының ағымдағы күйілерін белгілейтін символдарды қамтиды; басқарушы құрылғы ағымдағы күйілердің біреуінде болады; бастиек таспадағы ең сол емес және ең оң емес ағымдағы кіріс символды оқиды; егер жұмыс жады болса, онда оның алдын ала орнатылған ағымдағы қамтуы болады.

Автоматтың *соңғы кескінінде* ішкі жад алдын ала жазылған басқарушы құрылғының соңғы күйлерін белгілейтін символдарды қамтиды; басқарушы құрылғы соңғы күйлердің бірінде болады; егер таспаның оң шетінің маркері болса, онда бастиек оны оқиды, әйтпесе, таспаның сыртына шығып кетеді; егер жұмыс жады болса, онда ол қандай-да бір шарттарды қанағаттандырады.

Автомат жұмысынының алдында ішкі жадқа басқарушы құрылғының алғашқы күйін белгілейтін символы, кіріс таспасына кіріс тізбесін жадқа алғашқы кіріс тізбесі, және егер жұмыс жады қарастырылған болса, онда оған сәйкес ақпарат жазылады.

Автомат ақырлы *тактылар* тізбесінен тұратын программа арқылы жұмыс істейді. Әрбір такт ағымдағы (алғашқы) және келесі (соңғы) кескіндерден құрылады. Такты басында ішкі жадтан ағымдағы күйдің символы оқылады, кіріс таспасынан ағымдағы кіріс символы оқылады және жұмыс жадындағы ақпарат зерттеледі, егер ол бар болса. Кейін автоматтың әрекеті анықталады:

(1) кіріс таспасының ағымдағы ұяшығына жаңа символ жазылады немесе ондағы бұрынғы символ өзгерілмейді;

(2) кіріс бастиегі оңға, солға жылжиды немесе орнында қалады;

(3) жұмыс жадына ақпарат жазылады, егер ол бар болса;

(4) шығыс таспасына символ жазылады, егер ол бар болса;

(5) басқарушы құрылғы қайсы бір күйге көшеді.

Автоматтың бір тактысында кіріс бастиегі оңға, солға жылжиды немесе орнында қалады. Автомат жұмысында кіріс таспасының, жұмыс жадының, шығыс таспасының қамтулары өзгеруі мүмкін.

Егер автомат кіріс тізбесін қарастыруды алғашқы кескіннен бастап соңғы кескінмен аяқталатын тактылар қатарын орындайтын болса, онда ол осы кіріс тізбесін *танитын* болады. Автоматпен танылатын *тіл* танылатын тізбелер жиыны болады.

І.3.3-мысалдар:

1. Таксофон автоматты жүзеге асырудың мысалы болады: ендірген тиынды таниды және нөмірді теру күйіне өтеді.

2. Банкомат автомат қызметін атқарады: ендірген карточканы таниды және пинкодды ендіру күйіне өтеді.

3.Метродағы турникет автомат болады: ендірген жетонды таниды және кірісті ашу күйіне көшеді.

І.3.3-тапсырмалар:

1. Автоматтың кіріс таспасының құрылысын сипаттаңыз.
2. Автомат бастиегінің функциясын сипаттаңыз.
3. Автомат жадының функциясын сипаттаңыз.

І.3.3-сұрақтар:

1. Кіріс лентасы нені қамтиды?
2. Автоматтың кескіні неден тұрады?
3. Кескіндердің қандай түрлері бар?

I.3.4. Автоматтар типтері және алгоритмдік проблемалар

Бұл параграфта автоматтардың типтері олардың құрамына, күйлерді бекіту мен анықтау әдісіне, функциялық мүмкіншілігіне және басқа параметрлеріне байланысты қарастырылады, мысалдар беріледі, тапсырмалар мен сұрақтар қойылады [1-13,15-21,23-32].

Бастиектің жылжу бағытына байланысты автоматтар келесі екі топқа бөлінеді:

1) *Бір жақты автомат*, егер ол өзінің бастиегін ешқашан солға жылжытпаса;

2) *Екі жақты автомат*, егер ол өзінің бастиегін солға да, оңға да жылжытса.

Алғашқы күйдің бекітілуіне байланысты автоматтар келесі екі топқа бөлінеді:

1) *Инициалды автомат*, егер оның алғашқы мезгілінде басқарушы құрылғының күйі бекітілген болса;

2) *Бейинициалды автомат*, егер оның алғашқы мезгілінде басқарушы құрылғының күйі бекітілмеген болса;

Келесі тактыдағы күйлердің қалай анықталуына байланысты автоматтар келесі екі топқа бөлінеді:

1) *Детерминді автомат* (ДА), егер әрбір тактыда басқарушы құрылғының келесі күйі бірмәнді анықталса, яғни егер оның әрбір кескіні үшін тек бір ғана келесі такты бар болса;

2) *Бейдетерминді автомат* (БА), егер әрбір тактыда басқарушы құрылғының келесі күйі көпмәнді анықталса, яғни егер оның әрбір кескіні үшін мүмкін тактылардың ақырлы жиыны бар болса.

I.3.4-ескертпелер:

1. Бейдетерминділік автоматтың басқарушы құрылығысы өзінің барлық келесі күйлеріне көшетіндігін, егер қажет болса, өзін-өзі қайталап әрбір мүмкін келесі күйлерде осы құрылғының бір данасы болатындығын білдіреді.

2. Бейдетерминдікті бекітілген ықтималдықтары бар келесі күйлердің бірін кездейсоқ таңдай алатын "кездейсоқтықпен" шатастыруға болмайды.

3. Бейдетерминді автоматтар – ыңғайлы математикалық абстракция, бірақ, өкінішке орай, оларды практикада моделдеу қиын, ал детерминді автоматтарды моделдеу оңай. Сондықтан практикалық есептерді шешу үшін детерминді автоматтар қолданылады.

Функционалдық мүмкіншіліктігіне байланысты автоматтар келесі екі топқа бөлінеді:

1) *таныгыштар* – бір күйден екінші күйге көшіретін ауысу функциясын ғана қамтитын және кіріс тізбесі τ берілген $\boldsymbol{L}$ тіліне жататындығын танитын шығысы жоқ автоматтар, яғни мына сұраққа жауап береді: $\boldsymbol{\tau \in L}$**?**

2) *түрлендіргіштер* – ауысу функциясымен қатар, $\tau \in \boldsymbol{L}$ шарты орындалғанда τ кіріс тізбесін υ шығыс тізбесіне түрлендіретін түрлендіру функциясын да қамтитын шығысы бар автоматтар.

Жұмыс жадының күрделігіне (сипаттау күшіне) байланысты автоматтар келесі төрт топқа бөлінеді:

1) *ақырлы автоматтар* (АА) – құрамында жұмыс жады жоқ автоматтар;

2) *стектік автоматтар* (СА) – жұмыс жады стек прінсіпінде ұйымдастырылған автоматрт: соңғы енген, бірінші шығады - Last Input First Output (LIFO);

3) *сызықты-шенеуленген автоматтар* (США) – жұмыс жады кіріс тізбесінің ұзындығына сызықты функция болатын автоматтар;

4) *Тьюринга машиналары* (ТМ) – таспасы екі жағынан да (солдан да, оңнан да) шенеуленбеген автоматтар;

0-топтағы тілдер (шенеуленбеген); МТ, ШГ практикада қолданылмайды.

1-топтағы тілдер (контексті-тәуелді); ЛОА, КТГ табиғи тілдерден аударушыларды құру үшін қолданылады.

2-топтағы тілдер (контексті-бос); СА, КБГ программалау тілдердердің синтаксистік талдаушыларын құру үшін қолданылады.

3-топтағы тілдер (регулярлық); АА, СГ мәтіндік редакторларды және программаларды баптаушыларды құру үшін қолданылады.

Формалды грамматикалар мен автоматтар арасында мынадай байланыс бар:

Формалды грамматикалардағы секілді, автоматтардың да өз алгоритмдік проблемалары бар. Оларға мыналар жатады:

1. *Бостық проблемасы* – берілген $\boldsymbol{M}$ автоматы үшін $\boldsymbol{L}(\boldsymbol{M})$ бос тіл бола ма, яғни , $\boldsymbol{L}(\boldsymbol{M}) = \varnothing$?

2. *Шексіздік проблемасы* – берілген $\boldsymbol{M}$ автоматы үшін $\boldsymbol{L}(\boldsymbol{M})$ шексіз тіл бола ма, яғни $\boldsymbol{L}(\boldsymbol{M}) = \infty$?

3. *Тиістілік проблемасы* – кез келген ξ тізбе берілген $\boldsymbol{M}$ автоматы танитын $\boldsymbol{L}(\boldsymbol{M})$ тіліне тиісті бола ма, яғни $\xi \in \boldsymbol{L}(\boldsymbol{M})$?

4. *Эквиваленттілік проблемасы* – кез келген екі $\boldsymbol{M_i}$ және $\boldsymbol{M_j}$ $(\boldsymbol{i \neq j})$ автоматтар эквивалентті бола ма, яғни $\boldsymbol{L}(\boldsymbol{M_i}) = \boldsymbol{L}(\boldsymbol{M_j})$?

5. *Тұйықтылық проблемасы* – белгілі бір типтегі кез келген екі тілге жиындық амалдарды қолданғанда нәтижесі сол топқа жататындығын анықтау.

6.Тоқтату проблемасы – берілген $\boldsymbol{M}$ автоматы және берілген кіріс деректері үшін осы деректермен $\boldsymbol{M}$ автоматының жұмысы тоқтай ма жоқ па?

Осы проблемалардың кейбіреулері алгоритмдік шешімді болады.

I.3.4-тапсырмалар:

1. Біржақты және екіжақты автоматтардың айырмашылығын көрсетіңіз.
2. Инициалды және бейинициалды автоматтардың айырмашылығын көрсетіңіз.
3. Детерминді және бейдетерминді автоматтардың айырмашылығын көрсетіңіз.
4. Танығыштар мен түрлендіргіштердің айырмашылағын көрсетіңіз.
5. Регулярлы өрнектер мен оң сызықты грамматикаларға сәйкес автоматты көрсетіңіз.
6. Контексті-бос грамматикаларға сәйкес автоматты көрсетіңіз.
7. Контексті-тәуелді грамматикаларға сәйкес автоматты көрсетіңіз.
8. Жалпы түрдегі грамматикаларға сәйкес автоматты көрсетіңіз.
9. Ақырлы автоматтар мен стекті автоматтардың айырмашылығын көрсетіңіз.
10. Сызықты-шенеуленген автоматтар мен Тьюринга машиналарының айырмашылығын көрсетіңіз.

I.3.4-сұрақтар:

1. Регулярлық грамматикалар қандай автоматтарға сәйкес?
2. Контекстісіз грамматикалар қандай автоматтарға сәйкес?
3. Контексті грамматикалар қандай автоматтарға сәйкес?
4. Шектелген грамматикалар қандай автоматтарға сәйкес?
5. Ақырлы автомат тілдердің қандай сыныбын таниды?
6. Стекті автомат тілдердің қандай сыныбын таниды?
7. Сызықты шенеуленген автомат қандай тілдерді таниды?
8. Тьюринг машинасы тілдердің қандай сыныбын таниды?
9. Автоматтардағы бостық проблемасы деген не?
10. Автоматтардағы шексіздік проблемасы деген не?
11. Автоматтардағы тиістілік проблемасы деген не?
12. Автоматтардағы эквиваленттілік проблемасы деген не?
13. Автоматтардағы тұйықтылық проблемасы деген не?
14. Автоматтардағы тоқтату проблемасы деген не?

I.3.5. Автоматтың формалды анықтамасы

Бұл параграфта шығысы жоқ автоматтардың формалды анықтамасы, олардың кескіні, жұмыс тактысы және олармен танылатын тілдер, сонымен қатар, олардың алгоритмдік проблемалары қарастырылады, мысалдар беріледі, тапсырмалар ұсынылады, сұрақтар қойылады [1-5,12,13,16,21, 23,25-32].

I.3.5.1-анықтама. *Бейинициалды автомат **A*** мына алтылықпен $\boldsymbol{A} = <S, I, O, F_s, F_o, S_f>$ беріледі, мұнда:

$S = \{s_1, s_2, \ldots, s_K\}$ – күйлер жиыны (күйлер әліпбиі);

$I = \{i_1, i_2, \ldots, i_l\}$ – кіріс символдар жиыны (кіріс әліпбиі);

$O = \{o_1, o_1, \ldots, o_M\}$ – шығыс символдар жиыны (шығыс әліпбиі);

F_s: $S \times I \rightarrow \vartheta(S)$ – алмастыру функциясы, қандай-да бір (s_k, i_l) күй-кіріс символ жұбына автомат күйін $s_k = F_s(s_n, i_l)$ сәйкес қояды; $s_k, s_n \in S$; $k \neq n$; $k=1, 2, \ldots, K$; $n=1, 2, \ldots, K$; $i_l \in I$; $l=1, 2, \ldots, L$; $\vartheta(S)$ – S жиынының барлық ішжиындарының жиыны;

F_o: $S \times I \rightarrow O$ – шығыс функциясы, қандай-да бір (s_k, i_l) күй-кіріс жұбына шығыс символын $o_n = F_s(s_k, i_l)$ сәйкес қояды; $o_n = F_s(s_k, i_l)$; $s_k, s_n \in S$; $k \neq n$; $k=1, 2, \ldots, K$; $n=1, 2, \ldots, K$; $i_l \in I$; $l=1, 2, \ldots, L$; $o_m \in O$; $m=1, 2, \ldots, M$;

$S_f \subseteq S$ – соңғы (финалдық) күйлер жиыны.

I.3.5.2-анықтама. *Инициалды автомат **A*** мына жетілікпен

$\boldsymbol{A} = <S, I, O, F_s, F_o, s_b, S_f>$ беріледі, мұнда

S, I, O, F_s, F_o, S_f – бейинициалды автоматтағы сияқты;

$s_b \in S$ – алғашқы күй.

Белгілеулер келесі атауларға сәйкес алынды: $\boldsymbol{A}$ – от «*automaton*», S / s – «*state*», I / i – «*input*», O / o – «*output*», F – «*function*», s_b – «*state begin*», s_f – «*state final*».

Ары қарай біз инициалды және бейинициалды автоматтарды ажыратпаймыз, себебі қандай автомат туралы сөз болып жатқаны мәнмәтіннен көрінеді.

Автомат *аракідік анықталған* деп аталады, егер F_s және F_o функцияларының анықталу облысы $S\times I$ жиынының ішжиыны болса, яңни, $D(F_s)\subseteq S\times I$, $D(F_o)\subseteq S\times I$ және *толық анықталған* деп аталады, егер $D(F_s)= D(F_s)=S\times I$.

Ары қарай біз тек алмастыру функциясымен сәйкес жұмыс істейтін автоматтар танығышты ғана қарастырамыз.

І.3.5-ескертпе. Кейде алмастыру (түрлендіру) функциясының орнына логикалық «ақиқат» немесе «жалған» мәнін қабылдайтын алмастыру (түрлендіру) қатынасын пайдалану ыңғайлы.

Алмастыру (түрлендіру) функциясынан алмастыру (түрлендіру) қатынасына көшу үшін осы функцияның мәнін қатынастың қосымша аргументі ретінде қабылдау керек.

Жұмыс жады жоқ кескіннің формалды анықтамасы мына жұп $(s, \omega)\in S\times I^*$, мұнда $s\in S$ және $\omega\in I^*$, I^* - барлық мүмкін болатын ақырлы ұзындықты кіріс тізбесінің жиыны.

Автомат кескіні алғашқы (*соңғы*) болады, егер жұптағы бірінші элемент *алғашқы* (*соңғы*) күй s_b (s_f), ал екінші элемент – кіріс (бос) тізбе болса.

Автомат тактысы – екі кескін арасында анықталған бинарлы қатынас $\models$ таңбасымен белгіленеді.

$(s_t, i\omega) \models (s_{t+1}, \omega)$ қатынасының бар болғандығы, егер автомат s_t күйінде болса және бастиек i символын қарастырып тұрса, онда детерминді (бейдетерминді) автомат бір такты ішінде s_{t+1} (s_{t+1} күйден басқаларға) өтетіндігін және бастиек бір ұяшыққа оңға жылжитындығын білдіреді.

$\models^n$ қатынасы $\models$ қатынасының n–ші дәрежесі болады, егер $s_0=s_b$, $\tau_0=\tau$, $s_k= s_f$, $\tau_k=\varepsilon$ және $\forall\ k,\ 1 \le k \le n$ болатындай (s_0,τ_0), (s_1,τ_1), $(s_2,\tau_2),\ldots,$ (s_{n-1},τ_{n-1}), (s_n,τ_n) тізбесі бар болып мына $(s_{k-1},\tau_{k-1}) \models (s_k,\tau_k)$ қатынас орындалады, мұнда ε – бос тізбе.

Егер $\forall k$, $k\geq 1$ (немесе $k\geq 0$) қатынас $(s_0,\tau)\models^{k}(s_k,\varepsilon)$ орындалса, онда ол былай $(s_0,\tau)\models^{+}(s_k,\varepsilon)$ жазылады (немесе$(s_0,\tau)\models^{*}(s_k,\varepsilon)$), мұнда $\models^{+}$ – $\models$ қатынасының транзитивті тұйықталуы, а $\models^{*}$ – $\models$ қатынасының рефлексивті және транзитивті тұйықталуы.

Кіріс тізбесі шығыссыз автоматпен танылады, егер ол кіріс тізбесі таспаға жазылған алғашқы кескіннен бастап соңғы кескінге дейін кескін тізбегін орындаса.

Автоматпен танылатын тіл деп осы автомат танитын тізбелер жиынын айтады.

Мейлі (s_b, τ) – автоматтың алғашқы кескіні, (s_f, ε) – автоматтың соңғы кескіні, сонда:

1) $\boldsymbol{A}$ автомат-танушы $\tau \in I^*$ кіріс тізбесін таныса, егер мына $(s_b, \tau) \models^* (s_f, \varepsilon)$ қатынас орындалса;

2) $\boldsymbol{L}(\boldsymbol{A})$ тілі $\boldsymbol{A}$ автомат-танығышпен танылса, егер ол былай анықталса $\boldsymbol{L}(\boldsymbol{A}) = \{\tau\colon \tau \in I^* \ \&\ (s_b, \tau) \models^* (s_f, \varepsilon)\}$;

Пәрмен бар кезде автомат жұмыс істейді: егер t тактысында басқарушы құрылғы s күйінде болса және кіріс таспасында k символын қарастырып тұрса, онда ол шығыс таспасына o символын жазады және келесі тактыға көшеді t+1, яғни

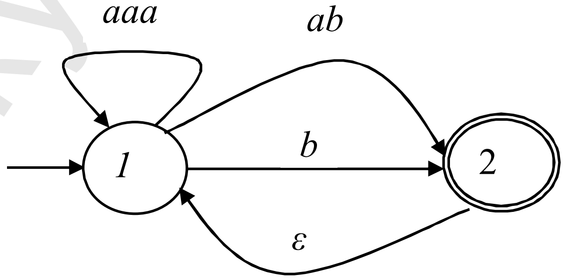
$$o(t) = F_s(s(t), k(t)), \ \ s(t+1) = F_o(s(t), k(t)).$$

Егер қандай-да бір пәрмен жоқ болса, автомат бұғатталып, t мезгілінде қабылданған символға қарамайды және келесі мезгілдерде символдарды қабылдауды тоқтатады.

Кейде автомат пәрмендерін үштік түрінде беруге болады:

$$<s(t), k(t), o(t)>, \ \ <s(t), k(t), s(t+1)>$$

I.3.5-мысалдар: Мына параметрлі $\boldsymbol{Q} = \{1,2\}$; $\boldsymbol{T} = \{a,b\}$; $\boldsymbol{I} = \{1\}$; $\boldsymbol{F} = \{2\}$; $\Delta = \{<1, aaa, 1>, <1, ab, 2>, <1, b, 2>, <2, \varepsilon, 1>\}$ ақырлы автомат диаграммасы 1.3.5-суретте берілген.



I.3.5-сурет. Ақырлы автомат диаграммасы

Сонда автомат тактысы былай болады $(1, abba) \models (2, ba)$.

I.3.5-тапсырмалар:

1. Инициалды автоматтың формалды анықтамасын беріңіз.
2. Бейинициалды автоматтың формалды анықтамасын беріңіз.
3. $\{a, b\}^*$ тілін танитын автоматтты құрыңыз.
4. $\{\tau \in \{a, b, c\}^* : |\tau| \neq 1\}$ тілін танитын автоматты табыңыз.
5. $\{a, b\}^* \setminus (\{a^n : n \geq 0\} \cup \{b^n : n \geq 0\})$ тілі үшін автомат табыңыз.
6. $\{a\omega b:\ \omega \in \{a, b\}^*\} \cup \{b\omega a:\ \omega \in \{a, b\}^*\}$ тілін танитын автоматты құрыңыз.
7. a мен b әріптерінен тұратын және a әріпімен басталатын тізбені танитын ақырлы автоматты тұрғызыңыз.
8. a мен b әріптерінен тұратын және b әріпімен аяқталатын тізбені танитын ақырлы автоматты тұрғызыңыз.
9. a, b, c әріптерінен тұратын және c әріпімен басталатын және c әріпіне аяқталатын тізбені танитын ақырлы автоматты тұрғызыңыз.
10. Кіші латын әріптерінен және араб цифрларынан тұратын барлық тізбелерді танитын ақырлы автоматты тұрғызыңыз.
11. Мына тілді $\{a, bb\}^*$ танитын автомат құрыңыз.
12. Мына тілді $\{\omega \in \{a, b\}^* : |\omega| \geq 3\}$ танитын автомат құрыңыз.

I.3.5-сұрақтар:

1. Автомат кескін неден тұрады?
2. Автомат кескінінің қандай түрлері бар?
3. Автомат тактысы неден тұрады?
4. Автоматтың алмастыру функциясы қалай беріледі?
5. Автоматтың шығару функциясы қалай беріледі?
6. Қашан автомат аракідік анықталған болады?
7. Қашан автомат толық анықталған болады?
8. Автоматпен танылған тізбелер жиыны нені білдіреді?
9. Ақырлы автоматтың пәрмені қалай жазылады?
10. Инициалды автомат қалай анықталады?
11. Автоматтың диаграммасы деген не?
12. Автомат тілді қалай таниды?
13. Ақырлы автомат $\{w \in \{a, b, c\}^* : |w|_c = 1\}$ тілді тани ма?

II. РЕГУЛЯРЛЫҚ ТІЛДЕР

II.1. Регулярлық тілдерді тудырушы механизмдер

II.1.1. Регулярлық жиындар мен регулярлық өрнектер

Бұл бөлім регулярлық тілдерді анықтау тәсілдері (тудырушы механизмдерге жататын регулярлық өрнектер, регулярлық теңдеулер, регулярлық теңдеулер жүйелері, сызықты грамматикалар және танушы механизмдері болатын ақырлы автоматтар) қарастырылады, регулярлық тілдердің қасиеттері және осы тілдерді анықтайтын әртүрлі механизмдердің эквиваленттілігі талқыланады, мысалдар беріледі, тапсырмалар ұсынылады, сұрақтар қойылады [1-3,16-21,24,25,28-32].

Мейлі $\varnothing$ – бос жиын, ал $\{\varepsilon\}$ – бос тізбектер жиыны, $\boldsymbol{T}$ – ақырлы әліпби, ал t – осы әліпбидегі символ, яғни $t\in\boldsymbol{T}$ болсын. Сонда мынадай анықтама жасауға болады.

II.1.1.1-анықтама. $\boldsymbol{T}$ әліпбиіндегі *регулярлық жиын* рекурсивті былай анықталады:

Рекурсия базисі:

(1) $\varnothing$ – $\boldsymbol{T}$ әліпбиіндегі регулярлық жиын;

(2) $\{\varepsilon\}$ – $\boldsymbol{T}$ әліпбиіндегі регулярлық жиын;

(3) $\{t\}$ – $\boldsymbol{T}$ әліпбиіндегі регулярлық жиын;

Рекурсивті кеңейту:

Егер $\boldsymbol{P}$ және $\boldsymbol{Q}$ – $\boldsymbol{T}$ әліпбиіндегі регулярлық жиындар болса, онда келесі жиындар да регулярлық болады:

(4) $\boldsymbol{P}\cdot\boldsymbol{Q}$ – $\boldsymbol{P}$ және $\boldsymbol{Q}$ жиындарының тіркесуі,

(5) $\boldsymbol{P}\cup\boldsymbol{Q}$ – $\boldsymbol{P}$ және $\boldsymbol{Q}$ жиындарының бірігуі ,

(6) $\boldsymbol{P}^*$ – $\boldsymbol{P}$ жиынының итерациясы;

Рекурсивті тұжырымдау:

$\boldsymbol{T}$ әліпбиіндегі регулярлық жиын тек (1) – (6) ережелерімен анықталады.

Сонымен, $\boldsymbol{T}$ әліпбиіндегі регулярлық жиын болады тек қана және тек қана, егер ол немесе $\varnothing$, немесе $\{\varepsilon\}$, немесе қайсы бір $t\in\boldsymbol{T}$ үшін $\{t\}$ болатын болса, немесе ол осы жиындардан тіркесу, бірігу және итерация амалдарын ақырлы сан рет қолдану арқылы алынатын болса.

ІІ.1.1.2-анықтама. Ақырлы $\boldsymbol{T}$ әліпбиіндегі регулярлық жиындарды анықтайтын регулярлық өрнек рекурсивті болады:

Рекурсия базисі:

(1) ϕ – $\boldsymbol{T}$ әліпбиіндегі $\varnothing$ регулярлық жиынды анықтайтын регулярлық өрнек;

(2) ε – $\boldsymbol{T}$ әліпбиіндегі $\{\varepsilon\}$ регулярлық жиынды анықтайтын регулярлық өрнек;

(3) $\{t\}$ – $\boldsymbol{T}$ әліпбиіндегі $\{t\}$ регулярлық жиынды анықтайтын регулярлық өрнек;

Рекурсивті кеңейту:

Егер p және q – $\boldsymbol{T}$ әліпбиіндегі $\boldsymbol{P}$ және $\boldsymbol{Q}$ регулярлық жиындарды беретін сәйкес регулярлық өрнектер болса, онда:

(4) $p{\cdot}q$ – тіркесуден пайда болатын $\boldsymbol{T}$ әліпбиіндегі $\boldsymbol{P{\cdot}Q}$ регулярлық жиынды сипаттайтын регулярлық өрнек;

(5) $p{\vee}q$ – бірігуден пайда болатын $\boldsymbol{T}$ әліпбиіндегі $\boldsymbol{P}{\cup}\boldsymbol{Q}$ регулярлық жиынды сипаттайтын регулярлық өрнек;

(6) p^* – итерациядан пайда болатын $\boldsymbol{T}$ әліпбиіндегі $\boldsymbol{P}^*$ регулярлық жиынды сипаттайтын регулярлық өрнек;

Рекурсивті тұжырымдау:

Регулярлық өрнектер тек (1) – (6) ережелерімен анықталады.

Регулярлық өрнектерді құру үшін тіркесу амалы ·, таңдау амалы ∨ және итерация амалы * қолданылады. Ең жоғарғы орындалу ретке * амалы, сонан кейінгіге · амалы, ал ең соңындағыға ∨ амалы ие болады. Әдетте, регулярлық өрнектің жазуында итерация амалылының таңбасы · жазылмайды.

Біз p^+ жазбасын pp^* жазбасын қысқарту үшін қолданамыз, сонымен қатар, регулярлық өрнектен артық жақшалардан құтыламыз, мысалы, $0{\vee}\mathit{1}0^*$ деген $(0{\vee}(\mathit{1}(0^*)))$ болады.

Әрбір регулярлық жиынды сипаттайтын ең болмаса бір регулярлық өрнек табылады және керісінше, әрбір регулярлық өрнекпен сипатталатын регулярлық жиын бар болады.

Әрбір регулярлық жиын үшін оны сипаттайтын шексіз көп регулярлық өрнек бар екендігін ескеру керек.

Екі регулярлық өрнекті *тең* деп атаймыз, егер олар бір жиынды сипаттайтын болса. Регулярлық өрнектерді эквивалентті түрлендіру үшін бірқатар алгебралық заңдар (аксиомалар мен теоремалар) бар.

Лексикалық құрылымдарды практикалық сипаттағанда регулярлық өрнектектерге қандай да бір атаулар беріп, оларға сілтемелерді осы атаулар арқылы жасау пайдалы. Осындай атауларды анықтау үшін мынадай жазбаларды пайдаланамыз:

$$d_1 = r_1$$
$$d_2 = r_2$$
$$\dots$$
$$d_n = r_n$$

мұнда d_i – әртүлі атаулар, ал әрбір r_i – мына $\boldsymbol{T} \cup \{d_1, d_2, \dots, d_{i-1}\}$ символдардағы, яғни негізгі әліпби символдардағы және алдын ала анықталған атау-символдарындағы регулярлық өрнек. Сонымен, кез келген r_i үшін регулярлық өрнектердің атауын олар атайтын регулярлық өрнектерге қайта ауыстыру арқылы $\boldsymbol{T}$ әліпбиінде регулярлық өрнек тұрғызуға болады

II.1.1.1-мысалдар:

1. $(a \vee b)^*$ регулярлық өрнегі $\{a,b\}^*$ регулярлық жиынын анықтайды.

2. $(0 \vee 1)^*011$ регулярлық өрнегі 0 мен *1*-ден құралған және 0*11* тізбесіне аяқталатын тізбелердің жиынын анықтайды.

3. $a(a \vee 0)^*$ регулярлық өрнегі $\{0,a\}^*$ жиынындағы, *a*-дан басталатын тізбелердің жиынын анықтайды.

4. $(a \vee b)(a \vee b \vee 0 \vee 1)^*$ регулярлық өрнегі $\{0,1,a,b\}^*$ жиынындағы, a-дан немесе b-дан басталатын тізбелердің жиынын анықтайды.

5. $(00 \vee 11)^* \cdot ((01 \vee 10) \cdot (00 \vee 11)^* \cdot (01 \vee 10)(00 \vee 11)^*)^*$ регулярлық өрнегі 0 мен 1-ден құралып, жұп сан 0 және жұп сан 1-ден тұратын барлық тізбелер жиынын анықтайды.

6. $a(\varepsilon \vee a) \vee b$ регулярлық өрнегі $\{a, b, aa\}$ регулярлық жиынын анықтайды.

7. $a(a \vee b)^*$ регулярлық өрнегі a-дан басталатын және a мен b-дан тұратын барлық тізбелер жиынын анықтайды.

8. $(a \vee b)^*(a \vee b)(a \vee b)^*$ регулярлық өрнегі a мен b-дан тұратын барлық бос емес тізбелер жиынын, яғни $\{a, b\}^+$ жиынын анықтайды.

9. $((s \vee 1)(s \vee 1)(s \vee 1))^*$ регулярлық өрнегі ұзындығы 0 мен 1-ден тұратын және 3-ке бөлінетін барлық тізбелер жиынын анықтайды.

10. ϕ^* регулярлық өрнегі $\{\varepsilon\}$ жиынын анықтайды.

11. $\varepsilon \vee \phi$ регулярлық өрнегі $\{\varepsilon\} \cup \varnothing$ жиынын анықтайды.

12. $(a \vee b) \cdot (a \vee b) = aa \vee ab \vee ba \vee bb$.

13. Әріптер мен цифрлардан тұратын және әріптен басталатын айқындағыш ұғымының құрылымы былай жазылады:

$L = a \vee b \vee c \vee d \vee e \vee f \vee g \vee h \vee i \vee j \vee k \vee l \vee m \vee n \vee o \vee p \vee q \vee s \vee t \vee u \vee v \vee w \vee x \vee y \vee z$

$D = 0 \vee 1 \vee 2 \vee 3 \vee 4 \vee 5 \vee 6 \vee 7 \vee 8 \vee 9$

$I = L(L \vee D)^*$

14. Мейлі $\boldsymbol{T} = \{x,y\}$ әліпбиі берілсін. Сонда анықтама бойынша

$$\boldsymbol{L}((xy)^* \cdot (1 \vee x)) = \{(xy)^n : n \geq 0\} \cup \{(xy)^n x : n \geq 0\}.$$

15. Ондық жазудағы сандар жиыны үшін регулярлық өрнек:

$Digit = 0 \vee 1 \vee 2 \vee 3 \vee 4 \vee 5 \vee 6 \vee 7 \vee 8 \vee 9$

$Integer = Digit^+$

$Fraction = Integer \vee \varepsilon$

$Exponent = (E(+ \vee - \vee e)Integer) \vee \varepsilon$

II.1.1-тапсырмалар:

1. $\{a,b\}^*$ жиынын сипаттайтын регулярлық өрнекті табыңыз.
2. $\{a,bc^*\}^*$ жиынын сипаттайтын регулярлық өрнекті табыңыз.

3. $\{ab,cd\}^*$ тілінің толықтауы үшін регулярлық өрнекті табыңыз.

4. $\{ab,b\}^*$ жиынын сипаттайтын регулярлық өрнекті табыңыз.

5. $\{a^*,b^*\}^*$ жиынын сипаттайтын регулярлық өрнекті табыңыз.

6. Құрамы сөздерді, кетіктерді және нүктелерді қамтитын және құрылымы тек латынның кіші әріптерінен тұратын сөзден басталатын, сөздер арасында кетік болатын және барлығы тек жалғыз нүктемен аяқталатын сөйлемдерді тудыратын регулярлық өрнекті жазыңыздар.

7. Құрамы күнді, айды, жылды қамтитын және құрылымы “<цифр><цифр>.<цифр><цифр>.<цифр><цифр><цифр><цифр>” түрінде болатын құрылымды тудыратын регулярлық өрнекті жазыңыздар.

ІІ.1.1-сұрақтар:

1. ***T*** әліпбиінде регулярлық жиын қалай анықталады?

2. $(ab)^+$ регулярлық өрнегі қандай регулярлық жиынды сираттайды?

3. $(aa\vee bb)$ регулярлық өрнегі қандай регулярлық жиынды сираттайды?

4. $a(\varepsilon\vee a)\vee b$ регулярлық өрнегі қандай регулярлық жиынды анықтайды?

5. $a(a\vee b)^*$ регулярлық өрнегі қандай регулярлық жиынды анықтайды?

6. ε^* регулярлық өрнегі қандай регулярлық жиынды анықтайды?

7. ϕ регулярлық өрнегі қандай регулярлық жиынды анықтайды?

8. $((a\vee bc)^* \cdot a)$ регулярлық өрнегі қандай регулярлық жиынды анықтайды?

9. $(c\cdot(ab\vee cd)^*)$ регулярлық өрнегі қандай регулярлық жиынды анықтайды?

10. $((a\vee b)^* \vee (c\vee d)^*)$ регулярлық өрнегі қандай регулярлық жиынды анықтайды?

11. Мына тіл $\{uvu^R\colon u \in \{a, b\}^+, v \in \{a, b\}^*\}$ регулярлық тіл бола ма?

II.1.2. Регулярлық алгебра мен регулярлық теңдеулер

Бұл параграфта регулярлық алгебра және регулярлық теңдедеулер қарастырылады, мысалдар беріледі, тапсырмалар ұсынылады, сұрақтар қойылады [1,16-21,25,29,32,32].

Әдетте алгебра ретінде берілген элементтер жиыны, оларда анықталған қайсы бір амалдар және осы амалдардың қасиеттері қарастырылады. Регулярлық алгебра элементтері ретінде ***T*** әліпбиіндегі тізбелер, оларда анықталған ·, ∨, * амалдармен берілген регулярлық өрнектер қабылданады, ал амалдардың қасиеттері келесі леммамен орнатылады.

II.I.2.1-лемма. Егер α, β және γ кез келген ϕ мен ε-нен өзгеше регулярлық өрнектер болса, онда мыналар ақиқат:

(1) $\phi\cdot\alpha=\alpha\cdot\phi=\phi$

(2) $\varepsilon\cdot\alpha=\alpha\cdot\varepsilon=\alpha$

(3) $\alpha\cdot\beta\neq\beta\cdot\alpha$

(4) $\phi\vee\alpha=\alpha\vee\phi=\alpha$

(5) $\alpha\vee\alpha=\alpha$

(6) $\alpha\vee\beta=\beta\vee\alpha$

(7) $\alpha\cdot(\beta\cdot\gamma)=(\alpha\cdot\beta)\cdot\gamma=\alpha\cdot\beta\cdot\gamma$

(8) $\alpha\vee(\beta\vee\gamma)=(\alpha\vee\beta)\vee\gamma=\alpha\vee\beta\vee\gamma$

(9) $\alpha\cdot(\beta\vee\gamma)=\alpha\cdot\beta\vee\alpha\cdot\gamma$

(10) $(\alpha\vee\beta)\cdot\gamma=\alpha\cdot\gamma\vee\beta\cdot\gamma$

(11) $\varepsilon^* = \varepsilon$

(12) $\phi^* = \varepsilon$

(13) $\alpha^*=\alpha\vee\alpha^*$

(14) $(\alpha^*)^*=\alpha^*$

Дәлелдеу. Регулярлық өрнектер α,β және γ сәйкес $\boldsymbol{A}$, $\boldsymbol{B}$ және $\boldsymbol{G}$ регулярлық жиындарды сипаттасын. Сонда:

(1) $\phi\cdot\alpha$ регулярлық өрнегі $\boldsymbol{\emptyset\cdot A}$ регулярлық жиынын анықтайды, ал $\alpha\cdot\phi$ регулярлық өрнегі $\boldsymbol{A\cdot\emptyset}$ регулярлық жиынын анықтайды. Жиындардың тік көбейтуі $\cdot$ амалының қасиеті бойынша $\boldsymbol{\emptyset\cdot A = A\cdot\emptyset} = \emptyset$ орын алады, сондықтан $\phi\cdot\alpha=\alpha\cdot\phi=\phi$.

(2) $\varepsilon\cdot\alpha$ регулярлық өрнегі $\{\varepsilon\}\cdot\boldsymbol{A}$ регулярлық жиынын анықтайды, ал $\alpha\cdot\varepsilon$ регулярлық өрнегі $\boldsymbol{A}\cdot\{\varepsilon\}$ регулярлық жиынын анықтайды. Жиындардың тік көбейтуі $\cdot$ амалының қасиеті бойынша $\{\varepsilon\}\cdot\boldsymbol{A} = \boldsymbol{A}\cdot\{\varepsilon\} = \boldsymbol{A}$, сондықтан $\varepsilon\cdot\alpha=\alpha\cdot\varepsilon = \alpha$.

(3) $\alpha\cdot\beta$ регулярлық өрнегі $\boldsymbol{A\cdot B}$ регулярлық жиынын анықтайды, ал $\beta\cdot\alpha$ регулярлық өрнегі $\boldsymbol{B\cdot A}$ регулярлық жиынын анықтайды. Жиындардың тік көбейтуі $\cdot$ амалының қасиеті бойынша $\boldsymbol{A\cdot B \neq B\cdot A}$ орын алады, сондықтан $\alpha\cdot\beta \neq \beta\cdot\alpha$.

(4) $\phi\vee\alpha$ регулярлық өрнегі $\emptyset\cup\boldsymbol{A}$ регулярлық жиынын анықтайды, ал $\alpha\vee\phi$ регулярлық өрнегі $\boldsymbol{A}\cup\emptyset$ регулярлық жиынын анықтайды. Жиындардың бірігу $\cup$ амалының қасиеті бойынша $\emptyset\cup\boldsymbol{A} = \boldsymbol{A}\cup\emptyset = \boldsymbol{A}$, сондықтан $\phi\vee\alpha=\alpha\vee\phi=\alpha$.

(5) $\alpha\vee\alpha$ регулярлық өрнегі $\boldsymbol{A}\cup\boldsymbol{A}$ регулярлық жиынын анықтайды, ал $\boldsymbol{A}\cup\boldsymbol{A}=\boldsymbol{A}$ екендігі белгілі, сондықтан $\alpha\vee\alpha=\alpha$.

(6) $\alpha\vee\beta$ регулярлық өрнегі $\boldsymbol{A}\cup\boldsymbol{B}$ регулярлық жиынын анықтайды, ал $\beta\vee\alpha$ регулярлық өрнегі $\boldsymbol{B}\cup\boldsymbol{A}$ регулярлық жиынын анықтайды. Жиындардың бірігу $\cup$ амалының қасиеті бойынша $\boldsymbol{A}\cup\boldsymbol{B} = \boldsymbol{B}\cup\boldsymbol{A}$ орын алады, сондықтан $\alpha\vee\beta=\beta\vee\alpha$.

(7) $\alpha\cdot(\beta\cdot\gamma)$ регулярлық өрнегі $\boldsymbol{A\cdot(B\cdot G)}$ регулярлық жиынын анықтайды, ал $(\alpha\cdot\beta)\cdot\gamma$ регулярлық өрнегі $\boldsymbol{(A\cdot B)\cdot G}$ регулярлық жиынын анықтайды. Жиындар теориясынан мынау $\boldsymbol{A\cdot(B\cdot G) = (A\cdot B)\cdot G = A\cdot B\cdot G}$ белгілі. Сондықтан $\alpha\cdot(\beta\cdot\gamma) = (\alpha\cdot\beta)\cdot\gamma = \alpha\cdot\beta\cdot\gamma$.

(8) $\alpha\vee(\beta\vee\gamma)$ регулярлық өрнегі $\boldsymbol{A}\cup(\boldsymbol{B}\cup\boldsymbol{G})$ регулярлық жиынын анықтайды, ал $(\alpha\vee\beta)\vee\gamma$ регулярлық өрнегі $(\boldsymbol{A}\cup\boldsymbol{B})\cup\boldsymbol{G}$ регулярлық

жиынын анықтайды. $A\cup(B\cup G)=(A\cup B)\cup G=A\cup B\cup G$ екендігі белгілі. Сондықтан $\alpha\vee(\beta\vee\gamma)=(\alpha\vee\beta)\vee\gamma=\alpha\vee\beta\vee\gamma$.

(9) $\alpha\cdot(\beta\vee\gamma)$ регулярлық өрнегі $A\cdot(B\cup G)$ регулярлық жиынын анықтайды, ал $\alpha\cdot\beta\vee\alpha\cdot\gamma$ регулярлық өрнегі $A\cdot B\cup A\cdot G$ регулярлық жиынын анықтайды. Жиындар теориясынан мынау $A\cdot(B\cup G)=A\cdot B\cup A\cdot G$ белгілі. Сондықтан $\alpha\cdot(\beta\vee\gamma) = \alpha\cdot\beta\vee\alpha\cdot\gamma$.

(10) $(\alpha\vee\beta)\cdot\gamma$ регулярлық өрнегі $(A\cup B)\cdot G$ регулярлық жиынын анықтайды, ал $\alpha\cdot\gamma\vee\beta\cdot\gamma$ регулярлық өрнегі $A\cdot G\cup B\cdot G$ регулярлық жиынын анықтайды. Жиындар теориясынан мынау $(A\cup B)\cdot G=A\cdot G\cup B\cdot G$ белгілі. Сондықтан $(\alpha\vee\beta)\cdot\gamma=\alpha\cdot\gamma \vee\beta\cdot\gamma$.

(11) ε^* регулярлық өрнегі $\{\varepsilon\}^*$ жиынын анықтайды, ал өз кезегінде мынау $\{\varepsilon\}^* = \{\varepsilon\}\cup\{\varepsilon\}\cup\{\varepsilon\}\cdot\{\varepsilon\}\cup\{\varepsilon\}\cdot\{\varepsilon\}\cdot\{\varepsilon\}\cup\ldots = \{\varepsilon\}$ белгілі, яғни $\{\varepsilon\}^* = \{\varepsilon\}$. Ал $\{\varepsilon\}$ жиыны ε регулярлық өрнекпен сипатталады, сондықтан $\varepsilon^* = \varepsilon$.

(12) Кез келген α үшін $\alpha^* = \varepsilon\vee\alpha\vee\alpha\cdot\alpha\vee\alpha\cdot\alpha\cdot\alpha\vee\ldots$. Егер мұндағы барлық α-ны ϕ-ге алмастырып, алдымен (1)-ші сонан кейін (5)-ші қасиеттерді бірнеше рет қолданып, ал ең соңында (4)-ші қаиетті қолдансақ, онда $\phi^* = \varepsilon$ шығады.

(13) $\alpha^* = \varepsilon \vee \alpha \vee \alpha\cdot\alpha \vee\ldots$ тепе-теңдіктің екі жағына бірдей $\vee$ амалы арқылы α-ны қоссақ және оның оң жағына (5)-ші қасиетті бір рет қолдансақ, онда $\alpha \vee \alpha^*=\alpha^*$, яғни $\alpha^* = \alpha \vee \alpha^*$.

(14) Жақшаның ішіндегі итерацияны тарқатқаннан соң $(\alpha^*)^*=(\varepsilon\vee\alpha\vee\alpha\cdot\alpha\vee\alpha\cdot\alpha\cdot\alpha\vee\ldots)^*$. Сыртқы итерацияны тарқатып және (2)-ші, (4)-ші, (5)-ші, (11)-ші, (13)-ші қасиеттерді бірнеше рет қолданып, бізге керек $(\alpha^*)^* = \alpha^*$ алуға болады.

Осы қасиеттерді пайдаланып күрделі регулярлық өрнектерді қарапайымдауға болады. Мысалы, $(0\vee(1(0^*)))$ өрнегі тең мәнді $0\vee 10^*$ өрнегіне қарапайымдалады.

Ары қарай біз регулярлық өрнекті және ол анықтайтын регулярлық жиынды бір-бірінен, егер олар түсінбеушілікке келтірмесе, ажыратпаймыз. Мысалы, осы келісім бойынша, a символы $\{a\}$ жиынын береді.

II.1.2.1-мысал.

$\varnothing^* \mathrel{\underset{|}{=}} \varepsilon$, себебі $\varnothing^*=\varepsilon \vee \varnothing \vee \varnothing\varnothing \vee \varnothing\varnothing\varnothing \ldots = \varepsilon \vee \varnothing = \varepsilon \Rightarrow \varnothing^* = \varepsilon$.

Енді регулярлық өрнектерден шығатын регулярлық теңдеулерді қарастырамыз.

II.I.2.1-анықтама. *Регулярлық теңдеу* деп айнымалылары регулярлық жиын, ал коэффициенттері регулярлық өрнек болатын теңдеуді айтамыз. Мысалы, мынадай теңдеуді

$$X = \alpha X \vee \beta, \quad (1)$$

айнымалы X қайсы бір әліпбидегі регулярлық жиын, ал коэфициенттер α мен β сол әліпбидегі регулярлық өрнектер.

II.I.2.1-теорема:

1) Егер (1)-ші теңдеудегі коэфициент α бос тізбені қамтымайтын болса, яғни $\varepsilon \notin \alpha$, онда осы теңдеудің шешімі жалғыз $\alpha^* \beta$ болады.

2) Егер (1)-ші теңдеудегі коэфициент α бос тізбені қамтыйтын болса, яғни $\varepsilon \in \alpha$, онда осы теңдеудің шешімі $\alpha^*(\beta \vee \gamma)$ болады.

Дәлелдеу:

1) Егер $\varepsilon \notin \alpha$ болса, онда $\forall i$ және $\forall X$ үшін $\alpha^i \beta \subset X$, яғни $\alpha^* \beta \subset X$. Мейлі, $\xi \in X$, $\xi \notin \alpha^* \beta$, ξ – ең қысқа болатындай бар. $\xi = \xi_\alpha \xi'$, мұнда $\xi_\alpha \in \alpha$. Сонда $\xi_\alpha \notin \varepsilon$ және ξ-ден ξ' қысқа. Сондықтан $\xi' \in \alpha^* \beta \Rightarrow \xi \in \alpha^* \beta \Rightarrow X = \alpha^* \beta$.

Ал енді $\alpha^* \beta$ регулярлық өрнегін теңдеудің оң жағына айнымалы X-тің орнына қойып мынаны алуға болады:

$$\alpha\alpha^* \beta \vee \beta = \alpha^+ \beta \vee \beta = (\alpha^+ \vee \varepsilon)\, \beta = \alpha^* \beta$$

Осылардан (1)-ші теңдеудің шешімі $X = \alpha^* \beta$ шығады.

2) Мейлі, $\varepsilon \in \alpha$ (яғни $\alpha\alpha^* = \alpha^*$) және $\alpha^*(\beta \vee \gamma)$ теңдеудің шешімі болсын, мұнда γ – кез келген тіл. Бұдан $\alpha^*(\beta \vee \gamma)$ өрнегіне мына теңдеу сәйкес.

$$\alpha\alpha^*(\beta \vee \gamma) \vee \beta = \alpha^+ \beta \vee \alpha^* \gamma \vee \beta = (\alpha^+ \vee \varepsilon)\beta \vee \alpha^* \gamma = \alpha^* \beta \vee \alpha^* \gamma = \alpha^*(\beta \vee \gamma)$$

Демек, (1)-ші теңдеудің шешімі $X = \alpha^{*}(\beta \vee \gamma)$ екендігі дәлелденді. Бұл осы теңдеудің шешімі шексіз көп болатындығын көрсетеді. Осындай жағдайларда теңдеудің *ең кіші жылжымайтын нүктесі* деп аталатын ең кіші шешімін алу керек, ол $X = \alpha^{*}\beta$ жиыны болады.

II.1.2-тапсырмалар:

1. Мына $(a \vee b \vee ab)^{*}$ регулярлық өрнегін қарапайымдаңыз.
2. Мына $(a^{*}b)^{*} \vee (b^{*}a)^{*}$ регулярлық өрнегін қарапайымдаңыз.
3. Мына $(ba \vee a^{*}ab)^{*}$ регулярлық өрнегін қарапайымдаңыз.
4. Мына $(b^{+}a)^{*}b \vee 1)b^{*}$ регулярлық өрнегін қарапайымдаңыз.
5. Мына $Z = \varphi Z \vee \psi$ регулярлық теңдеудің шешімін табыңыз.
6. Мына $X = (a \vee \beta)X$ регулярлық теңдеудің шешімін табыңыз.
7. Мына $X = (a \vee \beta)X \vee \beta$ регулярлық теңдеудің шешімін табыңыз.
8. Мына $X = (a \vee \beta)^{*}X$ регулярлық теңдеудің шешімін табыңыз.

II.1.2-сұрақтар:

1. Регулярлық өрнектер $((a \vee b)^{*} \vee aa)^{*}$ және $(aa \vee b \vee ab)^{*}$ тең бе?
2. Регулярлық өрнектер $(ab \vee b)^{*}(a \vee b)$ және $(a \vee b)^{*}(a \vee b)^{*}$ тең бе?
3. $(b \vee cd^{*}a)^{*}cd^{*}$, $b^{*}c(d \vee ab^{*}c)^{*}$ регулярлық өрнектер тең бе?
4. Қай кезде регулярлық теңдеудің шешімі біреу ғана болады?
5. Қай кезде регулярлық теңдеудің шешімі көп болады?
6. Регулярлық теңдеулердің шешімін қандай әдіспен табады?
7. Келесі регулярлық өрнектермен қандай тілдер туындайды: $(0^{*}1^{*})0$; $(01^{*})0$; $(00 \vee 11 \vee (01 \vee 10)^{*}(00 \vee 11)$?
8. Мына регулярлық өрнектерді қалай қарапайымдауға болады:

   $(00^{*})0 \vee (00)^{*}$?

   $(0 \vee 1)(\varepsilon \vee 00)^{+} \vee (0 \vee 1)$?

   $(aa \vee \varepsilon)(a \vee b)\ ab$?

   $a^{*}(a \vee \mathrm{b})\ )(\varepsilon \vee a)b$?

   $a(a \vee b)(a \vee b)b$?
9. Регулярлық өрнектер $(p^{*}q^{*})^{*}$ және $(q^{*}p^{*})^{*}$ эквивалентті ме?
10. Регулярлық өрнектер $p(qp)^{*}$ және $(pq)^{*}p$ эквивалентті ме?
11. Регулярлық өрнектер $p^{*}(p \vee q)^{*}$ және $(p \vee q)^{*}$ эквивалентті ме?

II.1.3. Регулярлық теңдеулер жүйесі

Бұл параграфта регулярлық тілді анықтайтын регулярлық теңдеулер жүйелері қарастырылады, олардың стандарттық түрі көрсетіледі және оларды шешу әдісі беріледі, мысалдар беріледі, тапсырмалар ұсынылады, сұрақтар қойылады [1,16-21,25,29,32].

Алдымен екі теңдеулер жүйесін қарастырайық:

$$\begin{cases} X = \alpha_1 X \vee \alpha_2 Y \vee \alpha_3 \\ Y = \beta_1 X \vee \beta_2 Y \vee \beta_3 \end{cases} \qquad (1)$$

мұнда α_i және β_i барлық $i = 1,2,3$ үшін ***T*** әліпбиіндегі регулярлық өрнектер болады.

Теңдеулер жүйесін шешу үшін мына қадамдар орындалады:

1-қадам. Мына өрнек $X = \alpha_1{}^*(\alpha_2 Y \vee \alpha_3)$ табылады.

2-қадам. Мына өрнек $Y = \beta_2{}^*(\beta_1 X \vee \beta_3)$ табылады.

3-қадам. 1-ші қадамдағы өрнектегі X-ты 2-ші қадамдағы өрнекке қойғанда $Y = \beta^*_2(\beta_1(\alpha_1{}^*(\alpha_2 Y \vee \alpha_3)) \vee \beta_3)$ алынады.

4-қадам. 3-ші қадамдағы Y-тен құтылғанда, мына өрнек $Y = \beta^*_2(\beta_1(\alpha_1{}^*(\alpha_2{}^*\alpha_3)) \vee \beta_3)$ алынады.

5-қадам. 4-ші қадамдағы Y-тің мәнін 1-ші қадамдағы өрнекке қойғанда $X = \alpha_1{}^*(\alpha_2(\beta_2{}^*(\beta_1(\alpha_1{}^*(\alpha_2{}^*\alpha_3)) \vee \beta_3)) \vee \alpha_3)$ шығады.

Демек, берілген теңдеулер жүйесінің шешімдері келесі:

$$X = \alpha_1{}^*(\alpha_2(\beta_2{}^*(\beta_1(\alpha_1{}^*(\alpha_2{}^*\alpha_3)) \vee \beta_3)) \vee \alpha_3)$$

$$Y = \beta^*_2(\beta_1(\alpha_1{}^*(\alpha_2{}^*\alpha_3)) \vee \beta_3)$$

Енді регулярлық теңдеулердің стандартты жүйесін қарастыруға болады.

II.1.3.1-анықтама. Белгісіздер жиыны $\Delta = \{X_1, X_2, \ldots, X_n\}$ және регулярлық коэффициенттері бар n регулярлық теңдеулер жүйесін *стандартты жүйе* дейміз, егер ол мынадай болса:

$$\begin{cases} X_1 = \alpha_{10} \vee \alpha_{11} X_1 \vee \alpha_{12} X_2 \ldots \vee \alpha_{1n} X_n \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ X_n = \alpha_{n0} \vee \alpha_{n1} X_1 \vee \alpha_{n2} X_2 \ldots \vee \alpha_{nn} X_n \end{cases}$$

Мұнда барлық коэффициенттер α_{ij} – Δ жиынымен қиылыспайтын $\boldsymbol{T}$ әліпбиіндегі регулярлық өрнектер, $\Delta \cap \boldsymbol{T} \neq \varnothing$.

Теңдеулердің коэфициенттері α_{ij} өрнегі болады. Ескерте кетсек, егер $\alpha_{ij} = \phi$ (мұндай регулярлық көрнек болуы мүмкін), онда X_i белгісізі үшін теңдеуде X_j белгісізін қамтитын мүше болмайды. Соған ұқсас, егер $\alpha_{ij} = \varepsilon$ болса, онда X_i белгісізі үшін теңдеуде X_j белгісізін қамтитын мүше – ол жай ғана X_j болады. Басқа сөзбен айтқанда, ϕ және ε коэфициенттері әдеттегі сызықтық теңдеулердегі сәйкес 0 және 1 коэфициенттерінің рөлін атқарады.

Енді регулярлық коэфициентті теңдеулердің стандартты жүйесін шешудің алгоритмін қарастырайық:

Кіріс. Белгісіздері $\Delta = \{X_1, X_2, \ldots, X_n\}$ жиыны, ал коэффициенттері Δ жиынымен қиылыспайтын $\boldsymbol{T}$ әліпбиіндегі регулярлық өрнектер болатын n регулярлық теңдеулі *стандартты* жүйесі.

Шығыс. Стандартты жүйенің $X_1 = \alpha_1, \ldots, X_n = \alpha_n$ шешімі, мұнда α_i – $\boldsymbol{T}$ әліпбиіндегі регулярлық өрнек, $i = 1, 2, \ldots, n$.

Әдіс. Стандартты жүйені шешу әдісі сызықтық теңдеулер жүйесін шешуге арналған Гаусс әдісіне ұқсас келеді.

1-қадам. $i=1$ болсын.

2-қадам. Егер $i=n$, онда 4-қадамға көшу, әйтпесе 1-лемманың тепе-теңдіктері көмегімен X_i үшін теңдеуді $X_i = \alpha X_i \vee \beta$ түрінде жазу, мұнда α – $\boldsymbol{T}$ әліпбиіндегі регулярлық өрнек, β – $\beta_0 \vee \beta_i X_{i+1} \vee \ldots \vee \beta_n X_n$ түріндегі регулярлық өрнек, ал барлық β_i – $\boldsymbol{T}$ әліпбиіндегі регулярлық өрнектер. Онан кейін, $X_{i+1}, \ldots, X_n$ үшін теңдеулердің оң жағындағы X_i-ді регулярлық өрнек $\alpha^* \beta$-мен ауыстыру керек.

3-қадам. i-ді 1-ге көбейту және 2-қадамға оралу.

4-қадам. X_n үшін теңдеуді $X_n=\alpha X_n \vee \beta$ түрінде жазу, мұнда α және β – ***T*** әліпбиіндегі регулярлық өрнектер. (2-қадамды орындағаннан кейін әрбір $i<n$ үшін X_i теңдеуінің оң жағында $X_1,\ldots,X_{i-1}$ айнымалылар болмайды. Айта кетсек, 2-қадамда бұл қасиетке X_n үшін теңдеу де ие болады). 5-қадамға көшу (бұл кезде $i=n$).

5-қадам. X_i үшін теңдеу $X_i=\alpha X_i \vee \beta$ түрінде болады, мұнда α және β – ***T*** әліпбиіндегі регулярлық өрнектер. Шығысында $X_i = \alpha^*\beta$ жазу және $X_{i-1},\ldots,X_1$ үшін теңдеулерде X_i орнына $\alpha^*\beta$-ні қою.

6-қадам. Егер $i = 1$, онда тоқтау керек, әйтпесе i-ді 1-ге кемітіп 5-қадамға оралу керек.

II.1.3.2-анықтама. Мейлі ***Q*** - белгісіздері $\Delta = \{X_1, X_2,\ldots,X_n\}$ жиыны және коэффициенттері Δ жиынымен қиылыспайтын ***T*** әліпбиіндегі регулярлық өрнектер болатын n регулярлық теңдеулі стандартты жүйесі. Δ жиынының ***T*** әліпбиіндегі тілдер жиынына f бейнесі ***Q*** жүйесінің шешімі деп аталады, егер әрбір теңдеуде X-тің орнына $f(X)$-ті әрбір $X\in\Delta$ үшін қойғанда теңдеулер жиындардың тепе-теңдіктерге айналса.

f: $\Delta\rightarrow\vartheta(\boldsymbol{A}^*)$ бейнесі ***Q*** жүйесінің *ең кіші жылжымайтын нүктесі* деп аталады, егер барлық $X\in\Delta$ үшін f шешімі мен кез келген басқа g шешімі арасында $f(X) \subseteq g(X)$ болса, мұнда $\vartheta(\boldsymbol{A}^*)$ – $\boldsymbol{A}^*$ жиынының барлық ішжиындарының жиыны.

Келесі екі лемма *ең кіші жылжымайтын нүктесі* туралы болады.

II.1.3.1-лемма. Белгісіздері Δ болатын әрбір стандартты теңдеулер жүйесі ***Q*** ең кіші жылжымайтын нүктеге ие болады.

Дәлелдеу. Барлық $X\in\Delta$ үшін $f(X)=\{\omega$: $\omega\in g(X)$ стандартты жүйе ***Q***-дің барлық шешімдері g үшін$\}$. Шешім f және барлық шешімдер g үшін $f(X)\subseteq g(X)$ орындалатынын көреміз. Сонымен, f – стандартты жүйе ***Q***-дің жалғыз ең кіші жылжымайтын нүктесі.

II.1.3.2-лемма. Мейлі $\boldsymbol{Q}_1$ және $\boldsymbol{Q}_2$ – 1-ші алгоритмнің 2-ші қадамын орындағанға дейінгі және бір рет орындағаннан кейінгі теңдеулер жүйесі. Сонда $\boldsymbol{Q}_1$ және $\boldsymbol{Q}_2$ бірдей ең кіші жылжымайтын нүктеге ие болады.

II.1.3.3-лемма. Мейлі $\boldsymbol{Q}_1$ және $\boldsymbol{Q}_2$ – 1-ші алгоритмнің 5-ші қадамын орындағанға дейінгі және бір рет орындағаннан кейінгі теңдеулер жүйесі. Сонда $\boldsymbol{Q}_1$ және $\boldsymbol{Q}_2$ бірдей ең кіші жылжымайтын нүктеге ие болады.

II.1.3.2-теорема. *I.1.4-алгоритмі* стандартты теңдеулер жүйесінің ең кіші жылжымайтын нүктесін табады.

Дәлелдеу. 5-ші қадамды қолданғаннан кейін барлық i үшін барлық теңдеулер $X_i=\alpha_i$ түріне ие болады, мұнда α_i – $\boldsymbol{T}$ әліпбиіндегі регулярлық өрнек. Бұл жүйенің ең кіші жылжымайтын нүктесі $f(X_i)=\alpha_I$ қылатын f бейнесі болады.

II.1.3-мысалдар:

1. $\begin{cases} X_1=\varepsilon\vee 01^*X_1\vee 11X_2 \\ X_2=00X_1\vee 1X_2\vee 1 \end{cases}$

Шешуі:

$X_1=(01^*)^*(\varepsilon\vee 11X_2)$

$X_2=00(01^*)^*(\varepsilon\vee 11X_2)\vee 1X_2\vee 1$

$X_2=(00(01^*)^*11\vee 1)^*(00(01^*)^*\vee 1$

$X_1=(01^*)^*(\varepsilon\vee 1100(01^*)^*11\vee 1)^*(00(01^*)^*\vee 1))$

2. $\begin{cases} X_1=aX_2\vee b\vee cX_1 \\ X_2=cX_1\vee cX_2\vee ab \end{cases}$

Шешуі:

$X_1=c^*(aX_2\vee b)$

$X_2=c(c^*(aX_2\vee b))\vee cX_2\vee ab$

$X_2=(cc^*a\vee c)X_2\vee cc^*b\vee ab$

$X_2=(cc^*a\vee c)^*(cc^*b\vee ab)$

$X_1=c^*(a(cc^*a\vee c)^*(cc^*b\vee ab)\vee b)$

3. $\begin{cases} X_1=a\vee bX_2\vee cX_1 \\ X_2=c\vee bX_1\vee aX_2 \end{cases}$

Шешуі:

$$X_1=c^*(a\vee bX_2)$$

$$X_2=c\vee b(c^*(a\vee bX_2))\vee aX_2$$

$$X_2=c\vee bc^*a\vee(bc^*b\vee a)X_2$$

$$X_2=(bc^*b\vee a)^*(c\vee bc^*a)$$

$$X_1=c^*(a\vee b(bc^*b\vee a)^*(c\vee bc^*a))$$

II.1.3-тапсырмалар:

1. Регулярлық теңдеулер жүйесін шешіңіз:

$$\begin{cases} X = \alpha_1 X \vee \beta_1 Y \\ Y = \alpha_2 X \vee \beta_2 Y \end{cases}$$

2. Регулярлық теңдеулер жүйесін шешіңіз:

$$\begin{cases} X = \alpha_1 X \vee \beta_1 Y \vee \gamma_1 \\ Y = \alpha_2 X \vee \beta_2 Y \vee \gamma_2 \end{cases}$$

3. Регулярлық теңдеулер жүйесін стандартты түрге келтіріңіз:

$$\begin{cases} X = a_1 X \vee b_1 Y \vee c_1 \\ Y = a_2 X \vee b_2 Y \vee c_2 \end{cases}$$

4. Стандартты регулярлық теңдеулер жүйесін шешіңіз:

$$\begin{cases} X_1 = abX_1 \vee b \vee aX_2 \\ X_2 = aaX_3 \vee bX_1 \\ X_3 = abX_2 \vee c \vee cX_3 \end{cases}$$

5. Стандартты регулярлық теңдеулер жүйесін шешіңіз:

$$\begin{cases} X_1 = a_{10} \vee a_{11} X_1 \vee a_{12} X_2 \vee a_{13} X_3 \\ X_2 = a_{20} \vee a_{21} X_1 \vee a_{22} X_2 \vee a_{23} X_3 \\ X_3 = a_{30} \vee a_{31} X_1 \vee a_{32} X_2 \vee a_{33} X_3 \end{cases}$$

6. Стандартты регулярлық теңдеулер жүйесін шешіңіз:

$$\begin{cases} X_1 = a_{10} \vee a_{11} X_1 \vee a_{12} X_2 \\ X_2 = a_{20} \vee a_{21} X_1 \vee a_{22} X_2 \end{cases}$$

7. Регулярлық теңдеулер жүйесін шешіңіз:

$$\begin{cases} X_1 = 001^* X_1 \vee 1X_3 \vee 0X_2 \\ X_2 = 11X_2 \vee 0X_3 \vee 01X_1 \\ X_3 = 01X_3 \vee (0 \vee 1)X_1 \end{cases}$$

8. Регулярлық теңдеулер жүйесін шешіңіз:

$$\begin{cases} X_1 = 00X_2 \vee 11X_2 \\ X_2 = 0X_3 \vee 01X_2 \\ X_3 = 1X_1 \vee 0X_3 \end{cases}$$

9. Регулярлық теңдеулер жүйесін шешіңіз:

$$\begin{cases} X_1 = (010^* \vee \varepsilon)X_3 \vee 0X_1 \\ X_2 = X_3 \vee 1X_1 \vee 00 \\ X_3 = \varepsilon \vee 1X_2 \vee (0 \vee 1)X_3 \end{cases}$$

10. Регулярлық теңдеулер жүйесін шешіңіз:

$$\begin{cases} X_1 = abX_1 \vee b \vee aX_2 \\ X_2 = aaX_3 \vee bX_1 \\ X_3 = abX_2 \vee c \vee cX_3 \end{cases}$$

II.1.3-сұрақтар:

1. Мына регулярлық теңдеулер жүйесін шешу үшін қандай қадамдар жасау керек?

$$\begin{cases} X = (\alpha_1 \vee \alpha_2 \beta_2{}^* \beta_1)^* (\alpha_3 \vee \alpha_2 \beta_2{}^* \beta_3) \\ Y = (\beta_2 \vee \beta_1 \alpha_1{}^* \alpha_2)^* (\beta_3 \vee \beta_1 \alpha_1{}^* \alpha_3) \end{cases}$$

2. Қандай регулярлық теңдеулер жүйесін стандартты дейміз?

3. Регулярлық теңдеулер жүйесінің ең кіші жылжымайтын нүктесі деген не?

4. Регулярлық теңдеулер жүйесінде қандай жағдайда ең кіші жылжымайтын нүктені таба аламыз?

5. Мына $X_1 = \alpha X_1 \vee \beta X_2$ және $X_2 = \beta\beta\alpha X_1 \vee \beta\alpha X_2$ регулярлық теңдеулерден құрылған жүйенің шешуі қалай табылады?

II.1.4. Сызықты грамматикалар

Бұл параграфта регулярлық тілдерді тудыратын сызықты грамматикалар және олардың түрлері мен қасиеттері қарастырылады, сонымен қатар, мысалдар беріледі, тапсырмалар ұсынылады, сұрақтар қойылады [1-9,11-13,15-21,22-25,28-32].

Мейлі $\boldsymbol{G} = <N, \boldsymbol{T}, \boldsymbol{P}, S>$ формалды грамматикасы берілсін, мұнда $\boldsymbol{N}$ – бейтерминалдардың ақырлы жиыны, $\boldsymbol{T}$ – терминалдардың ақырлы жиыны, $\boldsymbol{T} \cap \boldsymbol{N} = \emptyset$, $\boldsymbol{P}$ – шығарым ережелерінің ақырлы жиыны, S – *бастапқы бейтерминал*, $S \in \boldsymbol{N}$. Сонда мынадай анықтамаларды беруге болады:

II.1.4-анықтамалар:

1. Егер $\boldsymbol{G}$ грамматикасындағы барлық шығарым ережелері әрбір A, $B \in \boldsymbol{N}$, $\tau \in \boldsymbol{T}^*$ үшін $A \to \tau B$ немесе $A \to \tau$ түрінде берілсе, онда ол оң сызықты *грамматика* деп аталады.

Оң сызықты грамматикамен туындаған тізбелер жиыны оң сызықты *тіл* болады.

2. Егер $\boldsymbol{G}$ грамматикасындағы барлық шығарым ережелері әрбір A, $B \in \boldsymbol{N}$, $\tau \in \boldsymbol{T}^*$ үшін $A \to B\tau$ немесе $A \to \tau$ түрінде берілсе, онда ол сол сызықты *грамматика* деп аталады.

Сол сызықты грамматикамен туындаған тізбелер жиыны сол сызықты *тіл* болады.

3. Егер $\boldsymbol{G}$ грамматикасындағы барлық шығарым ережелері әрбір A, $B \in \boldsymbol{N}$, $\alpha \in \boldsymbol{T}^*$, $\beta \in \boldsymbol{T}^*$ үшін $A \to \alpha B \beta$ немесе $A \to \alpha$ түрінде берілсе, онда ол *сызықты грамматика* болады.

Сызықты грамматикада шығарым үдерісі кезіндегі тізбеде артық бейтерминал болмайды және, егер $\beta = \varepsilon$, онда ол оң сызықты, ал егер $\alpha = \varepsilon$, онда ол оң сызықты грамматика болады. Сызықты грамматикамен туындаған тізбелер жиыны *сызықты тіл* болады.

4. Егер оң сызықты G = <N, T, P, S> грамматикасындағы әрбір ереже $A \to \varepsilon$, $A \to a$ немесе $A \to aB$ түрінде болса, онда бұл грамматика *нормалды пішімде* тұр деп айтады, мұнда $A \in \mathbf{N}$, $B \in \mathbf{N}$, $a \in \mathbf{T}$.

5. Егер сол сызықты G = <N, T, P, S> грамматикасындағы әрбір ереже $A \to \varepsilon$, $A \to a$ немесе $A \to Ba$ түрінде болса, онда бұл грамматика *нормалды пішімде* тұр деп айтады, мұнда $A \in \mathbf{N}$, $B \in \mathbf{N}$, $a \in \mathbf{T}$.

6. Егер сызықты G = <N, T, P, S> грамматикасындағы әрбір ереже $A \to \varepsilon$, $A \to a$, $A \to aB$ немесе $A \to Ba$ түрінде болса, онда бұл грамматика *нормалды пішімде* тұр деп айтады, мұнда $A \in \mathbf{N}$, $B \in \mathbf{N}$, $a \in \mathbf{T}$.

7. Оң сызықты грамматика G = <N, T, P, S> *регулярлық* деп аталады, егер:

(1) $S \to \varepsilon$ ережесінен басқа барлық ережелер $\boldsymbol{A \to aB}$ немесе $\boldsymbol{A \to a}$ түрінде болса, мұнда $\boldsymbol{A}$, $\boldsymbol{B} \in N$, $a \in \boldsymbol{T}$ болса;

(2) $S \to \varepsilon \in \boldsymbol{P}$ болған жағдайда, бастапқы бейтерминал S ережелердің оң жағында кездеспесе.

Енді осы анықталған грамматикалардың және олармер туындаған тілдердің қасиеттері келесі теоремалар арқылы орнатылады:

II.1.4.1-теорема. Әрбір оң сызықты грамматика қандай да бір нормалды пішімдегі оң сызықты грамматикаға эквивалентті.

II.1.4.2-теорема. Егер оң сызықты тіл бос тізбені қамтымаса, онда ол қандай да бір нормалды пішімдегі оң сызықты ε-ережесіз грамматикамен туындайды.

II.1.4.3-теорема. Әрбір оң сызықты грамматика қандай да бір регулярлық грамматикаға эквивалентті.

II.1.4-мысалдар:

1. Оң сызықтық G =< N, T, P, S> грамматикасын қарастырайық, мұндағы $\boldsymbol{N}=\{S, A\}$, $\boldsymbol{T} = \{a, b\}$, $\boldsymbol{P} = \{S \to aA,\ A \to aA,\ A \to b\}$ болсын. Бұл грамматикада мынадай $S \Rightarrow aA \Rightarrow aaA \Rightarrow aaaA \Rightarrow \ldots \Rightarrow a \ldots aaab$ шығарыммен оң сызықты тіл $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{a^n b \colon n = 1, 2, \ldots\}$ туады.

2. Сол сызықтық $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ грамматикасы қарастырайық, мұндағы $\boldsymbol{N}=\{S, A\}$, $\boldsymbol{T} = \{a, b\}$, $\boldsymbol{P} = \{S \to Aa, A \to Aa, A \to b\}$ болсын. Бұл грамматикада мынадай $S \Rightarrow Aa \Rightarrow Aaa \Rightarrow Aaaa \Rightarrow ... \Rightarrow baaa...a$ шығарыммен сол сызықты тіл $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{ba^n: n = 1, 2, ...\}$ туады.

3. a, aaa, $aaaaa$, ... сияқты тізбектер жиынынан тұратын $\{a^{2n-1}\}$ оң сызықты (сол сызықты) тілді қарастырайық. Ол $\boldsymbol{T}=\{a\}$, $\boldsymbol{N}=\{S\}$ және $P=\{S\to a, S\to aaS\}$ ($P=\{S\to a, S\to Saa\}$) жиындарынан тұратын оң сызықты (сол сызықты) $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ грамматикасынан туындайды. Ережелердің түрінен берілген тілдің оң сызықты (сол сызықты) тіл болатындығын көруге болады.

4. $\{a,b\}$ әліпбиінде мына $\boldsymbol{L} = \{a^m b^m a^n b^n: m\geq 0, n\geq 0\}$ тілін қарастырайық. Ешқандай натурал сан k үшін II.1.4.7-теоремасының тұжырымы орындалмайды. Сондықтан $\boldsymbol{L}$ тілі сызықтық емес.

5. Мейлі $\{\omega\in\{a,b\}^*: |\omega|_a=2, |\omega|_b=2\}$ тілі сызықтық грамматикамен туындасын. Осы тіл бос тізбені қамтымайды, кез келген тізбеде a мен b әріптерінің кіру саны 2-ге тең болады. Сондықтан ол оң сызықтық грамматикамен туындайды.

6. Мейлі $\boldsymbol{G}$ грамматикасында мынадай $S\to aA$, $A\to bA$, $A\to c$ ережелер болсын. Осы грамматикамен туындайтын тіл қай топқа жататындығын анықтаймыз. Ол үшін мынадай шешімдерді $S\to aT\to abT\to abc$, $S\to aT\to ac$ құрамыз. Нәтижесінде мынадай оң сызықты $\boldsymbol{L}(\boldsymbol{G})=\{ab^nc : n>=0\}$тілін аламыз.

7. Мейлі $\boldsymbol{G}$ грамматикасында мынадай $S\to 0A|1S|\varepsilon$, $A\to 0B$, $B\to 0S$ ережелер болсын. Осы грамматикамен туындайтын тіл қай топқа жататындығын анықтаймыз. Ол үшін мынадай шешімдерді құрамыз:

$$S\to 0A\to 00B\to 000S\to 0001S\to 0001\varepsilon\to 0001;$$

$$S\to 1S\to 10A\to 100B\to 1000S\to 1000e\to 1000.$$

Нәтижесінде оң сызықты $\boldsymbol{L}(\boldsymbol{G})=\{(0^n1): n\geq 1\}$ тілін аламыз

II.1.4-тапсырмалар:

1. $\{\tau\in\{a,b\}^*: |\tau|_a \geq 2, |\tau|_b\geq 2\}$ тілін тудыратын оң сызықты грамматиканы табу.

2. $S\to E$; $S\to bE$; $S\to caE$; $E\to a$; $E\to bS$ грамматикасына эквивалентті нормалды пішімдегі оң сызықты грамматиканы табу.

3. $\{a^k b^m c^n$: $k\geq 0$, $m\geq 1$, $n\geq 0\}$ тілін тудыратын нормалды пішімдегі оң сызықты ε-ережесіз грамматиканы табыңыз.

4. $\{a,b\}^*$ - $(\{a^n$: $n\geq 0\}\cup\{\{a^k b^m c^n$: $k\geq 0$, $m\geq 1$, $n\geq 0\}$ b^n: $n\geq 0\}$ тілін тудыратын нормалды пішімдегі оң сызықты ε-ережесіз грамматиканы табыңыз.

5. $\{a^n b^n c^m$: $n\geq 1$, $m\geq 1\}$ тілін тудыратын нормалды пішімдегі оң сызықты ε-ережесіз грамматиканы табыңыз.

6. Ережелері мынадай $S\rightarrow 0A|1S|\varepsilon$, $A\rightarrow 0B|1A$, $B\rightarrow 0S|1B$ грамматикамен туындайтын тілді сипаттаңыз.

7. Параметрлері мынадай $\boldsymbol{S} = A$, $\boldsymbol{T} = \{a, b, d\}$, $\boldsymbol{N} = \{A, B, D\}$, $S{=}A$, $\boldsymbol{P}{=}\{A\rightarrow aB, B\rightarrow aB, B\rightarrow b, B\rightarrow bD, D\rightarrow d, D\rightarrow dD, A\rightarrow aD, A\rightarrow a\}$ грамматикамен туындайтын тілді сипаттаңыз.

II.1.4-сұрақтар:

1. $\boldsymbol{L_1}$ оң сызықты және $\boldsymbol{L_2}$ сол сызықты болатын, ал $\boldsymbol{L_1}\cup\boldsymbol{L_2}$ сызықты болмайтын $\boldsymbol{L_1}$ және $\boldsymbol{L_2}$ тілдері бар ма?

2. $\boldsymbol{L_1}$ сызықты және $\boldsymbol{L_2}$ регулярлық болатын, ал $\boldsymbol{L_1}\cap\boldsymbol{L_2}$ сызықты болмайтын $\boldsymbol{L_1}$ және $\boldsymbol{L_2}$ тілдері бар ма?

3. $\boldsymbol{L}(\boldsymbol{G})^R$ тілі ережелер саны $\boldsymbol{G}$ грамматикасындағыдай болатын ешбір оң сызықты грамматикамен туындамайтындай оң сызықты $\boldsymbol{G}$ грамматикасы бар ма?

4. $\boldsymbol{L}(\boldsymbol{G})^R$ тілі ережелер саны $n+1$ болатын (мұнда n – $\boldsymbol{G}$ грамматикасындағы ережелер саны) ешбір оң сызықты грамматикасымен туындамайтындай оң сызықты $\boldsymbol{G}$ грамматикасы бар ма?

5. $\boldsymbol{L}(\boldsymbol{G})^R$ тілі үш бейтерминалды грамматикамен туындамайтындай оң сызықты үш бейтерминалды $\boldsymbol{G}$ грамматикасы бар ма?

6. Мына ережелер $S\rightarrow 0A|1S|\varepsilon$, $A\rightarrow 0B|1A$, $B\rightarrow 0S|1B$ қай топтағы грамматикаға жатады?

7. Мына ережелер $S\rightarrow AB$, $A\rightarrow Aa|bB$, $B\rightarrow a|Sb$ оң сызықты грамматикаға жата ма?

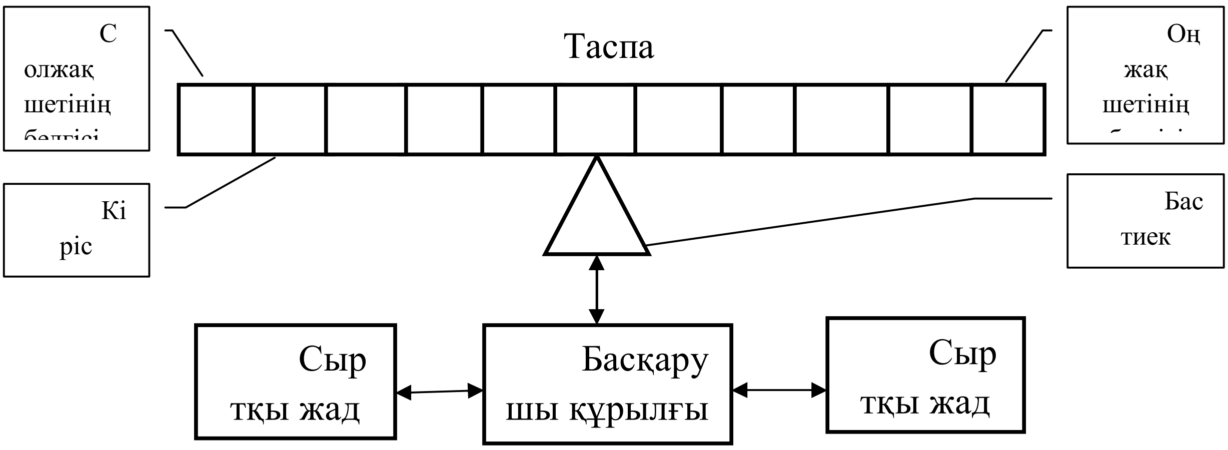
II.2. Регулярлық тілдерді танушы механизмдер

II.2.1. Бейдетерминді және детерминді ақырлы автоматтар

Бұл тарауда бейдетерминді және детерминді ақырлы автоматтар қарастырылады, регулярлық тілдерді ақырлы автоматтар арқылы анықтау сипатталады, сонымен қатар, мысалдар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1-3,5,10,13,19,23,25,26,32].

Ақырлы автоматтар регулярлық тілдерді танығыштар. Алдымен бейдетерминді және детерминді ақырлы автоматтардың формалды анықтамалары беріледі, кейін олармен танылатын тілдер сипатталады, ал соңында олардың эквиваленттілігі дәлелденеді.

Ақырлы автоматтар танығыштардың ішіндегі қарапайымы және ең көп тарағаны болып есептелінеді. Кез келген ақырлы автоматтың құрамында ақырлы кіріс таспасы, ішкі жад, сыртқы жад, оң жақты бастиек, басқарушы құрылғы болады. Ақырлы автоматтың құрылымы төмендегі II.2.1.1-суретте көрсетілген.



II.2.1.1-сурет. Ақырлы автоматтың құрылымы

Ақырлы автомат (АА) бейдетерминді немесе детерминді болуы мүмкін, бірақ оның бастиегі бір жақты және тек оң жаққа жылжи алады. Олардың формалды анықтамалары былай беріледі.

II.2.1.1-анықтама. *Бейдетерминді ақырлы автомат* (БАА) деп мына жетілікті $\boldsymbol{M} = \langle \boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{I}, \boldsymbol{F}, \vdash, \dashv, \Delta \rangle$ айтады, мұнда:

$\boldsymbol{Q}$ – басқарушы құрылғы күйлерінің ақырлы жиыны;

$\boldsymbol{T}$ – кіріс символдарының ақырлы жиыны, $\boldsymbol{Q} \cap \boldsymbol{T} = \varnothing$;

$\boldsymbol{I}$ – басқарушы құрылғының алғашқы күйлерінің жиыны, $\boldsymbol{I} \subseteq \boldsymbol{Q}$;

$\boldsymbol{F}$ – басқарушы құрылғының соңғы күйлерінің жиыны, $\boldsymbol{F} \subseteq \boldsymbol{Q}$;

$\vdash, \dashv$ – таспаның сол және оң жақ шетін белгілегіштер, $\vdash, \dashv \notin \boldsymbol{T}$;

Δ – ауысулар қатынастарының жиыны, $\Delta \subseteq \boldsymbol{Q} \times \boldsymbol{T}^* \times \vartheta(\boldsymbol{Q})$, ал $\vartheta(\boldsymbol{Q})$ – $\boldsymbol{Q}$ жиынының барлық ішжиындырының жиыны;

II.2.1.1-ескертпе. БАА ыңғайлы математикалық абстракция, бірақ, өкінішке орай, оларды моделдеу өте қиын, сондықтан практикада пайдалынатын барлық АА детерминді болады.

Детерминді ақырлы автомат (ДАА) БАА-ың дербес жағдайы.

II.2.1.2-анықтама. Ақырлы автомат $\boldsymbol{M} = \langle \boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{I}, \boldsymbol{F}, \vdash, \dashv, \Delta \rangle$ *детерминді* деп аталады, егер:

(1) алғашқы күйлер жиыны $\boldsymbol{I}$ бір ғана элемент қамтиды;

(2) әрбір ауысу $\langle q, \tau, p \rangle \in \Delta$ үшін $|\tau| = 1$ орындалады;

(3) кез келген күй $q \in \boldsymbol{Q}$ және кез келген кіріс символы $t \in \boldsymbol{T}$ үшін $\langle q, t, p \rangle \in \Delta$ қасиетіне ие болатын бірден көп күй $p \in \boldsymbol{Q}$ бар;

(4) қалған символдар БАА-ндағыдай болады.

II.2.1.2-ескертпелер:

1. Кейде мәні ‘ақиқат’ немесе ‘жалған’ болатын Δ ауысулар қатынасының жиынының орнына мәні $\boldsymbol{Q}$ жиынында жататын ауысулар функциясын пайдаланады, мұнда $\delta\colon \boldsymbol{Q} \times \boldsymbol{T}^* \to \vartheta(\boldsymbol{Q})$ – БАА үшін және $\delta\colon \boldsymbol{Q} \times \boldsymbol{T}^* \to \boldsymbol{Q}$ – ДАА үшін. δ ауысулар функциясынан Δ ауысулар қатынасына былай көшуге болады:

$$\Delta = \{\langle q, \tau, \delta(q, \tau) \rangle : q \in \boldsymbol{Q}, \tau \in \boldsymbol{T}^*\}.$$

2. Ары қарай, біз айырықша ескертусіз, ауысулар қатынасын да, ауысулар функциясын да пайдаланамыз. Сонымен, кез келген $q \in \boldsymbol{Q}$, $p \in \boldsymbol{Q}$ және $\tau \in \boldsymbol{T}^*$ үшін былай да жазуға болады:

1)ауысулар қатынасы: $<q,\tau,\{p\}>$ – БАА үшін, $<q,\tau,p>$ – ДАА үшін;

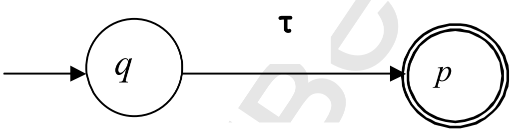
2)ауысулар функциясы: $\delta(q,\tau) = \{p\}$ – БАА үшін, $\delta(q,\tau) = p$ – ДАА үшін.

3. Егер біз ауысулар қатынасының орнына ауысулар функциясын пайдалансақ, АА-ның формалды анықтамасында Δ символын δ символына алмастырып, ал қалған символдардың мәндерін өзгертпей қалдыру керек, яғни $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{I}, \boldsymbol{F}, \vdash, \dashv, \delta>$ деп аламыз.

АА-ты әрбір күйді дөңгелекпен, ауысуды бағыттамамен белгілеп *диаграммамен бейнелеуге болады*. $q\in\boldsymbol{Q}$ күйінен $p\in\boldsymbol{Q}$ күйіне баратын бағыттама БАА-ның ауысуы $<q,\tau,p>$ болады. Әрбір алғашқы күйге қысқа бағыттама кіреді, ал әрбір соңғы күйді екіленген дөңгелек белгілейді.

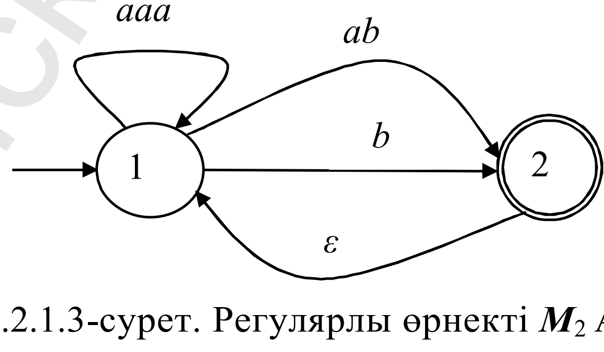
II.2.1.1-мысалдар:

1. Параметрлері $\mathbf{Q}=\{q,p\}$; $\mathbf{T^*}=\{\tau\}$; $\mathbf{I}=\{q\}$; $\mathbf{F}=\{p\}$; $\delta(q, \tau) = p$ болатын бір ауысудан тұратын $\boldsymbol{M}_1$ АА-ның диаграммасы II.2.1.2-суретте көрсетілген.



II.2.1.2-сурет. Бір ауысудан тұратын $\boldsymbol{M}_1$ АА-тың диаграммасы.

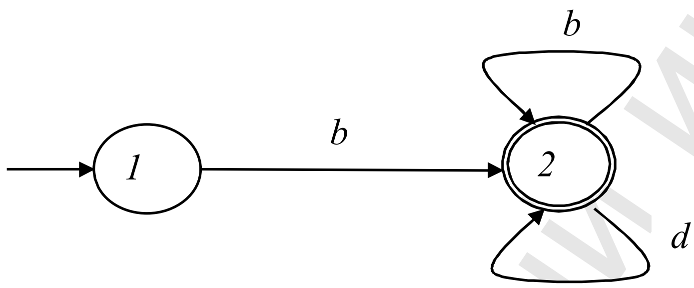
2. Параметрлері $\Delta = \{<1, aaa, 1>, <1, ab, 2>, <1, b, 2>, <2, \varepsilon, 1>\}$, $\boldsymbol{Q}=\{1,2\}$, $\boldsymbol{T}=\{a,b\}$, $\boldsymbol{I}=\{1\}$, $\boldsymbol{F}=\{2\}$ болатын $\boldsymbol{M}_2$ АА-ның диаграммасы II.2.1.3-суретте көрсетілген.



II.2.1.3-сурет. Регулярлы өрнекті $\boldsymbol{M}_2$ АА диаграммасы.

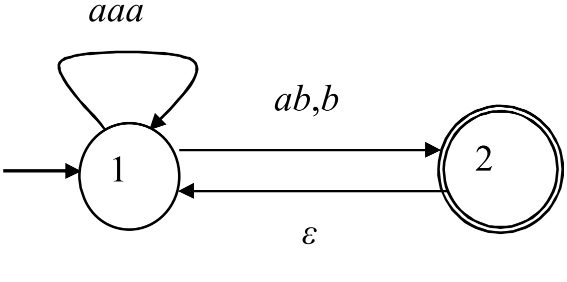
Бұл ауысу диаграммасында қабырғалардың белгілері ретінде регулярлық өрнектер *aaa*, *ab*, *b*, ε пайдаланылған. Диаграмманың мұндай кескінделуі, оны жинақы және көрнекі етіп салуды оңайлатады.

3. Әріптер мен цифрлардан тұратын және цифрдан басталатын айқындағыштарды танитын АА $\boldsymbol{M}_3$ мынадай параметры $\boldsymbol{Q}$={1,2}; $\boldsymbol{T}$={*b*,*d*}; $\boldsymbol{I}$={1}; $\boldsymbol{F}$={2}; δ(*1*,*b*)=*2*, δ(*2*,*b*)=*2*, δ(*2*,*d*)=*2*, мұнда *b* – әріп, *d* – цифра. $\boldsymbol{M}_3$ АА-ның диаграммасы II.2.14-суретте көрсетілген.



II.2.1.4-сурет. Айқындағыштарды танитын $\boldsymbol{M}_3$ АА-тың диаграммасы.

II.2.1.3-ескертпе. Егер диаграммада ортақ басы және ортақ соңы бар бірнеше ауысулар болса, онда мұндай ауысулар параллелді болады. Әдетте, параллелді ауысулар диагрммада бір бағыттамамен бейнеленеді және олардың белгілері үтір арқылы жазылады. II.2.1.5-суретте $\boldsymbol{M}_4$ БАА-ның диаграммасында *ab*, *b* тізбелері үшін параллелді ауысулар бар.



II.2.1.5-сурет. Параллелді ауысулары бар $\boldsymbol{M}_4$ БАА-тың диаграммасы.

АА ауысуларын кесте немесе пәрмен арқылы функция түрінде беруге болады.

II.2.1.3-мысал. $\boldsymbol{Q} = \{q_0, q_1, q_2, q_3\}$, $\boldsymbol{T} = \{t_1, t_2, t_3\}$ жиындарымен анықталған АА-ның ауысулар функциясы II.2.1.1-кестеде берілген.

II.2.1.1-кесте. АА-ның δ ауысулар функциясының мәндері.

δ		Кіріс		
		t_1	t_2	t_3
Күй	q_0	q_2	q_2	q_2
	q_1	q_3	q_0	q_0
	q_2	q_2	q_2	q_2
	q_3	q_3	q_2	q_0

II.2.1.1-кестесіндегі ауысулар функциясын пәрмендер түрінде келесідегідей беруге болады:

$$\delta(q_0, t_1) = q_2,\ \delta(q_0, t_2) = q_2,\ \delta(q_0, t_3) = q_2,$$
$$\delta(q_1, t_1) = q_3,\ \delta(q_1, t_2) = q_0,\ \delta(q_1, t_3) = q_0,$$
$$\delta(q_2, t_1) = q_2,\ \delta(q_2, t_2) = q_2,\ \delta(q_2, t_3) = q_2,$$
$$\delta(q_3, t_1) = q_3,\ \delta(q_3, t_2) = q_2,\ \delta(q_3, t_3) = q_0.$$

II.2.1.1-келісім. АА-ның барлық күйлері арасында q_s алғашқы күй және q_f соңғы күй ерекшеленеді, мұнда s және f сандық айнымалы ретінде емес, басы (*start*) мен соңын (*final*) білдіретін мнемоникалық таңба ретінде қолданылады.

Мейлі, алғашқы күйі $q_s \in \boldsymbol{Q}$, ағымдағы күйі $q \in \boldsymbol{Q}$, соңғы күйі $q_f \in \boldsymbol{F}$, қолданылмаған кіріс тізбесі $\tau \in \boldsymbol{T}^*$ болатындай $\boldsymbol{M}$ ақырлы автоматы берілсін. Сонда мынадай анықтама беруге болады:

II.2.1.3-анықтама:

1. Егер бастиек кіріс тізбесі τ-дың ең сол жағындағы символды қарастырып тұрса, онда $(q_s, \tau) \in \boldsymbol{Q} \times \boldsymbol{T}^*$ қосағы АА-тың *алғашқы кескіні* деп аталады.
2. Егер бастиек кіріс тізбесі τ-дың ағымдағы символын қарастырып тұрса, онда $(q, \tau) \in \boldsymbol{Q} \times \boldsymbol{T}^*$ қосағын АА-тың *ағымдағы кескіні* деп аталады.

3. Егер бастиек τ кіріс тізбесін толық оқыса, онда $(q_f, \varepsilon)\in \boldsymbol{Q}\times\boldsymbol{T}^*$ қосағы АА-тың *соңғы кескіні* деп аталады.

II.2.1.4-ескертпе. Мағына бойынша кескін ақырлы автоматтың "лездегі сипаты" болады. Егер алғашқы тізбенің қарастырылған тілге жататындығын тексеру керектігі таспада берілген болса, онда (q,τ) кескініндегі τ тізбесі таспадағы алғашқы тізбенің әзірше қаралмай қалған бөлігі.

Ақырлы автоматтың қадамы басқарушы құрылғының ағымдағы күйі және осы сәтте бастиек оқитын кіріс символымен анықталады. Қадамның өзі басқарушының күйін өзгерту мен бастиекті бір ұяшыққа оңға жылжытудан тұрады.

$\boldsymbol{M}$ ақырлы автоматының *қадамы* оның $\boldsymbol{Q}\times\boldsymbol{T}^*$ жиынындағы кескіндерінде анықталған $\models_{\boldsymbol{M}}$ бинар қатынасы арқылы беріледі. Мұнда егер автомат алдын ала белгілі болса, онда $\models_{\boldsymbol{M}}$ қатынасындағы $\boldsymbol{M}$ әрпін жазбауға болады.

Айталық, $t\in \boldsymbol{T}$ – кіріс тізбесінің әлі оқылмаған ең сол символы және $q\in\boldsymbol{Q}$, $p\in\boldsymbol{Q}$ үшін $\delta(q, t)= p$ болса, онда тізбе $\tau\in \boldsymbol{T}^*$ үшін автоматтың q күйінде болып, бастиек t символын оқығанда p күйіне көшіп, бастиекті оңға бір ұяшыққа жылжытқанын білдіретін және оның қадамын беретін $(q,\ t\tau)\models(p,\ \tau)$ бинарлы қатынасы орындалады. Мұнда егер $\tau = \varepsilon$, онда кіріс тізбесі *толық оқылды* деп есептелінеді.

II.2.1.4-мысал. II.2.1.3-суреттегі $\boldsymbol{M}_2$ АА диаграммасында мынадай $(1, abba)\models(2, ba)$ қадам бар.

II.2.1.4-анықтама. $\models^k$ деген $\models$ қатынасының k-сыншы дәрежесі болады, егер кез келген i $(1\le i\le k)$ үшін $(q_{i-1},\tau_{i-1})\models(q_i,\tau_i)$ қатынасы орындалатындай $k+1$ кескіндерден тұратын

$$(q_0,\tau_0), (q_1,\tau_1), (q_2,\tau_2),\ldots, (q_{k-1},\tau_{k-1}), (q_k,\tau_k)$$

тізбе бар болса, мұнда $q_0=q_s$, $\tau_0=\tau$, $q_k= q_f$, $\tau_k=\varepsilon$.

Егер кез келген $i\ge 1$ немесе $i\ge 0$ үшін $(q_0,\tau)\models^i(q_i,\varepsilon)$ қатынасы орындалса, онда оны $(q_0,\tau)\models^+(q_i,\varepsilon)$ немесе $(q_0,\tau)\models^*(q_i,\varepsilon)$ деп сәйкес жазуға болады. Мұнда $\models^{+}$ арқылы $\models$ қатынасының транзитивтік тұйықталуын, ал $\models^{*}$ арқылы $\models$ қатынасының рефлексивтік және транзитивтік тұйықталуын белгілейді.

ІІ.2.1.5-анықтама. Егер $(q_s, \tau) \models^{*} (q_f, \varepsilon)$ қатынасы орындалса, онда ақырлы $\boldsymbol{M}$ автоматы τ кіріс тізбесін таниды.

ІІ.2.1.5-мысал. Мейлі $\tau = aaaab$ болсын. Сонда ІІІ.2.1.3-суреттегі $\boldsymbol{M}_2$ АА-та мына қатынастар $(1,aaaab) \models (1,ab)$ және $(1,ab) \models (2,\varepsilon)$ орындалады.

ІІ.2.1.6-анықтама. Егер $\boldsymbol{L}$ тілі тек ақырлы автомат $\boldsymbol{M}$ таныған кіріс тізбелерінен құралса, онда бұл тіл осы $\boldsymbol{M}$ автоматымен танылатын болады және ол $\boldsymbol{L}(\boldsymbol{M})$ арқылы белгіленеді, яғни

$$\boldsymbol{L}(\boldsymbol{M}) \rightleftharpoons \{\tau\colon \tau \in \boldsymbol{T}^{*} \;\&\; (q_s, \tau) \models^{*} (q_f, \varepsilon)\}.$$

ІІ.2.1.1-лемма. Егер $(q_1,x) \models^{*} (q_2,\varepsilon)$ және $(q_2,y) \models^{*} (q_3,\varepsilon)$ рас болса, онда $(q_1,xy) \models^{*} (q_3,\varepsilon)$ рас болады.

Дәлелдеу. Ол үшін АА жұмыс программасының (q_1,x) кескінінен (q_2,ε) кескініне жеткізетін қадам саны бойынша индукция жүргізу керек.

ІІ.2.1.6-мысал. Мейлі $\boldsymbol{M}_6 = <\{q_s, q_1, q_f\}, \{0,1\}, q_s, \{q_f\}, \vdash, \dashv, \delta>$ АА үшін келесі ауысулар қатынастары берілсін:

$<q_s,0,\{q_1\}>, <q_s,1,\{q_s\}>, <q_1,0,\{q_f\}>, <q_1,1,\{q_s\}>, <q_f,0,\{q_f\}>, <q_f,1,\{q_f\}>$

$\boldsymbol{M}_6$ АА құрамында қатар екі нөлі бар барлық нөл мен бірден тұратын тізбелерді таниды. Мұндағы күйлерді былай түсіндіруге (интерпретациялауға) болады:

q_s – алғашқы күйі «қатар тұрған екі нөл әлі пайда болған жоқтығын және алдыңғы цифр нөл еместігін» білдіреді;

q_1 – күйі «қатар тұрған екі нөл әлі пайда болған жоқтығын және алдыңғы цифр нөл екендігін» білдіреді;

q_f – соңғы күйі «қатар тұрған екі нөл пайда болғандығын» білдіреді.

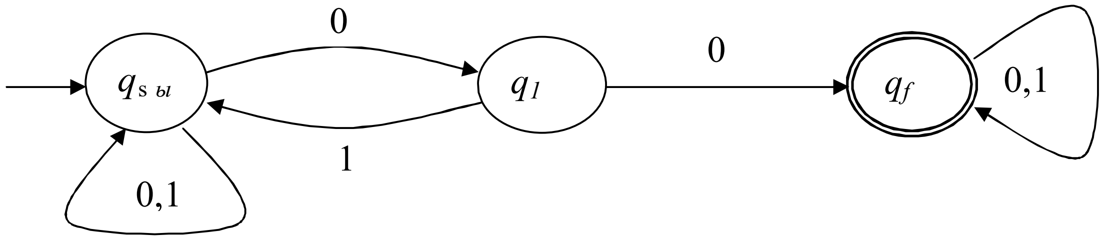
$\boldsymbol{M}_6$ АА соңғы q_f күйіне түскеннен кейін осы күйде қалатындығын байқаймыз.

01001 кіріс тізбесі үшін (q_0, 01001) кескінінен басталатын мүмкін болатын жалғыз кескіндер тізбесі мынадай болады

$(q_s, 01001) \models (q_1, 1001) \models (q_s, 001) \models (q_1, 01) \models (q_f, 1) \models (q_f, \varepsilon)$

Сонымен, 01001 кіріс тізбесі $\boldsymbol{M}_6$ АА танитын $\boldsymbol{L}(\boldsymbol{M}_6)$ тіліне жатады, яғни $01001 \in \boldsymbol{L}(\boldsymbol{M}_6)$.

Осы автоматтың диаграммасы II.2.1.5-суретте көрсетілген.



II.2.1.5-сурет. $\boldsymbol{M}_{5\,\text{АА}}$-тың диаграммасы.

II.2.1.7-анықтама:

1. АА *жолы* (path) деп $<q_0, r_1, q_1, r_2, ..., q_n>$ кортежін айтамыз, мұнда $n \geq 0$, $r_i = < q_{i-1}, \tau_i, q_i > \in \Delta$ әрбір i, $1 \leq i \leq n$ үшін, q_0 – *жол басы*, q_n – *жол аяғы*, n – *жол ұзындығы*, $\tau_1 ... \tau_n$ – *жол белгісі*.

2. *Жол табысты* деп аталады, егер оның басы $\boldsymbol{I}$-ге кірсе, ал соңы $\boldsymbol{F}$-ке кірсе.

II.2.1.5-ескертпе. Кез келген күй $q \in \boldsymbol{Q}$ үшін $<q>$ жолы бар. Оның басы мен аяғы тең, ал белгісі ε болады.

II.2.1.7-мысал. II.2.1.3-суреттегі $\boldsymbol{M}_2$ АА-ын қарастырайық. Мейлі $\tau = baaab$. Сонда $<1,<1,b,2>,2,<2,\varepsilon,1>,1,<1,aaa,1>,1,<1,b,2>,2>$ жолы табысты болады. Оның белгісі *baaab*, ал ұзындығы 4, яғни:

$$q_0=1,\ q_1=2,\ q_2=1,\ q_3=1,\ q_4=2.$$

$$r_1=<1,b,2>,\ r_2=<2,\varepsilon,1>,\ r_3=<1,aaa,1>,\ r_4=<1,b,2>;$$

$$\tau_1=b,\ \tau_2=\varepsilon,\ \tau_3=aaa,\ \tau_4=b.$$

«Жол» деген түсінікті қолданып бұрын жоғарыда берілген танылатын тізбенің және тілдің баламалы анықтамасын берейік.

II.2.1.8-анықтама.

1. $\tau \in \boldsymbol{T}^*$ тізбесі $\boldsymbol{M}$ АА-мен *танылады* (*is recognized*), егер ол қандай да бір жолдың белгісі болса.

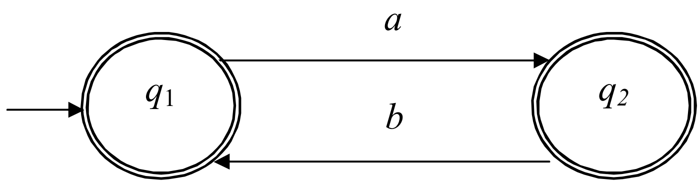
2. $\boldsymbol{M}$ АА $\boldsymbol{L}(\boldsymbol{M})$ тілін *таниды* (*recognizes*), егер ол тек табысты жолдардың белгілерінен тұрса.

II.2.1.6-ескертпе. Егер $\boldsymbol{I} \cap \boldsymbol{F} \neq \emptyset$, онда $< \boldsymbol{Q}, \boldsymbol{T}, \vdash, \dashv, \boldsymbol{I}, \boldsymbol{F}, \Delta >$ АА-пен танылатын тіл ε бос тізбесін қамтиды.

II.2.1.8-мысал. Егер АА $\boldsymbol{M_7} = < \boldsymbol{Q}, \boldsymbol{T}, \vdash, \dashv, \boldsymbol{I}, \boldsymbol{F}, \Delta >$ былай берілсе $\boldsymbol{Q} = \{q_1, q_2\}$, $\boldsymbol{T} = \{a, b\}$, $\boldsymbol{I} = \{q_1\}$, $\boldsymbol{F} = \{q_1, q_2\}$, $\Delta = \{<q_1, a, q_2>, <q_2, b, q_1>\}$, онда ол детерминді болады және келесі тілді таниды:

$$\boldsymbol{L}(\boldsymbol{M_7}) = \{(ab)^n : n \geq 0\} \cup \{(ab)^n a : n \geq 0\}.$$

Бұл автоматтың диаграммасы II.2.6-суретте көрсетілген.

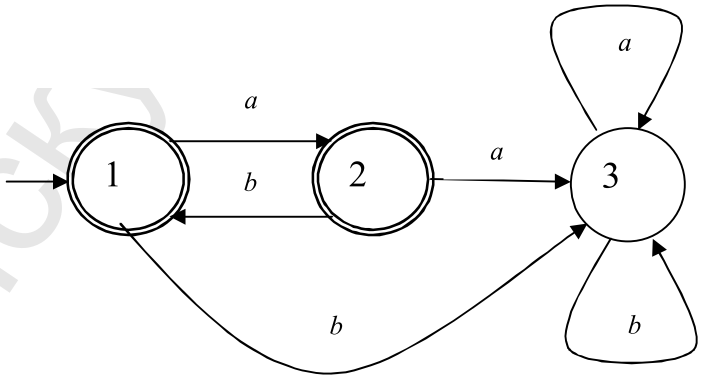


II.2.6-сурет. $\boldsymbol{M_6}$ АА-тың диаграммасы.

II.2.1.9-анықтама. ДАА $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, \vdash, \dashv, \boldsymbol{I}, \boldsymbol{F}, \Delta>$ *толық* деп аталады, егер әрбір күй $q \in \boldsymbol{Q}$ үшін және әрбір символ $t \in \boldsymbol{T}$ үшін $<q, t, p> \in \Delta$, яғни $\delta(q,t) = p$ болатындай бір $p \in \boldsymbol{Q}$ күйі табылады.

II.2.1.9-мысал. Параметрлері $\boldsymbol{Q}=\{1,2,3\}$, $\boldsymbol{T}=\{a,b\}$, $\boldsymbol{I}=\{1\}$, $\boldsymbol{F}=\{1,2\}$, $\Delta = \{<1, a, 2>, <1, b, 3>, <2, a, 3>, <2, b, 1>, <3, a, 3>, <3, b, 3>\}$ болатын толық автомат $\boldsymbol{M_8}$ диаграммасы II.2.1.7-суретте көрсетілген.

.



II.2.1.7-сурет. $\boldsymbol{M_8}$ АА-ның диаграммасы.

II.2.1-тапсырма.

1. $\{\alpha\beta$: $\alpha\in\{a,b\}^*$, $\beta\in\{a,b\}^*\}$ тілін танитын АА-ты құрыңыз.

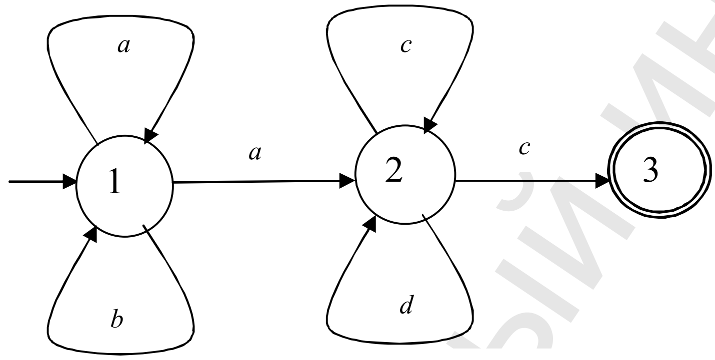
2. $\{a,b\}^* \setminus (\{a^n$: $n\geq 0\}\cup\{b^n$: $n\geq 0\})$ тілін танитын АА-ты құрыңыз.

3. $\{a\xi b$: $\xi\in\{a,b\}^*\cup\{b\xi a$: $\xi\in\{a,b\}^*\}$ тілін танитын АА-ты құрыңыз.

4. $\{\tau\in\{a,b\}^*$: $|\tau|_a\geq 3\}$ тілін танитын АА-ты құрыңыз.

5. $\{a^m b^n a^m b^n$: $m,n \geq 1\}$ тілін танитын АА-ты құрыңыз.

6. $(1,\ abaa) \models^* (q,\ \tau)$ шартын қанағаттандыратын келесі II.2.1.8-суреттегі $\boldsymbol{M_9}$ АА-тың барлық $(q,\ \tau)$ кескіндерін көрсетіңіз.



II.2.1.8-сурет. $\boldsymbol{M_9}$ АА-тың диаграммасы.

7. Мына түрде берілген автоматтың тактысын анықтаңыз

$$\boldsymbol{M} = <\{q_0, q_1, q_2, q_f\}, \{a, b, c\}, \delta, q_0, \{q_f\}>,$$

мұнда $\delta(q_0,a)=\{q_1, q_2\}$, $\delta(q_1,a)=\{q_1\}$, $\delta(q_1,b)=\{q_f\}$, $\delta(q_2,c)=\{q_f\}$,

$$\boldsymbol{L}(\boldsymbol{M}) = \{ac\}\cup\{a^n b: n\geq 1\}.$$

8. Мына тіл $(a\vee b)^*(aab\vee abaa\vee abb)(a\vee b)^*$ үшін толық детерминді ақырлы автоматты құрыңыз.

9. Мына тіл $(b\vee c)((ab)^*c\vee(ba)^*)^*$ үшін толық детерминді ақырлы автоматты құрыңыз.

10. Мына тіл $(b\vee c)^*((a\vee b)^*c(b\vee a)^*)^*$ үшін толық детерминді ақырлы автоматты құрыңыз.

II.2.1.1-сұрақтар:

1. Мына $(q_1,\ \alpha\beta) \models^* (q_2,\ \beta)$ және $\neg\ (q_1,\ \alpha\delta) \models^* (q_2,\ \delta)$ қатынастары орындалатындай АА, q_1 мен q_2 күйлері және α, β, δ тізбелері бар ма?

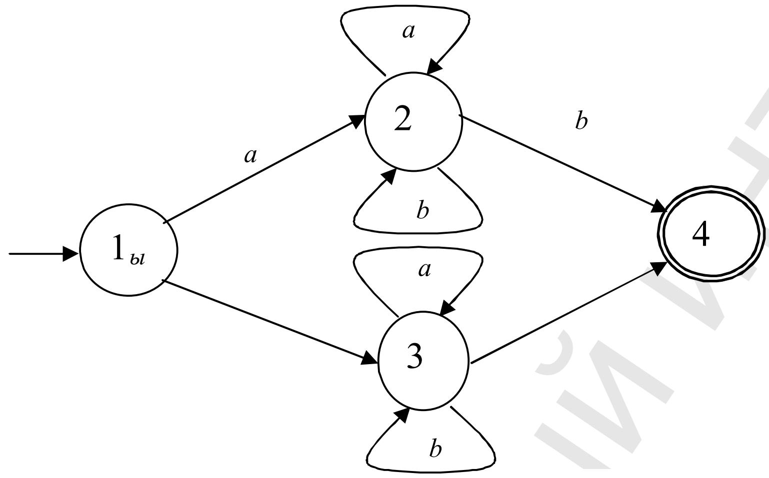
2. $\models^*$ мағынасында $|\boldsymbol{Q}|$, $|\boldsymbol{T}|$, $|\Delta|$,$|\tau|$және $(q,\ \tau)$ кескінінен қолжетімді кескіндер саны қандай байланыста болады?

3. $abab)\vee(aba)^*$ регулярлық өрнегімен туындаған тілді қандай автоматпен тануға болады?

4. Мына $\delta(q, a) = p$, $\delta(q, a) = \{p\}$, $\delta(q, a) = \varnothing$ функциялардың қайсысы детерминді ақырлы автоматтың ауысу функциясы болады?

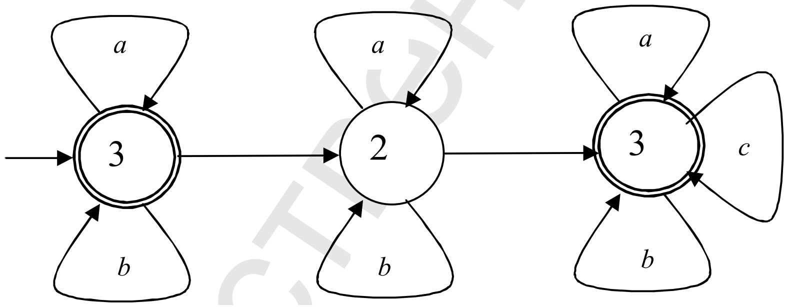
5. Бейдетерминді ақырлы автомат пен детерминді ақырлы автоматтың айырмашылығы неде?

6. II.2.1.9-суреттегі ақырлы автомат $\boldsymbol{M_{10}}$ детерминді бола ма?



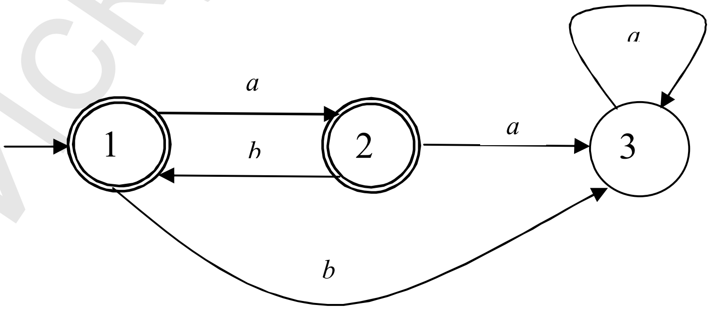
II.2.1.9-сурет. $\boldsymbol{M_{10}}$ АА-тың диаграммасы.

7. Төменде II.2.1.10-суреттегі $\boldsymbol{T} = \{a, b, c\}$ әліпбиі бар детерминді ақырлы автомат $\boldsymbol{M_{11}}$ толық бола ма?



II.2.1.10-сурет. $\boldsymbol{M_{11}}$ АА-тың диаграммасы

8. Төменде II.2.1.11-суреттегі $\boldsymbol{T} = \{a, b\}$ әліпбиі бар детерминді ақырлы автомат $\boldsymbol{M_{12}}$ толық бола ма?



II.2.1.11-сурет. $\boldsymbol{M_{12}}$ АА-тың диаграммасы.

II.2.2. Бейдетерминді және детерминді ақырлы автоматтар эквиваленттілігі

Бұл параграфта бір тілді ғана танитын бейдетерминді және детерминді ақырлы автоматтардың эквиваленттілігі қарастырылады, мысалдар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1-3,5,10,13,19,23,25,26,32].

Ақырлы автоматтар теориясының ең маңызды ғылыми нәтижелерінің бірі бейдетерминді ақырлы автоматтар танитын тілдер тобы толық анықталған детерминді ақырлы автоматтар танитын тілдер тобына сәйкес келетіндігінен тұрады. Ол үшін танитын бейдетерминді ақырлы автомат пен детерминді ақырлы автомат эквиваленттілігі туралы теорема дәлелдейміз.

II.2.2.1-теорема. $\boldsymbol{L}$ тілін танитын кез келген БАА $\boldsymbol{M}$ үшін осы тілді танитын ДАА $\boldsymbol{M'}$ әрдайым тұрғызылады.

Дәлелелдеу. Мейлі регулярлық $\boldsymbol{L}$ тілін танитын БАА $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, \delta, \boldsymbol{F}, \vdash, \dashv, \delta>$ берілсін, яғни $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}$, онда осы $\boldsymbol{L}$ тілін танитын ДАА $\boldsymbol{M'}=<\boldsymbol{Q'},\boldsymbol{T'},q'_0,\boldsymbol{F'},\vdash,\dashv,\delta'>$, яғни $\boldsymbol{L}(\boldsymbol{M'})=\boldsymbol{L}$ болатындай келесі жолмен тұрғызылады:

(1) $\boldsymbol{Q'} = \vartheta(\boldsymbol{Q}) = 2^{\boldsymbol{Q}}$, яғни $\boldsymbol{M}$ БАА-ның күйлер жиынының барлық ішжиындары $\boldsymbol{M'}$ ДАА-ның күйлері болады;

(2) $q_0' = \{q_0\}$;

(3) $\boldsymbol{F'}$ жиыны $\boldsymbol{F}$ жиынымен қиылыспайтын $\boldsymbol{Q}$ жиынының барлық ішжиындары $\boldsymbol{P}$ жиынынан тұрады, яғни

$$\boldsymbol{F'} = \{\boldsymbol{P}: \forall \boldsymbol{P}(\boldsymbol{P} \subseteq \vartheta(\boldsymbol{Q}) \Rightarrow \boldsymbol{P} \cap \boldsymbol{F} = \varnothing)\};$$

(4) $\boldsymbol{T'} = \boldsymbol{T}$;

(5) $\delta(\boldsymbol{P}, t) = \boldsymbol{P'}$ барлық $\boldsymbol{P} \subseteq \boldsymbol{Q}$ үшін, мұндағы $\boldsymbol{P'}=\{p$: кейбір $q \in \boldsymbol{P}\}$ үшін p күйі $\delta(q,t)$ жиынында қамтылады, яғни $\exists q(q \in \boldsymbol{P} \& p \in \delta(q,t))$.

$\boldsymbol{M'}$ автоматының күйі $[q_1, q_2, ..., q_i] \in \boldsymbol{Q'}$ арқылы белгіленеді, мұнда $q_1, ..., q_i \in \boldsymbol{Q}$; $q_0' = [q_0]$. Осыдан $\delta([q_1, ..., q_i], t) = [p_1, ..., p_j]$ тек қана және тек қана сонда

$$\delta([q_1, ..., q_k], t) = \bigcup_{i=1}^{k} \delta(q_i, t) = \{p_1, ..., p_j\}.$$

Енді келесі тұжырым $(\boldsymbol{P}, \tau) \models_M^i (\boldsymbol{P}', \varepsilon)$ тек қана және тек қана сонда, қашан $\boldsymbol{P}' = \{p\text{: } \delta(q,\tau) \models_M^i (p,\varepsilon)$ кейбір $q \in \boldsymbol{P}$ үшін$\}$ болғанда орындалатындығын i бойынша индукциямен дәлелдейік.

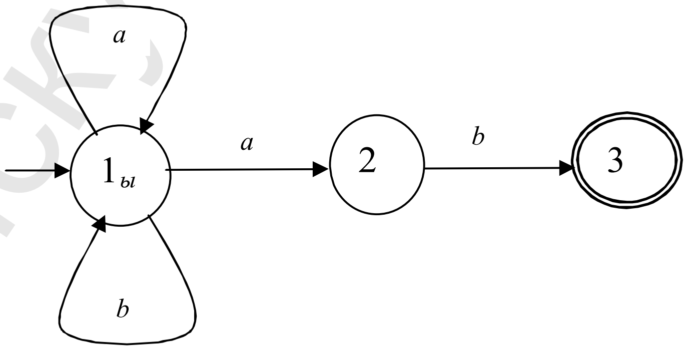
Индукция базисі. i=0 үшін $(q, \tau) \models_M^0 (p, \varepsilon)$ тек қана және тек қана сонда, қашан $\tau = \varepsilon$ және $p \in P'$ болатыны айқын.

Индукция қадамы. i, $\tau = t\zeta$, $|\zeta| = i$ үшін $(\boldsymbol{P}, \tau) \models_M^{i-1} (\boldsymbol{P}', \varepsilon)$ ақиқат деп алайық. Сонда кейбір $\boldsymbol{P} \in \boldsymbol{Q}$ үшін тұжырым $(\boldsymbol{P}, \tau) \models_M^i (\boldsymbol{P}', \varepsilon)$ және тұжырым $(q, t\zeta) \models_M^i (p,\varepsilon)$ тең мәнді.

Бұл жерден шығатыны, $(\{q_0\},\tau) \models_M^i (\boldsymbol{P}', \varepsilon)$, $\boldsymbol{P}' \in \boldsymbol{F}'$ үшін орындалса, кейбір $p \in \boldsymbol{F}$ үшін $(q_0,\tau) \models_M^i (p, \varepsilon)$ орындалады. Сонымен, $\boldsymbol{L}(\boldsymbol{M}') = \boldsymbol{L}(\boldsymbol{M})$.

Сонымен, ДАА-ның ауысу функциясы бір ғана күйді, ал БАА-ның ауысу функциясы бірден көп күйлерді анықтайды, яғни егер $q \in \boldsymbol{Q}$ – ағымдағы күй , $p \in \boldsymbol{Q}$ – келесі күй, $t \in \boldsymbol{T}$ – ағымдағы кіріс символы болса, онда $\delta(q, t)$ = р – ДАА үшін, $\delta(q, t)$ = {р} – БАА үшін және егер $\delta(q, t) = \varnothing$, онда $\delta(q, t)$ функциясы анықталмаған. Осыдан ДАА қайсы бір БАА-ның дербес жағдайы екендігі көрінеді.

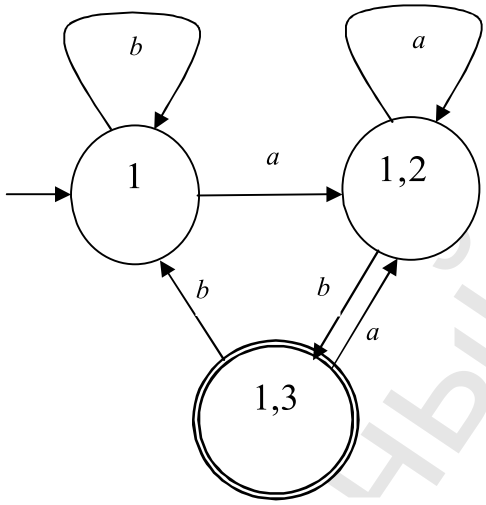
II.2.2.1-мысал. БАА $\boldsymbol{M_{13}} = \langle \boldsymbol{Q}, \boldsymbol{T}, \delta, q_0, \boldsymbol{F}, \vdash, \dashv, \delta \rangle$ мына параметрлерімен $\boldsymbol{Q}$={1, 2, 3}; $\boldsymbol{T}=\{a, b\}$; q_0=1; $\boldsymbol{F}$={3}; $\delta(1, a) = 1$; $\delta(1,b) = 1$; $\delta(1, a) = 2$; $\delta(2, b) = 3$ берілсін. $\boldsymbol{M_{13}}$ БАА-ның диаграммасы II.2.2.1-суретте көрсетілген.



II.2.2.1-сурет. $\boldsymbol{M_{13}}$ БАА-тың диаграммасы.

Бұл БАА M_{13} детерминдеу үшін II.2.2.1-теоремасын қолданамыз. Егер II.2.2.1-теоремасындағы дәлелдеу құрылымын пайдалансақ және бастапқы күйден қолжетім жоқ күйлерді жойсақ, онда келесі параметрлермен $\boldsymbol{Q} = \{\{1\},\{1,2\},\{1,3\}\}$; $\boldsymbol{T} = \{a,b\}$; $q_0 = \{1\}$, $\boldsymbol{F} = \{1,3\}$, $\delta(\{1\}, a) = \{1,2\}$; $\delta(\{1\}, b) = \{1\}$; $\delta(\{1,2\}, a) = \{1,2\}$; $\delta(\{1,2\}, b) = \{1,3\}$; $\delta(\{1,3\}, a)=\{1,2\}$; $\delta(\{1,3\}, b)=\{1\}$ толық ДАА M_{14} шығады.

M_{14} БАА-тың диаграммасы II.2.I.2.2-суретте көрсетілген.

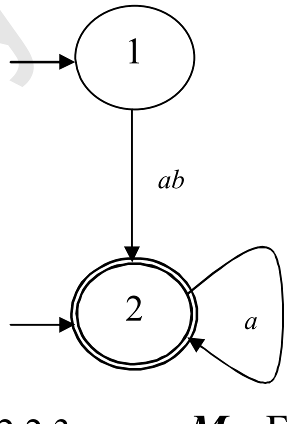


II.2.I.2.2-сурет. M_{14} ДАА-тың диаграммасы.

II.2.2-тапсырма.

1. Кіріс деректері БАА-ның күйлері мен ауысуларының кестесі, ал шығысы ДАА-ның күйлері мен ауысуларының кестесі болатын БАА-дан ДАА тұрғызатын алгоритмді жүзеге асырыңыз.

2. Диаграммасы төмендегі суретте көрсетілген M_{15}. БАА-қа эквивалентті толық ДАА табу.



II.2.2.3-сурет. M_{15} БАА-тың диаграммасы.

3. $\{a\xi b:\ \xi\in\{a,b\}^*\cup\{b\xi a:\ \xi\in\{a,b\}^*\}$ тілін танитын детерминді ақырлы автомат құрыңыз.

4. Келесілер ішінен бейдетерминді ақырлы автоматты көрсетіңіз:

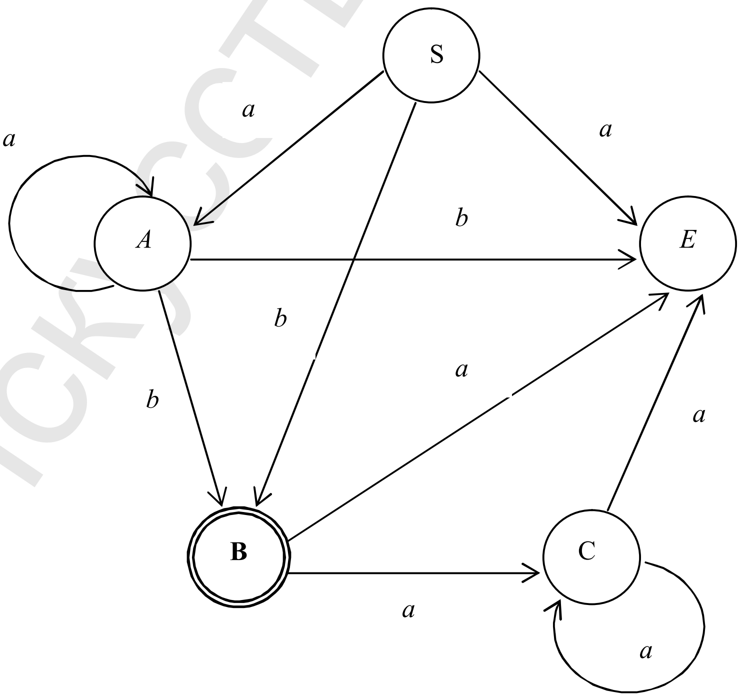
$$\boldsymbol{M}=<\boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{I}, \boldsymbol{F}, \vdash, \dashv, \Delta> \text{ және } \boldsymbol{G}=<N, \boldsymbol{T}, \boldsymbol{P}, S>$$

5. Келесілердің ішінен бейтерминді ақырлы автоматтың ауысу функциясын табыңыз:

$\delta(q,a)=p$;
$\delta(q,a)=\{p\}$;
$\delta(q,a)=\varnothing$;
$\delta(q,a)=q$.

II.2.2-сұрақтар:

1. Ақырлы автоматтардың ауысу функциясы деген не?
2. Ақырлы автоматтың кескіні неден тұрады?
3. Автоматпен танылатын тіл қалай анықталады?
4. Күйлері q_1, q_2 және кіріс сөздері α, β, δ болатын мына қатынастар $(q_1, \alpha\ \beta) \models^* (q_2, \beta\ )$ және $\neg((q_1, \alpha\ \delta) \models^* (q_2, \delta\ ))$ орындалатындай ақырлы автомат бар ма?
5. Келесі диаграмма қандай (бейдетерминді неме детерминді) ақырлы автоматқа қатысты болады?



II.3. Регулярлық тілдердің қасиеттері

II.3.1. Регулярлық өрнектер мен сызықты грамматикалардың эквиваленттілігі

Бұл параграфта регулярлық өрнектермен туындайтын регулярлық тілдер тобы оң сызықты грамматикалармен туындайтын оң сызықты тілдер тобына дәл сәйкес келетіндігі көрсетіледі, мысалдар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1-3,6,8,11-13,16-21,25,28-32].

Бұл тұжырым II.3.1.1-лемма және II.3.1.2-лемма көмегімен берілген регулярлық өрнекке (РӨ) сәйкес оң сызықты грамматиканы (ОСГ) тұрғызу, және керісінше, берілген ОСГ-ға сәйкес РӨ құру арқылы дәлелденеді.

II.3.1-лемма. $\boldsymbol{L(\rho)}$ регулярлық тілді тудырушы кез келген РӨ $\boldsymbol{\rho}$ үшін сәйкес $\boldsymbol{L(G)}$ тілін тудыратын және $\boldsymbol{L(G)} \subseteq \boldsymbol{L(\rho)}$ болатындай етіп ОСГ $\boldsymbol{G}$ тұрғызуға болады.

Дәлелдеу. $\boldsymbol{L(G)}$ тілін туғызатын ПЛГ $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ грамматикаы РӨ $\boldsymbol{\rho}$ ұзындығы, яғни $\boldsymbol{\rho}$ жазуындағы $\boldsymbol{T}$ әліпбиінің символы, $\boldsymbol{\phi}$, ε символдары, ∨ - селекция (таңдау), · - конкатенация (тіркесу), * - итерация амалдарының таңбалары мен жақшалардың жалпы саны, бойынша индукция арқылы тұрғызылады.

Индукция базисі. Айталық $\boldsymbol{\rho}$ РӨ-нің ұзындығы 1 болсын. Сонда $\boldsymbol{\rho}$ кейбір $t \in \boldsymbol{T}$ үшін мына $\emptyset$, $\{\varepsilon\}$ және $\{t\}$ регулярлық тілдерді тудыратын үш (1) $\boldsymbol{\phi}$, (2) ε және (3) t РӨ-нің біреуі болады.

РВ (1) ϕ, (2) ε және (3) t үшін ОСГ мынадай болады:

(1) Егер $\boldsymbol{N} = \{S\}$, $\boldsymbol{P} = \emptyset$, онда $\boldsymbol{L(G)} = \emptyset$;

(2) Егер $\boldsymbol{N} = \{S\}$, $\boldsymbol{P} = \{S \to \varepsilon\}$, онда $\boldsymbol{L(G)} \rightleftharpoons \{\varepsilon\}$;

(3) Егер $\boldsymbol{N} = \{S\}$, $\boldsymbol{P} = \{S \to t\}$, онда $\boldsymbol{L(G)} \rightleftharpoons \{t\}$.

Индукциялық қадам. Айталық $\boldsymbol{L(\rho')}$ және $\boldsymbol{L(\rho'')}$, $\boldsymbol{L(\rho')} \cap \boldsymbol{L(\rho'')} = \emptyset$ тілдерін тудыратын, ұзындығы $k \geq 1$ РӨ $\boldsymbol{\rho'}$ және $\boldsymbol{\rho''}$ үшін $\boldsymbol{L(G')} \subseteq \boldsymbol{L(\rho')}$ және $\boldsymbol{L(G'')} \subseteq \boldsymbol{L(\rho'')}$ болатындай сәйкес ОСГ ПЛГ $\boldsymbol{G'} = <\boldsymbol{N'}, \boldsymbol{T}, \boldsymbol{P'}, S'>$ және $\boldsymbol{G''} = <\boldsymbol{N''}, \boldsymbol{T}, \boldsymbol{P''}, S''>$ тұрғызылсын.

Ұзындығы k+1 болатын кез келген РӨ $\boldsymbol{\rho}$ қарастырайық. Бұл РӨ келесі үш түрдің біреуі болады:

(1) $\boldsymbol{L(\rho) = L(\rho')\cup L(\rho'')}$ тілін тудыратын $\boldsymbol{\rho = \rho'\vee\rho''}$;

(2) $\boldsymbol{L(\rho) = L(\rho')\cdot L(\rho'')}$ тілін тудыратын $\boldsymbol{\rho = \rho'\cdot\rho''}$;

(3) $\boldsymbol{L(\rho) = L(\rho')^*}$ тілін тудыратын $\boldsymbol{\rho = \rho'^*}$.

Сонда $\boldsymbol{L(G)\subseteq L(\rho)}$ тілін тудыратын ОСГ $\boldsymbol{G}=<\boldsymbol{N},\boldsymbol{T},\boldsymbol{P},S>$ келесігідей тұрғызылады:

(**I**) $\boldsymbol{\rho = \rho'\vee\rho''}$ РӨ үшін:

$\boldsymbol{N} = \boldsymbol{N'}\cup\mathbf{N''}\cup S$, мұнда S – жаңа алғашқы бейтерминал ;

$\boldsymbol{P} = \boldsymbol{P'}\cup\boldsymbol{P''}\cup\{S\to S'|S''\}$.

$\boldsymbol{L(G)}$ тілі $\boldsymbol{L(G')}$ және $\boldsymbol{L(G'')}$ тілдерінің барлық тізбелерін қамтиды, яғни $\boldsymbol{L(G)=L(G')\cup L(G'')\subseteq L(\rho')\cup L(\rho'')= L(\rho)}$.

(**II**) РӨ $\boldsymbol{\rho = \rho'\cdot\rho''}$ үшін:

$\boldsymbol{N} = \boldsymbol{N'}\cup\mathbf{N''}$;

$S = S'$;

$\boldsymbol{P = P'\cup P''}$.

$\boldsymbol{L(G)}$ тілі $\boldsymbol{L(G')}$ және $\boldsymbol{L(G'')}$ тілдерінің барлық тізбелерін қамтиды, яғни $\boldsymbol{L(G)=L(G')\cdot L(G'')\subseteq L(\rho')\cdot L(\rho'')= L(\rho)}$.

(**III**) РӨ $\boldsymbol{\rho = \rho'^*}$ үшін:

$\boldsymbol{N} = \boldsymbol{N'}\cup\mathbf{N''}$;

$S = S'$;

$\boldsymbol{P} = \boldsymbol{P'}\cup\{S\to\varepsilon|S\}$.

$\boldsymbol{L(G)}$ тілі $\boldsymbol{L(G')}$ тілінің барлық тізбелерін және ε бос тізбесін қамтиды, яғни $\boldsymbol{L(G)=L(G')^*\subseteq L(\rho')^*= L(\rho)}$.

$\boldsymbol{T}$ ақырлы әліпбиінде барлық t үшін $\{t\}$, $\varnothing$ және $\{\varepsilon\}$ жиындарын қамтитын регулярлық тілдер тобы бірігу, конкатенация және итерация амалдары бойынша тұйық болады болатындығы белгілі. Индукциялық базис бойынша регулярлық жиындар $\varnothing$, $\{\varepsilon\}$ және $\{t\}$ оң сызықты болады, демек, оң сызықты тілдер тобы бірігу, конкатенация және итерация амалдары бойынша тұйық болады.

II.3.2-лемма. $\boldsymbol{L(G)}$ оң сызықты тілді тудырушы кез келген ОСГ $\boldsymbol{G}$ үшін сәйкес $\boldsymbol{L(\rho)}$ тілін тудыратын және $\boldsymbol{L(\rho)\subseteq L(G)}$ болатындай етіп РӨ $\boldsymbol{\rho}$ тұрғызуға болады.

Дәлелдеу. Дәлелдеу оқушыларға жаттығу ретінде қалдырылады.

II.3.1.1-леммасынан және II.3.1.2-леммасынан келесі теореманың дәлелдеуі шығады.

II.3.1-теорема. Оң сызықты грамматикалармен туындайтын тілдер сыныбы регулярлық тілдер сыныбына сәйкес келеді.

II.3.1.1-мысал. Мейлі $\boldsymbol{G}$ грамматикасы мына $S\rightarrow 0A|1S|\varepsilon$, $A\rightarrow 0B|1A$, $S\rightarrow 0S|1B$ ережелермен анықталсын. Сонда сәйкес теңдеулер жүйесі X_1, X_2, X_3 айнымалыларын сәйкес S, A, B айнымалыларына аустырғаннан кейін шығады. $\boldsymbol{L}(\boldsymbol{G})$ – құрамындағы нөлдер саны үшке бөлінетін тізбелер жиыны. Бұл жиынның регулярлық өрнекпен белгіленетінін көру қиын емес.

II.3.1-тапсырмалар:

1. Мына регулярлық өрнектерге эквивалентті оң сызықты грамматикаларды тұрғызыңыздар:
   1) $(a\vee b\vee ab)^*$.
   2) $(a^*b)^*\vee(b^*a)^*$.
   3) $(ba\vee a^*ab)^*$.
   4) $(b^+a)^*b\vee a)b^*$.
   5) $(b^*a)b\vee(ab^*)a$.
   6) $a^*bc \vee b^*a$.
   7) $(01^*\vee 1)^*11$

II.3.1-сұрақтар:

1. Бір тілді тудыратын екі регулярлық өрнектің екеуіне де эквивалентті оң сызықты грамматика табылама?
2. $(0\vee(1(0^*)))$ және $0\vee 10^*$ регулярлық өрнектерге эквивалентті грамматикалар біп-біріне тең бола ма?
3. Оң сызықты $S\rightarrow E$; $S\rightarrow bE$; $S\rightarrow caE$; $E\rightarrow a$; $E\rightarrow bS$ грамматикасына эквивалентті регулярлық өрнек тұрғызыла ма?
4. $(a\vee b)^*(b\vee a)^*$ және $(a\vee b\vee ab)^*$ регулярлық өрнектерге эквивалентті оң сызықты грамматикалар бір-бірімен тең бола ма?

II.3.2. Регулярлық өрнектер және ақырлы автоматтардың эквиваленттілігі

Бұл параграфта регулярлық өрнектермен туындайтын регулярлық тілдер тобы бейдетерминді ақырлы автоматтармен танылатын тілдер тобына дәл сәйкес келетіндігі көрсетіледі, мысалдар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1-3,5-13,15-21,23,25-34].

Бұл берілген регулярлық өрнекке (РӨ) сәйкес бейдетерминді ақырлы автоматтарды (БАА) тұрғызу, және керісінше, берілген БАА-қа сәйкес РӨ құру арқылы дәлелденеді.

II.3.2.1-лемма. Регулярлық тіл $\boldsymbol{L(\rho)}$-ны тудырушы кез келген РӨ $\boldsymbol{\rho}$ үшін сәйкес $\boldsymbol{L(M)}$ тілін танитын және $\boldsymbol{L(M)}\subseteq\boldsymbol{L(\rho)}$ болатындай етіп БАА $\boldsymbol{M}$ тұрғызуға болады.

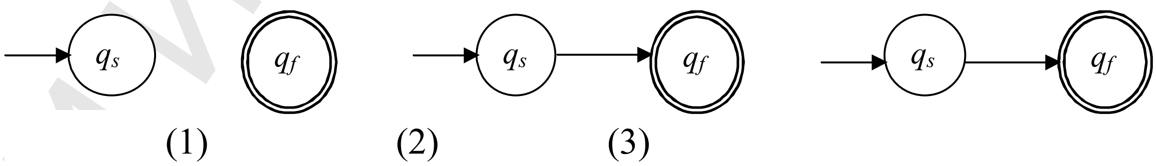
Дәлелдеу. $\boldsymbol{L(M)}$ тілін танитын БАА $\boldsymbol{M}$ РӨ $\boldsymbol{\rho}$ ұзындығы, яғни $\boldsymbol{\rho}$ жазуындағы $\boldsymbol{T}$ әліпбиінің символы, ϕ , ε символдары, ∨ - селекция (таңдау), · - конкатенация (тіркесу), * - итерация (қайталау) амалдарының таңбалары мен жақшалардың жалпы саны, бойынша индукция арқылы тұрғызылады.

Индукция базисі. Айталық $\boldsymbol{\rho}$ РӨ-нің ұзындығы 1 болсын. Сонда $\boldsymbol{\rho}$ кейбір $t\in\boldsymbol{T}$ үшін мына $\emptyset$, $\{\varepsilon\}$ және $\{t\}$ регулярлық тілдерді тудыратын үш (1) ϕ, (2) ε және (3) t РӨ-нің біреуі болады.

(1) ϕ, (2) ε және (3) t РӨ-не сәйкес БАА

$$\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, q_s, \boldsymbol{F}, \vdash, \dashv, \delta>$$

мыналарды қамтиды: $\boldsymbol{Q} = \{q_s, q_f\}$, q_s – алғашқы күй, q_f – соңғы күй $\boldsymbol{T} = \{t\}$, $\boldsymbol{F} = \{q_f\}$. Ауысу функциясы $\delta(q_s, t) = q_f$, ал қалған жағдайда δ анықталмаған, яғни $\boldsymbol{L(M)}\rightleftharpoons\{t\}$. II.3.2.1-суретте $\emptyset$, $\{\varepsilon\}$ және $\{t\}$ регулярлық тілдерді танитын БАА диаграммалары көрсетілген.



(1) (2) (3)

II.3.2.1-сурет. $\emptyset$, $\{\varepsilon\}$ және $\{t\}$ тілдерін танитын БАА.

Индукциялық қадам. Айталық ұзындығы $k \geq 1$ болатын $\boldsymbol{L}(\boldsymbol{\rho}')$ және $\boldsymbol{L}(\boldsymbol{\rho}'')$, $\boldsymbol{L}(\boldsymbol{\rho}')\cap\boldsymbol{L}(\boldsymbol{\rho}'')=\emptyset$ тілдерін танитын әрбір $\boldsymbol{\rho}'$ және $\boldsymbol{\rho}''$ РӨ үшін сәйкес БАА $\boldsymbol{M}'$ және $\boldsymbol{M}''$ тұрғызылды делік. Ұзындығы k+1 болатын кез келген РӨ $\boldsymbol{\rho}$ қарастыралық. Бұл РӨ келесі үш түрдің біреуіне ие болады:

(1) $\boldsymbol{\rho} = \boldsymbol{\rho}'\vee\boldsymbol{\rho}''$ РӨ $\boldsymbol{L}(\boldsymbol{\rho}) = \boldsymbol{L}(\boldsymbol{\rho}')\cup \boldsymbol{L}(\boldsymbol{\rho}'')$ тілін тудырады;

(2) $\boldsymbol{\rho} = \boldsymbol{\rho}'\cdot\boldsymbol{\rho}''$ РӨ $\boldsymbol{L}(\boldsymbol{\rho}) = \boldsymbol{L}(\boldsymbol{\rho}')\cdot\boldsymbol{L}(\boldsymbol{\rho}'')$ тілін тудырады;

(3) $\boldsymbol{\rho} = \boldsymbol{\rho}'^*$ РӨ $\boldsymbol{L}(\boldsymbol{\rho}) = \boldsymbol{L}(\boldsymbol{\rho}')^*$ тілін тудырады.

Мейлі $\boldsymbol{L}(\boldsymbol{M}') = \boldsymbol{L}(\boldsymbol{\rho}')$ және $\boldsymbol{L}(\boldsymbol{M}'') = \boldsymbol{L}(\boldsymbol{\rho}'')$ тілдерін танитын БАА $\boldsymbol{M}' = <\boldsymbol{Q}', \boldsymbol{T}, q'_s, \boldsymbol{F}', \vdash,\dashv, \delta'>$ және $\boldsymbol{M}'' = <\boldsymbol{Q}'', \boldsymbol{T}, q''_s, \boldsymbol{F}'', \vdash,\dashv, \delta''>$ болсын.

Жалпылықты шектемей, біз олардың күйлері бөлек деп алайық, яғни $\boldsymbol{Q}'\cap\boldsymbol{Q}'' = \emptyset$. Олай болмаған жағдайда күйлерді қайта атауға болады. Сонда $\boldsymbol{L}(\boldsymbol{M})\subseteq\boldsymbol{L}(\boldsymbol{\rho})$ тілін танитын $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, q_s, \{q_f\}, \vdash,\dashv, \delta>$ БАА, келесідегідей тұрғызылады:

(**I**) РӨ $\boldsymbol{\rho} = \boldsymbol{\rho}'\vee\boldsymbol{\rho}''$ үшін:

$\boldsymbol{Q} = \boldsymbol{Q}'\cup\boldsymbol{Q}''\cup\{q_s, q_f\}$, мұнда q_s – жаңа алғашқы күй, q_f – жаңа соңғы күй;

$$\boldsymbol{F} = \begin{cases} \boldsymbol{F}'\cup\boldsymbol{F}'', \text{ егер } \varepsilon\notin\boldsymbol{L}(\boldsymbol{\rho}')\cup \boldsymbol{L}(\boldsymbol{\rho}''); \\ \boldsymbol{F}= \boldsymbol{F}'\cup\boldsymbol{F}''\cup\{q_f\}, \text{ егер } \varepsilon\in\boldsymbol{L}(\boldsymbol{\rho}')\cup \boldsymbol{L}(\boldsymbol{\rho}''); \end{cases}$$

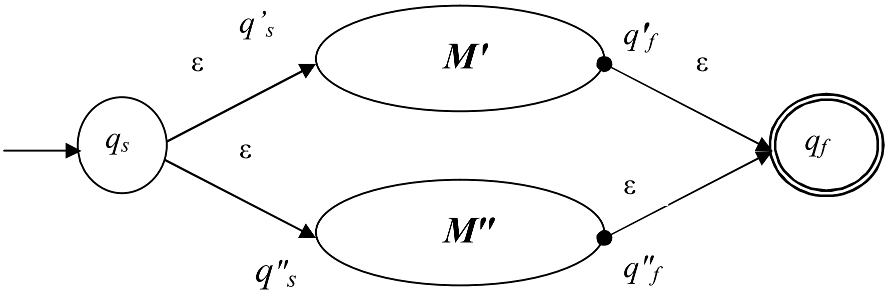
$\delta(q_s, t) = \delta(q'_s, t)\cup\delta(q''_s, t)$ барлық $t\in\boldsymbol{T}$ үшін;

$$\delta(q, t) = \begin{cases} \delta'(q, t) \text{ барлық } q\in\boldsymbol{Q}' \text{ және } t\in\boldsymbol{T} \text{ үшін,} \\ \delta''(q, t) \text{ барлық } q\in\boldsymbol{Q}'' \text{ және } t\in\boldsymbol{T} \text{ үшін.} \end{cases}$$

Сонымен қатар, δ келемі төрт ε-ауысуды қамтиды:

$$\delta(q_s, \varepsilon)=q'_s, \delta(q_s, \varepsilon)=q''_s, \delta(q'_f, \varepsilon)=q_f, \delta(q''_f, \varepsilon)=q_f.$$

$\boldsymbol{L}(\boldsymbol{M})$ тілі $\boldsymbol{L}(\boldsymbol{M}')$ және $\boldsymbol{L}(\boldsymbol{M}'')$ тілдеріндегі барлық тізбені қамтиды. Алдымен БАА $\boldsymbol{M}$ екі $\boldsymbol{M}'$ және $\boldsymbol{M}''$ БАА-ның қайсысын моделдейтінін табады. Кейін өзінің бейдетерминділігіне байланысты $\boldsymbol{M}'$ және $\boldsymbol{M}''$ БАА-ның екеуінде моделдейді. $k\geq 1$ бойынша индукция арқылы $(q_s,\tau)\models^k_{M}(q, \varepsilon)$ тек қана $q\in\boldsymbol{Q}'$ және $(q'_s,\tau)\models^k_{M'}(q, \varepsilon)$ немесе $q\in\boldsymbol{Q}''$ және $(q''_s, \tau)\models^k_{M''}(q, \varepsilon)$ болғанда. Осыдан $\boldsymbol{L}(\boldsymbol{M})=\boldsymbol{L}(\boldsymbol{M}')\cup\boldsymbol{L}(\boldsymbol{M}'')$. Бұл БАА $\boldsymbol{M}$ диаграммасы II.3.2.2-суретте көрсетілген.



II.3.2.2-сурет. $\mathscr{L}(\boldsymbol{M}) = \mathscr{L}(\boldsymbol{M'}) \cup \mathscr{L}(\boldsymbol{M''})$ тілін танитын БАА

Бұл жерде әрбір $\tau\in\boldsymbol{L}(\boldsymbol{M})$ q_s-тен в q_f-ке көшіреді және бірінші қадамнан кейін оны тасушы жол q'_s немесе q''_s арқылы өтеді. $\boldsymbol{M'}$ және $\boldsymbol{M''}$ күйлері қиылыспағандықтан, бірінші жағдайда бұл жол q'_f күйінен q_f күйіне тек ε-ауысу арқылы ғана жете алады және онда $\tau\in\boldsymbol{L}(\boldsymbol{M'})$. Сондай-ақ, екінші жағдайда q''_f-тен q_f-ке тек ε-ауысу арқылы ғана жете алады және $\tau\in\boldsymbol{L}(\boldsymbol{M''})$.

(**II**) РӨ $\boldsymbol{\rho} = \boldsymbol{\rho}'\cdot\boldsymbol{\rho}''$ үшін:

$\boldsymbol{Q} = \boldsymbol{Q'}\cup\boldsymbol{Q''}$;

$q_s = q'_s\in\boldsymbol{Q'}$ – алғашқы күй;

$$\boldsymbol{F} = \begin{cases} \boldsymbol{F''}, \text{ егер } q''\notin\boldsymbol{F''}; \\ \boldsymbol{F'}\cup\boldsymbol{F''}, \text{ егер } q''\in\boldsymbol{F''}; \end{cases}$$

ауысу функциясы δ мына теңдіктермен анықталады:

1) $\delta(q, t)= \delta'(q, t)$ барлық $q\in\boldsymbol{Q'} \setminus \boldsymbol{F'}$ және $t\in\boldsymbol{T}$ үшін;

2) $\delta(q, t)= \delta'(q, t)\cup\{\delta(q, \varepsilon)=q''_s\}\cup\delta''(q''_s, t)$ барлық $q\in\boldsymbol{F'}$ және $t\in\boldsymbol{T}$ үшін;

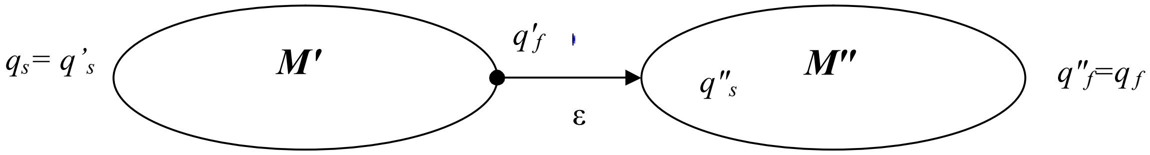
3) $\delta(q, t)= \delta''(q, t)$ барлық $q\in\boldsymbol{Q''}$ және $t\in\boldsymbol{T}$ үшін.

Сонымен, БАА $\boldsymbol{M}$ жұмысын $\boldsymbol{M'}$ БАА-ын моделдеуден бастайды. Қашан $\boldsymbol{M}$ БАА-ы $\boldsymbol{M'}$ БАА-ның соңғы күйіне жеткенде, ол ε-ауысуы $\delta(q'_f, \varepsilon) = q''_s$ арқылы бейдетерминді $\boldsymbol{M''}$ БАА-ның алғашқы күйіне өтеді және $\boldsymbol{M''}$ БАА-ын моделдейді.

Мейлі екі тізбе $\chi\in\boldsymbol{L}(\boldsymbol{M'})$ және $\xi\in\boldsymbol{L}(\boldsymbol{M''})$ берілсін. Сонда кейбір $q\in\boldsymbol{F'}$ үшін $(q'_s, \chi\xi) \models^{+}_{\text{м}} (q, \xi)$. Егер $\chi=\varepsilon$, онда $q=q'$. Егер $\xi\neq\varepsilon$, онда, (2)-ші теңдікті бір рет немесе (3) -ші теңдікті нөл немесе бірнеше рет кейбір $p\in\boldsymbol{F''}$ үшін мынаны $(q, \xi) \models^{+}_{\text{м}}(p, \varepsilon)$ аламыз. Егер $\xi=\varepsilon$, онда $q\in\boldsymbol{F}$, себебі $p\in\boldsymbol{F''}$. Осыдан $\chi\xi\in\boldsymbol{L}(\boldsymbol{M})$. Айталық, $\tau\in\boldsymbol{L}(\boldsymbol{M})$ болсын. Сонда $(q',\tau) \models^{*}_{\text{м}} (q_f, \varepsilon)$. Екі жағдайды жеке қарастырайық: $q_f\in\boldsymbol{F''}$ және

$q_f \in \boldsymbol{F'}$. Мейлі $q_f \in \boldsymbol{F''}$. Сонда $(q'_s,\ \chi t\xi) \models^* (q'_f,\ t\xi) \models (q''_s,\ \xi) \models^* (q''_f,\ \varepsilon)$ шартын қанағаттандыратын кейбір $t \in \boldsymbol{T}$ үшін $\tau = \chi t\xi$.

Демек, $\chi \in \boldsymbol{L}(\boldsymbol{M'})$ және $t\xi \in \boldsymbol{L}(\boldsymbol{M''})$. Мейлі $q_f \in \boldsymbol{F'}$, сонда $\varepsilon \in \boldsymbol{L}(\boldsymbol{M''})$. Сонымен, $\tau \in \boldsymbol{L}(\boldsymbol{M'})$. Осыдан $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}(\boldsymbol{M'}) \cdot \boldsymbol{L}(\boldsymbol{M''})$. Бұл $\boldsymbol{M}$ БАА-ның диаграммасы II.3.2.3-суретте көрсетілген.



II.3.2.3-сурет. $\mathscr{L}(\boldsymbol{M}) = \mathscr{L}(\boldsymbol{M'}) \bullet \mathscr{L}(\boldsymbol{M''})$ тілін танитын БАА $\boldsymbol{M}$.

Бұл жерде $q_s = q'_s$-тен $q_f = q''_f$-ке баратын барлық жол q'_f күйінен q''_s күйіне ε-ауысу арқылы өтеді. Сондықтан $\boldsymbol{M}$ танитын әрбір тізбе $\boldsymbol{L}(\boldsymbol{M'})$ тілінің қайсы бір тізбесі мен $\boldsymbol{L}(\boldsymbol{M''})$ тілінің қайсы бір тізбесінің тіркесі болады, және осындай кез келген тіркес жіберіледі. Демек, БАА $\boldsymbol{M}$ бізге керек $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}(\boldsymbol{\rho'} \cdot \boldsymbol{\rho''}) = \boldsymbol{L}(\boldsymbol{M'}) \cdot \boldsymbol{L}(\boldsymbol{M''})$ тілін таниды.

(**III**) РӨ $\boldsymbol{\rho} = \boldsymbol{\rho'}^*$:

$\boldsymbol{Q} = \boldsymbol{Q'} \cup \{q_s, q_f\}$, q_s – ағашқы жағдай, q_f – соңғы жағдай;

$\boldsymbol{F} = \boldsymbol{F}' \cup \{q_f\}$;

ауысу функциясы δ мына теңдіктермен анықталады:

(1)$\delta(q, t) = \delta'(q, t)$, егер $q \in \boldsymbol{Q} \backslash \boldsymbol{F'}$ және $t \in \boldsymbol{T}$,

(2)$\delta(q, t) = \delta'(q, t) \cup \delta'(q'_s, t)$, егер $q \in \boldsymbol{F'}$ және $t \in \boldsymbol{T}$,

(3)$\delta(q_s, t) = \delta'(q'_s, t)$ барлық $\in \boldsymbol{T}$ үшін.

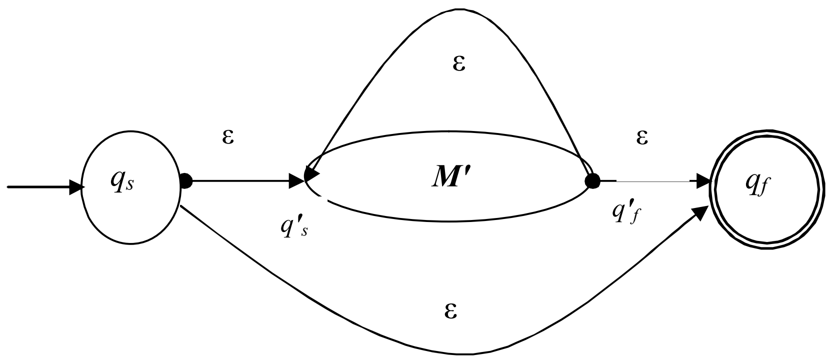
Сонымен, БАА $\boldsymbol{M}$ қашан $\boldsymbol{M'}$ БАА-ның соңғы күйіне жеткенде, ол бейдетерминді таңдап, иә $\boldsymbol{M'}$ БАА-ын моделдеуді жалғастырады, иә $\boldsymbol{M'}$ БАА-ын жаңадан алғашқы күйден бастап моделдейді.

$\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}(\boldsymbol{\rho})^*$ екендігін дәлелдеу (**II**)-ші бөлімнің дәлелдеуіне ұқсас. $\varepsilon \in \boldsymbol{L}(\boldsymbol{M})$ екендігі ескерілуі қажет, себебі $q' \in \boldsymbol{F'}$ – соңғы күй.

БАА $\boldsymbol{M}$-ның ауысу функциясы, $\boldsymbol{M'}$-ның аысу функциясынан басқа, төрт жаңа ε-ауысуын қамтиды:

$$\delta(q_s, \varepsilon) = q'_s;\ \delta(q_s, \varepsilon) = q_f;\ \delta(q'_f, \varepsilon) = q'_s;\ \delta(q'_f, \varepsilon) = q_f.$$

Бұл автоматтың диаграммасы II.3.2.4-суретте көрсетілген.



II.3.2.4-сурет. $\mathscr{L}(\boldsymbol{M'})^*$ тілін танитын БАА.

Бұл жерде бос емес тізбе $\tau \neq \varepsilon$ үшін итерация анықтамасы бойынша кейбір $k \geq 1$ үшін τ тізбесін k іштізбеге бөлуге болады: $\tau = \tau_1\tau_2\ldots\ \tau_k$ және барлық $\tau_i \in \boldsymbol{L}(\boldsymbol{M'})$. Әрбір $i = 1,\ldots\ , k$ үшін τ_i тізбесі q'_s күйін q'_f күйіне ауыстырады. Сонда τ тізбесі үшін БАА $\boldsymbol{M}$ диаграммасында II.3.2.5-суретте көрсетілгендей жол бар.

$$q_s \xrightarrow{\varepsilon} q'_s \quad \xrightarrow{\tau_1} q'_f \quad \xrightarrow{\varepsilon} q'_s \quad \xrightarrow{\tau_2} q'_f \;\cdots\; q'_s \xrightarrow{\tau_k} q'_f \quad \xrightarrow{\varepsilon} q_f$$

II.3.2.5-сурет. БАА $\boldsymbol{M}$ диаграммасындағы жол.

Демек, $\tau \in \boldsymbol{L}(\boldsymbol{M})$. Керісінше, егер кейбір тізбе q_s-тен q_f -ке ауыстырса, онда ол иә бос тізбе ε, иә оны q_s-тен q'_s-ке өтетін, сонан кейін бірнеше рет q'_s-тен q'_f -ке өтетін және ε-ауысу арқылы q'_f -тен в q'_s-ке қайтатын, ақырында ε-ауысу арқылы q'_f -тен q'_f –те бітетін жол алып жүреді. Сондықтан мұндай тізбе $\tau \in \boldsymbol{L}(\boldsymbol{M'})^*$.

Сонымен, ақырлы-автоматты тілдер тобы барлық $t \in \boldsymbol{T}$ символдары үшін $\varnothing$, $\{\varepsilon\}$, $\{t\}$ тілдерін қамтиды және бірігу, тіркесу және итерация амалдарына қатысты тұйық болады.

1-ші лемманың дәлелдеуінде РӨ $\boldsymbol{\rho}$ бойынша тұрғызылатын автомат $\boldsymbol{M}$ барлық кезде қарапайым емес. Мысалы:

1) $\boldsymbol{M'}$ мен $\boldsymbol{M''}$ бірігуі үшін тұрғызылатын автоматта олардың алғашқы күйлерін бір күйге біріктіруге болады, егер оларда басқа күйлерден ауысу болмаса; соңғы күйлерін бір күйге біріктіруге болады, егер олардан басқа күйлерге ауысу болмаса және $\boldsymbol{M'}$ мен $\boldsymbol{M''}$

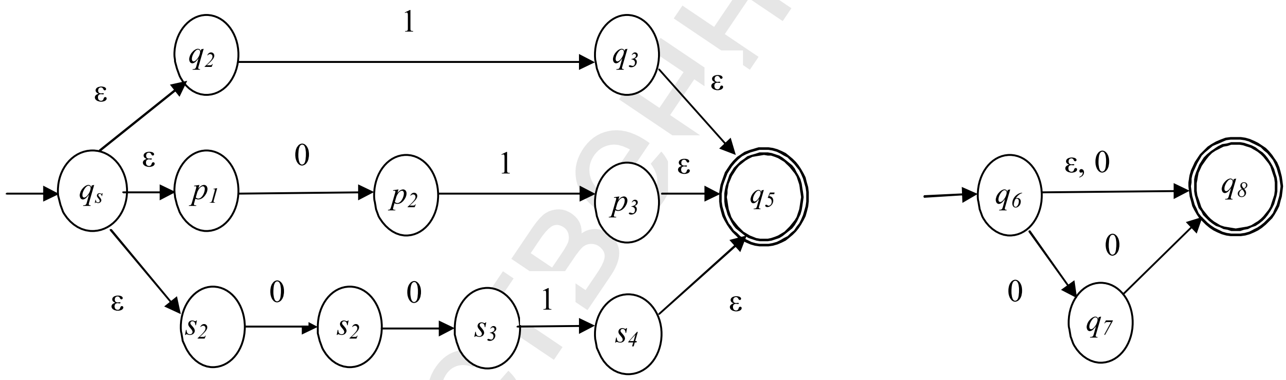
әліпбилері бірдей болса. Сонда жаңа алғашқы күй, жаңа соңғы күй және ε-ауысулар қажет болады.

2) ***M′*** мен ***M″*** тіркесуі үшін тұрғызылатын автомата, егер ***M′***-нің соңғы күйінен басқа күйлерге ауысу болмаса, онда оны ***M″***-нің алғашқы күйімен біріктіруге болады.

3) t_1t_2 … t_n, $t_i \in \boldsymbol{T}$ (i=1, 2, …, n) тізбесін жүзеге асыру үшін (n+1) күйлері $q_j \in \boldsymbol{Q}$ (j=0,1,2, … , n) және $\delta(q_{i-1}, t_i) = q_i$ сияқты пәрмендері бар, өзінде тіркесу үшін жалпы құрылымға қатысатын бос ε-ауысу жоқ автоматты пайдалануға болады.

II.3.2.1-мысал. '000' іштізбені қамтымайтын барлық тізбеден тұратын тілді тудыратын $\boldsymbol{\rho}$ = (1∨01∨001)*(ε∨0∨00) РӨ-не 1-ші лемманы қолданайық.

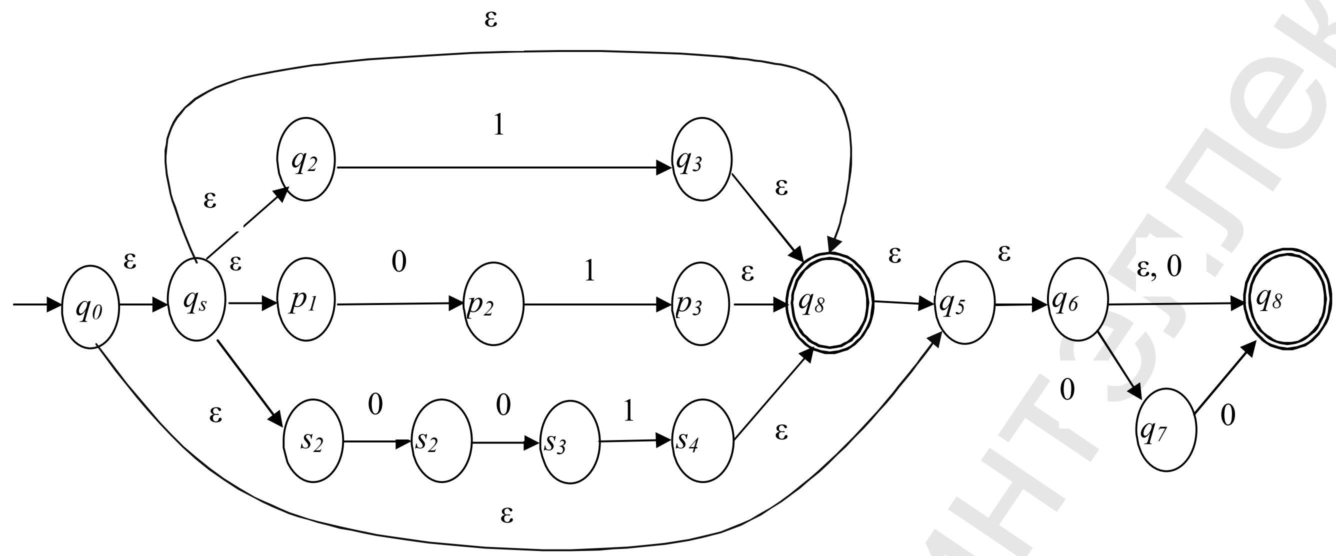
Төмендегі II.3.2.6-суретте $\boldsymbol{\rho_1}$ = (1∨01∨001) және $\boldsymbol{\rho}_2$= (ε∨0∨00) өрнектеріне сәйкес тіркесу мен бірігу құрылымдары арқылы тұрғызылған $\boldsymbol{M_1}$ және $\boldsymbol{M}_2$ автоматтарының диаграммалары берілген.



II.3.2.6 - сурет.a) ***M₁ AA***　　　　II.3.2.6 - сурет.b) ***M₂ AA***

Жоғарыда айтылғандай, $\boldsymbol{M_1}$ автоматын алғашқы q_2, p_1 және s_1 күйлерін, сонымен қатар соңғы q_3, p_3 және s_4 күйлерін біріктіріп, тағыда да қарапайымдауға болады.

$\boldsymbol{\rho}'$* = (1∨01∨001)* өрнегі үшін $\boldsymbol{M}_3$ автоматы $\boldsymbol{M}_1$ автоматынан жаңа алғашқы күй q_0 және соңғы күй q_5, сонымен қатар q_0-ден q_1-ге және q_5-ке, q_4-тен q_5-ке және q_1-ге ε-ауысуларды қосу арқылы алынады. Сонан кейін алғашқы РӨ $\boldsymbol{\rho}$ үшін нәтижелік автомат $\boldsymbol{M}_3$ және $\boldsymbol{M_2}$ автоматтарын тізбелеп біріктіруден шығады, ол II.3.2.7-суретте көрсетілген.



II.3.2.7-сурет. РӨ $\boldsymbol{\rho}$ үшін нәтижелік автомат

II.3.2.1-салдар. Әрбір РӨ үшін осы РӨ тудыратын тілді танитын детерминді ақырлы автоматты (ДАА) тұрғызуға болады.

II.3.2.2-лемма. $\boldsymbol{L}(\boldsymbol{M})$ тілін танитын кез келген БАА $\boldsymbol{M}$ үшін сәйкес $\boldsymbol{L}(\boldsymbol{\rho})$ регулярлық тілді тудыратын және $\boldsymbol{L}(\boldsymbol{\rho}) \subseteq \boldsymbol{L}(\boldsymbol{M})$ болатындай етіп РӨ $\boldsymbol{\rho}$ құруға болады.

Дәлелдеу. Мейлі БАА $\boldsymbol{M} = < \boldsymbol{Q}, \boldsymbol{T}, q_1, \boldsymbol{F}, \vdash, \dashv, \delta >$ тудыратын $\boldsymbol{L}(\boldsymbol{M})$ тілі берілсін. БАА $\boldsymbol{M}$-ға кеңейтілген ауысу функциясын ендірейік: $\delta^{\varepsilon}(q, \tau) = p$, мұнда $\tau \in \boldsymbol{T}^*$, $q, p \in \boldsymbol{Q}$ тек қана сол жағдайда, қашан $(q, \tau) \models^* (\mathrm{p}, \varepsilon)$.

ρ_{ij}^{k} арқылы $\delta^{\varepsilon}(q_i, x) = q_j$ және егер x пен ε-нен бөлек кез келген x тізбесінің префиксі y үшін $\delta^{\varepsilon}(q_i, y) = q_m$, онда $m \leqslant k$ болатындай барлық x тізбесінің жиынын белгілейік. Басқаша айтқанда, ρ_{ij}^{k} БАА-ны $m > k$ үшін ешқандай q_m күйінен өтпей q_i күйінен q_j күйіне көшіретін барлық тізбелердің жиыны. Бірақ, $i \geq k$ және $j \geq k$ болуы мүмкін.

ρ_{ij}^{k} келесідегідей рекурсивті анықталуы мүмкін:

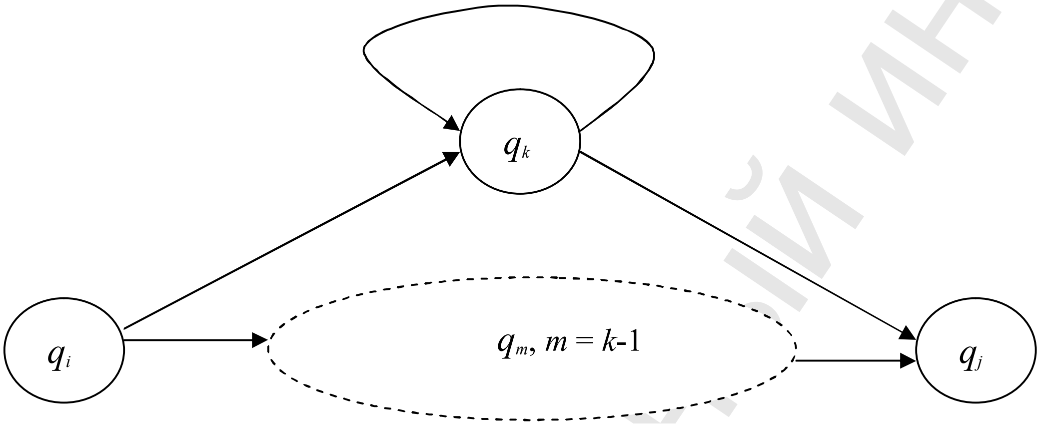
$$\rho_{ij}^{m} = \{t\colon t \in \boldsymbol{T} \ \&\ \delta(q_i, t) = q_j\},$$
$$\rho_{ij}^{k} = \rho_{ij}^{k-1} \cup \rho_{ik}^{k-1}(\rho_{kk}^{k-1})^{*}\rho_{kj}^{k-1}, \text{ мұнда } 1 \leqslant k \leqslant n.$$

Сонымен, ρ_{ij}^{k} анықтамасы, БАА $\boldsymbol{M}$ нөмірі k-дан үлкен күйден өтпей, q_i күйінен q_j күйіне көшіретін кіріс тізбесі τ үшін келесі екі тұжырымның дәл біреуі орындалатындығын білдіреді:

1) τ тізбесі $\rho_{ij}^{k\text{-}1}$ жиынына жатады, яғни τ тізбесін талдағанда БАА ешқашан нөмірлері k-ға тең немесе үлкен күйден өтпейді;

2) τ тізбесі мына түрде $\tau = \tau_1\tau_2\tau_3$ болуы мүмкін, мұнда іштізбе $\tau_1 \in \rho_{ik}^{k\text{-}1}$ ***M*** БАА-тын q_k күйіне көшіреді; іштізбе $\tau_2 \in (\rho_{kk}^{k\text{-}1})^*$ ***M*** БАА-тын нөмірі k-дан үлкен немесе тең күйден өтпей, q_k күйінен қайтадан q_k күйіне көшіреді; іштізбе $\tau_3 \in \rho_{kj}^{k\text{-}1}$ ***M*** БАА-тын q_k күйінен q_j күйіне көшіреді.

Бұл БАА ***M*** диаграммасы II.3.2.8-суретте көрсетілген.



II.3.2.8-сурет. РӨ $\boldsymbol{\rho}$ үшін ДАА ***M***

Сонда $\boldsymbol{L(\rho)} = \bigcup_{q_j \in F} \rho_{1j}^{k}$. Бұл жиынның k бойынша индукция арқылы регулярлық жиын (тіл) болатындығын көрсетуге болады.

II.3.2.1-лемма және II.3.2.2-леммадан кез келген регулярлық жиын үшін дәл осы жиынды танитын ақырлы автомат болатындығы және, керісінше, ақырлы автоматпен танылатын тіл регулярлық жиын болатындығы шығады, яғни келесі теорема қабылданады.

II.3.2.1-теорема. Тіл регулярлық болады тек қана және тек қана сол жағдайда, қашан ол ақырлы автоматпен танылса.

Сонымен, ақырлы автоматар танитын тілдер тобы регулярлық тілдер тобына дәл келеді. Оларды жай ғана автоматтық тілдер немесе регулярлық тілдер деп атауға болады. Осы айғақты әртүрлі тілдік процессорларын жасау кезінде кеңінен пайдаланылады.

II.3.2-тапсырмалар: Мына регулярлық өрнектерге эквивалентті ақырлы автоматтарды тұрғызыңыздар:

1. $(a \vee b \vee ab)^*$.
2. $(a^*b)^* \vee (b^*a)^*$.
3. $(ba \vee a^*ab)^*$.
4. $(b^+a)^*(b \vee a)b^*$.
5. $(b^*a)b \vee (ab^*)a$.
6. $(ba^*) \vee (b \vee a)b^*$.
7. $(a^* \vee b)b \vee (a \vee b^*)a$.
8. $(ba)^*b \vee a\, b^*$.
9. $a(a \vee b)^* \vee b(a \vee b)^*$.

II.3.2-сұрақтар:

1. Бір тілді тудыратын екі регулярлық өрнектің екеуіне де эквивалентті ақырлы автомат табылама?

2. Мына $(0 \vee (1(0^*)))$ және $0 \vee 10^*$ регулярлық өрнектеріне эквивалентті ақырлы автоматтар біп-біріне эквивалентті бола ма?

3. Параметрлері мынадай $\boldsymbol{Q} = \{1, 2\}$; $\mathbf{T} = \{a, b\}$; $\boldsymbol{I} = \{1\}$; $\boldsymbol{F} = \{2\}$; $\Delta = \{<1, aaa, 1>, <1, ab, 2>, <1, b, 2>, <2, \varepsilon, 1>\}$ болатын ақырлы автоматқа эквивалентті регулярлық өрнек бар ма?

4. Мына $(a \vee b)^*$ және $(a \vee b \vee ab)^*$ регулярлық өрнектеріне эквивалентті ақырлы автоматтар біп-біріне эквивалентті бола ма?

5. Мына $(ab \vee b)$ $(a \vee ba)$ және $(a \vee b)^*$ $(b \vee \mathrm{a})^*$ регулярлық өрнектеріне эквивалентті ақырлы автоматтар біп-біріне эквивалентті бола ма?

6. Мына $(a \vee b)^*(b \vee a)^*$ және $(a \vee b \vee ab)^*$ регулярлық өрнектеріне эквивалентті ақырлы автоматтар біп-біріне эквивалентті бола ма?

7. Параметрлері мынадай $\boldsymbol{Q} = \{A, B, C, D, E, F, G\}$; $\boldsymbol{T} = \{a,b,c\}$; $\boldsymbol{I} = A$; $\boldsymbol{F} = \{D\}$; $\delta(A, b) = \{D\}$, $\delta(D, c) = \{F\}$, $\delta(F, a) = \{D\}$, $\delta(F, b) = \{F\}$, $\delta(E, c) = \{C\}$, $\delta(C, b) = \{C\}$,$\delta(A, c) = \{B\}$, $\delta(A, a) = \{A\}$, $\delta(F, c) = \{C\}$, $\delta(E, a) = \{F\}$, $\delta(F, a) = \{E\}$, $\delta(C, a) = \{C\}$,$\delta(F, a) = \{B\}$, $\delta(B, b) = \{A\}$, $\delta(E, c) = \{D\}$ болатын бейдетерминді ақырлы автоматқа эквивалентті регулярлық өрнек бар ма?

II.3.3. Оң сызықты грамматикалар және ақырлы автоматтардың эквиваленттілігі

Бұл параграфта оң сызықты грамматикамен туындайтын оң сызықтық тілдер тобы бейдетерминді ақырлы автоматтармен танылатын тілдер тобына дәл сәйкес келетіндігі көрсетіледі, мысалдар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1,3,4,11,13,14,17,20,24,26,28,34].

Бұл II.3.3.1-лемма бойынша берілген оң сызықты грамматикаға (ОСГ) сәйкес бейдетерминді ақырлы автоматтарды (БАА) тұрғызу, және керісінше II.3.3.1-лемма бойынша берілген БАА-қа сәйкес ОСГ құру арқылы дәлелденеді.

II.3.3.1-лемма. Оң сызықты $\boldsymbol{L}(\boldsymbol{G})$ тілін тудыратын кез келген ОСГ $\boldsymbol{G}$ үшін, сәйкес $\boldsymbol{L}(\boldsymbol{M})$ тілін танитын және $\boldsymbol{L}(\boldsymbol{M})\subseteq\boldsymbol{L}(\boldsymbol{G})$ болатындай етіп АА $\boldsymbol{M}$ тұрғызуға болады.

Дәлелдеу. Мейлі ОСГ $\boldsymbol{G} = <\boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{P}, S>$ берілсін. Жалпылықты шектемей, алғашқы тіл ОСГ-мен туындаған және $A\to\alpha$ (мұнда $\alpha\in\boldsymbol{T}^{+}$) ережені қамтымайды. Сонда БАА $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, q_s, \boldsymbol{F}, \vdash, \dashv, \delta>$ келесідегідей тұрғызылады:

– $\boldsymbol{M}$-ның күйлері $\boldsymbol{G}$-ның бейтерминалдары плюс $\boldsymbol{N}$-ге кірмейтін жаңа p күйі, яғни $\boldsymbol{Q} = \boldsymbol{N}\cup\{p\}$;

– $\boldsymbol{M}$-ның алғашқы күйі ретінде алғашқы бейтерминал S алынады, яғни $q_{\mathrm{s}} = S$;

– егер $\boldsymbol{P}$ $S\to\varepsilon$ ережесін қамтыса, онда $\boldsymbol{F} = \{S, p\}$, әйтпесе $\boldsymbol{F} = \{p\}$;

– егер $\boldsymbol{P}$ $q\to tB$ ережесін қамтыса, мұнда, $t\in\boldsymbol{T}$, $B\in\boldsymbol{N}$, онда $\delta(q,t)=B$ детерминді АА үшін немесе $\delta(q,t)\ni B$ бейдетерминді АА үшін.

$\boldsymbol{M}$ АА τ кірісін оқи отырып, $\boldsymbol{G}$ грамматикасында τ шығысын моделдейді. $\boldsymbol{L}(\boldsymbol{M})\subseteq\boldsymbol{L}(\boldsymbol{G})$ болатындығын дәлелдейді.

Айталық $\tau=t_1t_2...t_n\in\boldsymbol{L}(\boldsymbol{G})$, $n\geqslant 1$ болсын. Сонда $\boldsymbol{G}$ грамматикасында кейбір бейтерминалдар тезбегі A_1, A_2,..., A_{n-1} үшін мына $S\Rightarrow t_1A_1\Rightarrow t_1t_2A_2\Rightarrow\ldots\Rightarrow t_1t_2...t_{n-1}A_{n-1}\Rightarrow t_1t_2...t_n$ шығымдар бар. Анықтама бойынша, $\delta(S, t_1)$ жиынында A_1 қамтылады, $\delta(A_1, t_2)$ жиынында A_2

қамтылады және с.с. $\delta(A_{n-2}, t_{n-1})$ жиынында A_{n-1} қамтылады, $\delta(A_{n-1}, t_n)$ жиынында p қамтылады. Яғни $\tau\in\boldsymbol{L}(\boldsymbol{G})$, себебі $\delta^e(S, \tau)$ жиынынды p қамтылады, ал $p\in\boldsymbol{F}$. Егер $\varepsilon\in\boldsymbol{L}(\boldsymbol{G})$, онда $S\in\boldsymbol{F}$, сондықтан $\varepsilon\in\boldsymbol{L}(\boldsymbol{G})$.

Сондай-ақ, егер тізбе $\tau = t_1t_2...t_n\in\boldsymbol{L}(\boldsymbol{G})$, $n\geqslant 1$, онда $\delta(S, t_1)$ жиынында A_1 қамтылатындай, $\delta(A_1, t_2)$ жиынында A_2 қамтылатындай және с.с. $\delta(A_{n-2}, t_{n-1})$ жиынында A_{n-1} қамтылатындай, $\delta(A_{n-1}, t_n)$ жиынында p қамтылатындай S, A_1, A_2, ..., A_{n-1}, p күйлер тізбегі бар, яғни $\tau\in\boldsymbol{L}(\boldsymbol{G})$ және $\boldsymbol{G}$ грамматикасында мына шығымдар болады:

$$S \Rightarrow t_1 A_1 \Rightarrow t_1t_2A_2 \Rightarrow \ldots \Rightarrow t_1t_2...t_{n-1} A_{n-1} \Rightarrow t_1t_2...t_n$$

Егер $\varepsilon\in\boldsymbol{L}(\boldsymbol{M})$, онда $S\in\boldsymbol{F}$, демек, $S\to\varepsilon\in\boldsymbol{P}$ және $\varepsilon\in\boldsymbol{L}(\boldsymbol{G})$.

II.3.3.2-лемма. $\boldsymbol{L}(\boldsymbol{M})$ тілін танитын кез келген БАА $\boldsymbol{M}$ үшін сәйкес $\boldsymbol{L}(\boldsymbol{G})$ оң сызықты тілді тудыратын және $\boldsymbol{L}(\boldsymbol{G}) \subseteq \boldsymbol{L}(\boldsymbol{M})$ болатындай етіп ОСГ $\boldsymbol{G}$ құруға болады.

Дәлелдеу. Айталық АА $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, q_s, \boldsymbol{F}, \delta>$ берілсін, $q\in\boldsymbol{Q}$, $p\in\boldsymbol{Q}$, $t\in\boldsymbol{T}$, $\tau\in\boldsymbol{T}^*$. Сонда грамматика $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ былай құрылады:

1) $\boldsymbol{M}$ АА-ның күйлері $\boldsymbol{G}$ грамматиканың бейтерминалдары болады, яғни $\boldsymbol{N} = \boldsymbol{Q}$;

2) $\boldsymbol{M}$ АА-ның алғашқы күйі $\boldsymbol{G}$ грамматиканың алғашқы бейтерминалы болады, яғни $S = q_s$;

3) $q\to tp\in\boldsymbol{P}$, егер $p= \delta(q, t)$ детерминді КА үшін иә $p\in\delta(q, t)$ бейдетерминді КА үшін және $(q, t\tau)\models(p, \tau)$;

4) $q\to tp\in\boldsymbol{P}$, егер кейбір $p\in\boldsymbol{Q}$ үшін АА детерминді болғанда $\delta(q,t)=p$ иә АА бейдетерминді болғанда $\delta(q,t)\ni p$, яғни $(q,t\tau)\models(p, \tau)$, мұнда егер $p\in\boldsymbol{F}$, онда $\tau = \varepsilon$;

5) $S\to\varepsilon\in\boldsymbol{P}$, егер $q_s\in\boldsymbol{F}$.

Егер $\boldsymbol{P}$ жиыны $S\to\varepsilon$ ережесін қамтыса, онда S ережелердің оң жағында кездеспейді және $\boldsymbol{F} = \{S, p\}$, әйтпесе $\boldsymbol{F} = \{p\}$.

$\boldsymbol{G}$ грамматикасындағы шығымның әрбір қадамы $\boldsymbol{M}$ автоматының тактысын моделдейтінін көрсетуге болады. Ол үшін i бойынша индукция арқылы $q\in\boldsymbol{Q}$ үшін $q\Rightarrow^{i+1}\tau$ тек қана және тек қана сол жағдайда, қашан кейбір $p\in\boldsymbol{F}$ үшін $(q, \tau)\models^{i}(p, \varepsilon)$ болатындығын дәлелдейміз.

Индукция базисі. i=0 үшін $q\Rightarrow\varepsilon$ тек қана және тек қана сол жағдайда, қашан $q\in\boldsymbol{F}$ үшін $(q, \varepsilon) \models^0 (q, \varepsilon)$.

Индукция қадамы. Айталық i үшін $q\Rightarrow^{i+1}\tau$ ақиқат болсын. $\tau=tx$ деп алайық, мұнда $|x|=i$. Сонда $q\Rightarrow^{i+1}\tau$ кейбір $p\in\boldsymbol{Q}$ үшін $q\Rightarrow tp\Rightarrow^{i}tx$ болатындығы тең мағыналы. Бірақ $q\Rightarrow tp$ және $\delta(q,t)=p$ тең мағыналы. Индукция болжамы бойынша $p\Rightarrow^{i}x$ тек қана және тек қана сол жағдайда, қашан кейбір $r\in\boldsymbol{F}$ үшін $(p,x)\models^{i-1}(r,\varepsilon)$. Демек, $r\in\boldsymbol{F}$ үшін $q\Rightarrow^{i+1}\tau$ және $(q,\tau) \models^{i} (r, \varepsilon)$ тең мағыналы. Сонымен барлық $i\geq 0$ үшін $q\Rightarrow^{i+1}\tau$ тұжырымы ақиқат.

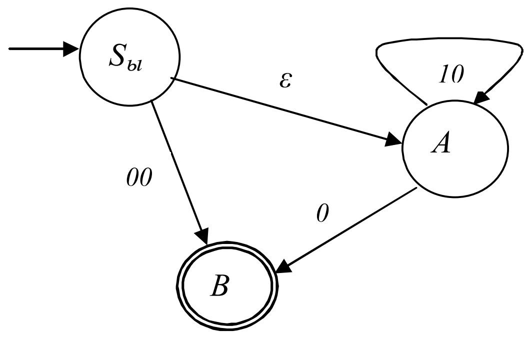
Осыдан $q_0 \Rightarrow^{+} \tau$ тек қана және тек қана сол жағдайда, қашан кейбір $r\in\boldsymbol{F}$ үшін $(q_0,\tau)\models^{*} (r, \varepsilon)$ деп қорытындаймыз. Сонымен, $\boldsymbol{L}(\boldsymbol{G})\subseteq\boldsymbol{L}(\boldsymbol{M})$.

II.3.3.1-лемма және II.3.3.2-леммадан кез келген оң (сол) сызықты тіл үшін дәл осы тілді танитын ақырлы автомат бар болады және керісінше, ақырлы автомат танитын тіл оң (сол) сызықты тіл, яғни келесі теорема қабылданады.

II.3.3.1-теорема. Тіл оң (сол) сызықты болады тек қана және тек қана сол жағдайда, қашан ол ақырлы автоматпен танылса.

Сонымен, ақырлы автомат танитын тіл тобы оң (сол) сызықты тілдер тобымен дәл болады.

II.3.3-мысал. Оң сызықты грамматика $\boldsymbol{G} = \langle N, \boldsymbol{T}, \boldsymbol{P}, S\rangle$ берілсін, мұнда $\boldsymbol{N} = \{S, A\}$ – бейтерминал символдар, $\boldsymbol{T} = \{0,1\}$ – терминал символдар, $\boldsymbol{P} = \{S\to 00, S\to A, A\to 10A, A\to 0\}$ – шешімдер ережелерінің жиыны. Осы сызықты грамматика грамматикамен туындайтын тіл $\boldsymbol{M} = \langle\boldsymbol{Q},\boldsymbol{T},q_s,\boldsymbol{F},\vdash,\dashv, \delta\rangle$ – функция бойынша ($\boldsymbol{M} = \langle\boldsymbol{Q},\boldsymbol{T},q_s,\boldsymbol{F},\vdash,\dashv,\Delta\rangle$ – қатынас бойынша) ақырлы автоматымен танылады. Мейлі оның параметрлері мынадай $\boldsymbol{Q} = \{S, A, B\}$, $q_s = \{S\}$, $\boldsymbol{F} = \{B\}$ және $\delta(S, 00)=B$, $\delta(S, \varepsilon)=A$, $\delta(A, 10)=A$, $\delta(A,0)=B$ – функция бойынша ($\Delta=\{\langle S,00,B\rangle,\langle S,\varepsilon,A\rangle,\langle A,10,A\rangle,\langle A,0, B\rangle\}$ – қатынас бойынша). Осы ақырлы автоматтың диаграммасы II.3.3-суретте көрсетілген.



II.3.3-сурет. ОСГ-ға эквивалентті БАА диаграммасы

II.3.3-тапсырмалар: Оң сызықты грамматикаларға эквивалентті ақырлы автоматарды тұрғызыңыздар:

1. $\boldsymbol{N}=\{S, A\}$, $\boldsymbol{T} = \{a, b\}$, $\boldsymbol{P} = \{S \to aA, A \to aA, A \to b\}$.
2. $\boldsymbol{N}=\{S, A\}$, $\boldsymbol{T} = \{a, b\}$, $\boldsymbol{P} = \{S \to Aa, A \to Aa, A \to b\}$.
3. $\boldsymbol{N}=\{S\}$, $\boldsymbol{T}=\{a\}$, $P=\{S\to a, S\to aaS\}$ $(P=\{S\to a, S\to Saa\}$.
4. $\boldsymbol{N}=\{S, A\}$, $\boldsymbol{T}=\{a, b\}$, $\boldsymbol{P}=\{S\to 0A|1S|\varepsilon, A\to 0B|1A, B\to 0S|1B\}$.
5. $\boldsymbol{N} = \{A,B, D\}$, $\boldsymbol{T} = \{a, b, d\}$, $\boldsymbol{S} = A$, $P=\{A\to aB, B\to aB, B\to b,$ $B\to bD, D\to d, D\to dD, A\to aD, A\to a\}$.
6. $\boldsymbol{N} = \{A,B, D\}$, $\boldsymbol{T} = \{a, b, d\}$, $\boldsymbol{S} = A$, $\boldsymbol{P}=\{A\to aB, B\to aB, B\to b,$ $B\to bD, D\to d, D\to dD, A\to aD, A\to a\}$.
7. $\boldsymbol{N}=\{S,E,F\}$, $\boldsymbol{T}=\{a,b\}$, $\boldsymbol{P}=\{S\to aaE, S\to F, F\to baF,$ $F\to aE, E\to \varepsilon\}$.

II.3.3-сұрақтар:

1.Бір тілді тудыратын екі оң сызықты грамматиканың екеуіне де эквивалентті ақырлы автомат табылама?

2. Мына ережелермен $S\to E$; $S\to bE$; $S\to caE$; $E\to a$; $E\to bS$ туындайтын тілді танитын ақырлы автомат тани ма?

3. Мына $\boldsymbol{N}=\{S\}$, $\boldsymbol{T}=\{a\}$, $P=\{S\to a, S\to aaS\}$ және $\boldsymbol{N}=\{S\}$, $\boldsymbol{T}=\{a\}$, $P=\{S\to a, S\to Saa\}$ грамматикаларға эквивалентті автоматтар тең бе?

4. Мынадай $S\to Sb$, $S\to b$ ережелері бар ОСГ-мен танылатын тіл АА тани ала ма?

5. Мынадай $\boldsymbol{Q}=\{1,2\}$, $\mathbf{T} = \{a,b\}$, $\boldsymbol{I} = \{1\}$, $\boldsymbol{F}=\{2\}$, $\Delta = \{<1,aaa,1>, <1,ab,2>, <1,b,2>,<2,\varepsilon,1>\}$ АА-қа эквивалентті ОСГ бар ма?

II.3.4. Регулярлық тілдердің алгоритмдік проблемалары

Бұл параграфта регулярлық тілдердің алгоритмдік проблемалары қарастырылады, мысалдар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1,3,4,11,13,14,17,20,24,26,28,34].

Алгоритмдік шешілімді проблемаларға мыналар жатады:

1. Тұйықтық проблемасы: "Белгілі бір типтегі тілдерге жиындық амалды қолданғаннан кейін нәтижесі осындай топқа жата ма?"

2. *Тиістілік проблемасы*: "Берілген тізбе ω регулярлық тілге тиісті ме?"

3. *Бостық проблемасы*: "Кез келген регулярлық тіл $\boldsymbol{L}$ үшін, регулярлық тіл $\boldsymbol{L}$ бос болатындығын анықтау керек, яғни $\boldsymbol{L}=\varnothing$?".

4. Қиылысудың бостық проблемасы: "$\boldsymbol{T}$ әліпбиіндегі кез келген екі регулярлық тіл $\boldsymbol{L_1}$ және $\boldsymbol{L_2}$ үшін олардың қиылысуы бос болатындығын анықтау, яғни $\boldsymbol{L_1} \cap \boldsymbol{L_2} = \varnothing$?".

5. Қамту проблемасы: "$\boldsymbol{T}$ әліпбиіндегі кез келген екі регулярлық тіл $\boldsymbol{L_1}$ және $\boldsymbol{L_2}$ үшін бірінші тіл екінші тілде қамтылатындығын анықтау, яғни $\boldsymbol{L_1} \subseteq \boldsymbol{L_2}$?".

6. *Эквиваленттік проблемасы*: "Берілген екі сипаттау әдісі бір регулярлық тілді анықтай ма?"

7. *Шексіздік проблемасы*: "Берілген грамматика $\boldsymbol{G}$ үшін $\boldsymbol{L}(\boldsymbol{G})$ тілі шектелмеген бола ма, яғни $\boldsymbol{L}(\boldsymbol{G}) = \infty$?"

Регулярлық тілдерді анықтаудың барлық әдістері (регулярлық өрнек арқылы, сызықтық грамматика арқылы, ақырлы автоматтар арқылы) бір-біріне эквивалентті болғандықтан, жоғарыда айтылған проблемалардың шешуін тек бір әдіс үшін ғана көрсетуге болады. Ал қалғандары осыған ұқсас болады

Біз осы проблемаларды шешешетін алгоритмдердің біразын регулярлық тілдерді ақырлы автоматтармен анықталған жағдайда ғана қарастырамыз. Алдымен регулярлық тілдер үшін тұйықтылық проблемасының шешетін алгоритмді оң сызықты грамматика арқылы береміз.

II.3.4.1-алгоритм. Оң сызықты грамматикалардың тұйықтылығы.

Кіріс. Оң сызықта грамматикалар $\boldsymbol{G_1} = <\boldsymbol{N_1}, \boldsymbol{T_1}, \boldsymbol{P_1}, S_1>$ және $\boldsymbol{G_2} = <\boldsymbol{N_2}, \boldsymbol{T_2}, \boldsymbol{P_2}, S_2>$, $\boldsymbol{N_1} \cap \boldsymbol{N_2} = \varnothing$.

Шығыс. "ИӘ", егер $\boldsymbol{L}(\boldsymbol{G_1}) \cup \boldsymbol{L}(\boldsymbol{G_2})$ тілі ына грамматикамен $\boldsymbol{G} = <\boldsymbol{N_1} \cup \boldsymbol{N_2} \cup \{S\}, \boldsymbol{T}, \boldsymbol{P_1} \cup \boldsymbol{P_2} \cup \{S \to S_1, S \to S_2\}, S>$ туындаса,

мұнда $S \notin \boldsymbol{N}_1 \cup \boldsymbol{N}_2 \cup \boldsymbol{T}$.

"ЖОҚ" кері жағдайда.

Әдіс. S-тен шығатын шығарымдарды алу. Егер нәтиже $\boldsymbol{T}$-дағы терминалдардың тізбесі болса, онда "ИӘ" деп айту, кері жағдайда "ЖОҚ" деп айту керек.

Регулярлық тілдер үшін тиістілік проблемасы ақырлы автомат арқылы келесі алгоритммен шешіледі.

II.3.4.2-алгоритм. Ақырлы автоматтар үшін тиістілік проблеманы шешу.

Кіріс. Ақырлы автомат $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, q_0, \boldsymbol{F}, \delta>$ және тізбе $\omega \in \boldsymbol{T}^*$.

Шығыс. "ИӘ", егер $\omega \in \boldsymbol{L}(\boldsymbol{M})$; "ЖОҚ", егер $\omega \notin \boldsymbol{L}(\boldsymbol{M})$.

Әдіс. Мейлі $\omega = a_1 a_2 ... a_n$. Біртіндеп күйлерді табу $q_1 = \delta(q_0, a_1)$, $q_2 = \delta(q_1, a_2)$,... $q_n = \delta(q_{n-1}, a_n)$. Егер $q_n \in F$, онда "ИӘ"; егер $q_n \notin F$, онда "ЖОҚ".

Регулярлық тілдер үшін бостық проблемасы ақырлы автомат арқылы келесі алгоритммен шешіледі.

II.3.4.3-алгоритм. Ақырлы автоматтар үшін бостық проблеманы шешу.

Кіріс. Ақырлы автомат $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, q_0, \boldsymbol{F}, \delta>$.

Шығыс. "ИӘ", егер $\boldsymbol{L}(\boldsymbol{M}) \neq \varnothing$, "ЖОҚ" басқа жағдайда.

Әдіс. q_0-ден қолжетілімді күйлер жиынын есептеу. Егер бұл жиын қандай-да бір ақырлы күйді қамтыса, онда "ИӘ" деп, ал кері жағдайда "ЖОҚ" деп айту керек.

Екі регулярлық тілдер қиылысуының бостығы келесі теорема арқылы анықталады.

II.3.4.1-теорема. Регулярлық тілдер тобы үшін қиылысудың бостық проблемасы алгоритмдік шешілімді.

Дәлелдеу. Егер регулярлық тілдер $\boldsymbol{L}_1$ және $\boldsymbol{L}_2$ бейдетерминді ақырлы $\boldsymbol{M}_1$ және $\boldsymbol{M}_2$ автоматтарымен танылса, онда $\boldsymbol{L} = \boldsymbol{L}_1 \cap \boldsymbol{L}_2$ тілін танитын $\boldsymbol{M}$ автоматын тұрғызып, $\boldsymbol{L}$ тілі үшін бостық проблемасын шешуге болады.

Екі регулярлық тіл үшін қамту проблемасы келесі теорема арқылы анықталады.

II.3.4.2-теорема. Регулярлық тілдер үшін қамту проблемасы алгоритмдік шешілімді.

Дәлелдеу. Регулярлық тілдер үшін қамту проблемасының алгоритмдік шешілімділігі келесі эквиваленттіліктен шығады

$$\boldsymbol{L_1} \subseteq \boldsymbol{L_2} \Leftrightarrow \boldsymbol{L_1} \cap (\boldsymbol{T}^* \setminus \boldsymbol{L_2}) = \varnothing$$

Екі регулярлық тіл үшін эквиваленттік проблема ақырлы автоматтар арқылы келесі алгоритммен шешіледі.

II.3.4.4-алгоритм. Ақырлы автоматтардың эквиваленттік проблемасын шешу.

Кіріс. $\boldsymbol{Q}_1 \cap \boldsymbol{Q}_2 = \varnothing$ болатындай екі АА $\boldsymbol{M}_1 = <\boldsymbol{Q}_1, \boldsymbol{T}_1, q_1, \boldsymbol{F}_1, \delta_1>$ және $\mathrm{M}_2 = <\boldsymbol{Q}_2, \delta_2, q_2, F_2, \boldsymbol{T}_2>$.

Шығыс. "ИӘ", егер $\boldsymbol{L}(\boldsymbol{M}_1) = \boldsymbol{L}(\boldsymbol{M}_2)$, "ЖОҚ" басқа жағдайда.

Әдіс. $\boldsymbol{M} = <\boldsymbol{Q}_1 \cup \boldsymbol{Q}_2, \boldsymbol{T}_1 \cup \boldsymbol{T}_2, q_1, F_1 \cup F_2, \delta_1 \cup \delta_2>$ ақырлы автоматын тұрғызу.

Автоматтардың q_1 және q_2 күйлері ерекшеленетінін анықтаймыз. Егер ерекшеленсе, онда "ЖОҚ" деп, ал кері жағдайда "ИӘ" деп айтамыз.

Эквиваленттік проблеманы шешу үшін II.3.4.2 алгоритмін пайдаланамыз, себебі $\boldsymbol{L}(\boldsymbol{M}_1) = \boldsymbol{L}(\boldsymbol{M}_2)$ тек қана сонда және тек қана сонда, қашан $(\boldsymbol{L}(\boldsymbol{M}_1) \cap \boldsymbol{L}(\boldsymbol{M}_2)) \cup (\boldsymbol{L}(\boldsymbol{M}_2) \cap \boldsymbol{L}(\boldsymbol{M}_1)) = \varnothing$ немесе мына эквиваленттікті $\boldsymbol{L}_1 = \boldsymbol{L}_2 \Leftrightarrow (\boldsymbol{L_1} \subseteq \boldsymbol{L_2})\ \&\ (\boldsymbol{L_2} \subseteq \boldsymbol{L_1})$ пайдаланамыз.

Енді ақырлы автоматты тілдің ақырсыздығы проблемасының шешілуін көрсетейік. Ол үшін келесі теорема дәлелденеді.

II.3.4.3-теорема. n күйлі $\boldsymbol{M}$ ақырлы автоматпен танылатын тізбелер жиыны ақырсыз тек қана сонда және тек қана сонда, қашан ол ұзындығы l, $n \le \boldsymbol{L} < 2n$ тізбесін қамтыса.

Дәлелдеу. *Қажеттілік* кері әдіспен дәлелденеді. Мейлі $\boldsymbol{M}$ ақырлы автомат тізбелердің ақырсыз жиынын таниды және олардың ешқайсысы l, $n \le \boldsymbol{L} < 2n$ ұзындықты болмайды. Егер $\boldsymbol{L}(\boldsymbol{M})$ жиыныны тек ұзындығы $\boldsymbol{L} < n$ тізбелерден тұрса, онда тіл ақырлы болар еді, бірақ ол ондай емес. Сондықтан ұзындығы $l \ge 2n$ болатындай тізбе бар болады. Мейлі $x \in \boldsymbol{L}(\boldsymbol{M})$ және $|x| \ge 2n$ болатындай – ең қысқа тізбе. Ал $x = x_1x_2x_3$, мұнда $1 \le |x_2| \le n$ және $\delta(q_0, x_1) = q$, $\delta(q, x_2) = q$, $\delta(q, x_3) \in F$ болсын. Бірақ сонда $x_1x_3 \in \boldsymbol{L}(\boldsymbol{M})$, себебі $\delta(q_0, x_1x_3) = \delta(q_0, x_1x_2x_3) \in F$ және $|x_1x_3| \ge n$ (мұнда $|x| = |x_1x_2x_3| \ge 2n$ және $1 \le |x_2| \le |n|$). Себебі болжау бойынша $\boldsymbol{L}(\boldsymbol{M})$ тілінде ұзындығы $n \le \boldsymbol{L} < 2n$ болатындай тізбе жоқ, сондықтан $|x_1x_3| \ge 2n$. Демек, болжауға қайшы $x = x_1x_2x_3 \in \boldsymbol{L}(\boldsymbol{M})$ – ұзындығы $2n$-нен үлкен немесе тең ең қысқа тізбе және ұзындығы $2n$-нен үлкен одан да қысқалау $x_1x_3 \in \boldsymbol{L}(\boldsymbol{M})$ тізбе табылды. Бұл қайшылық қажеттілікті дәлелдейді.

Жеткіліктілік келесі болжаудан шығады. Мейлі $x \in \boldsymbol{L}(\boldsymbol{M})$, мұнда $n \le |x| < 2n$. Бұрынғыдай, $q \in Q$, $x = x_1x_2x_3$ бар деп тұжырымдаймыз, мұнда $x_2 \ne \varepsilon$ және $\delta(q_0, x_1) = q$, $\delta(q, x_2) = q$, $\delta(q, x_3) \in F$. Бірақ кез келген i үшін тізбе $x_1x_2^ix_3 \in \boldsymbol{L}(\boldsymbol{M})$. Сондықтан $\boldsymbol{L}(\boldsymbol{M})$ жиыны ақырсыз. Бұл талап етілген дәлелдеме.

Енді мына теореманы дәлелдеуге болады.

II.3.4.4-теорема. Егер жиын регулярлық өрнекпен, оң сызықты грамматикамен немесе ақырлы автоматпен анықталса, онда регулярлық тілдер үшін тұйықтылық, тиістілік, бостық, эквиваленттілік және ақырсыздық проблемалары алгоритмдік шешілімді.

III.3.4-мысал. Мейлі $\boldsymbol{T} = \{a, b, c\}$. Сонда $\boldsymbol{L} = \boldsymbol{T}^* \setminus \{a^nb^nc^n | n > 0\}$ тілі оң сызықты болады, себебі $\boldsymbol{L} = \boldsymbol{L}_1 \cup \boldsymbol{L}_2 \cup (\boldsymbol{T}^* \setminus \boldsymbol{L}_3)$, мұнда тілдер

$\boldsymbol{L}_1 = \{a^m b^n c^k : m \geq n, k > 0\}$ және $\boldsymbol{L}_2 = \{a^m b^n c^k : n \geq k, m > 0\}$ оң сызықты, ал $\boldsymbol{L}_3 = \{a^m b^n c^k : m > 0, n > 0, k > 0\}$ автоматты болғандықтан.

III.3.4-тапсырмалар:

1. Мына берілген екі регулярлық тілді $\boldsymbol{L}(\boldsymbol{G}_1) = \{a, b\}^*$ және $\boldsymbol{L}(\boldsymbol{G}_2) = \{a, b, c\}^*$ тудыратын екі сызықтық грамматика $\boldsymbol{G}_1$ және $\boldsymbol{G}_2$, үшін $\boldsymbol{L}(\boldsymbol{G}_1) \cap \boldsymbol{L}(\boldsymbol{G}_2)$ өрнегінің мәнін есептеңіз.

2. Мына регулярлық тіл $\{\gamma\delta: \gamma \in \{a, b\}^*, \delta \in \{a, b\}^*\}$ үшін оны тудыратын сызықтық грамматиканы және оны танитын ақырлы автоматты табыңыз.

3. Мына регулярлық тіл $\{a\xi b: \xi \in \{a, b\}^*\} \cup \{b\xi a: \xi \in \{a, b\}^*\}$ үшін оны тудыратын регулярлық өрнекті және сызықтық грамматиканы табыңыз.

4. Мына регулярлық тіл $\{\xi b: \xi \in b\{a, b\}^*\} \cup \{a\xi: \xi \in a\{a, b\}^*\}$ үшін оны тудыратын регулярлық өрнекті және сызықтық грамматиканы табыңыз.

III.3.4-сұрақтар:

1. Берілген терминалдық тізбе оң сызықты грамматикамен туындайтын регулярлық тілге жататындығын анықтайтын алгоритм бар ма, жоқ па?

2. Ақырлы автомат арқылы танылатын регулярлық тілдің бос екендігін анықтайтын алгоритм бар ма, жоқ па?

3. Екі регулярлық өрнек бір ғана регулярлық тілді тудыра алатындығын анықтайтын алгоритм бар ма, жоқ па?

4. Екі оң сызықты грамматика бір ғана регулярлық тілді тудыра алатындығын анықтайтын алгоритм бар ма, жоқ па?

5. Екі ақырлы автомат бір ғана регулярлық тілді тани алатындығын анықтайтын алгоритм бар ма, жоқ па?

6. Мына ережелері $S \to aaCM$, $S \to aaaKF$, $M \to aMb$, $M \to bMa$, $M \to \varepsilon$, $C \to aCa$, $C \to bCb$, $K \to bT$, $K \to aT$, $T \to bKa$, $T \to ab$ бар грамматикамен туындаған тіл бос бола ма?

7. Мына ережелермен $S \to aSb$, $S \to aSa$, $S \to bSa$, $S \to bSb$, $S \to a$, $S \to b$ туындаған тіл автоматты бола ма?

III. КОНТЕКСТІСІЗ ТІЛДЕ*Р*

III.1. Констекстісіз тілдерді тудырушы механизмдер

III.1.1. Констекстісіз грамматика

Бұл бөлімде констекстісіз (контексті-бос) тілдер қарастыралады, оларды тудыру және тану механизмдер беріледі, контекстісіз грамматикалар мен стектік автоматтардың эквиваленттілігі көрсетіледі, сондай-ақ, мысалдар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1-9,11-13,15-22,24,25, 27-32].

III.1.1.1-анықтама. Контекстісіз грамматикалар тудыратын тілдер констекстісіз тілдер деп аталады.

Контекстісіз (констексті-бос) грамматикалар деп барлық шығарым ережелері мынадай $A \rightarrow \alpha$, $A \in \boldsymbol{N}$, $\alpha \in (\boldsymbol{T} \cup \boldsymbol{N})^*$ түрде болатын, яғни A бейтерминалы A кездесетін контекстке байланыссыз терминалдар мен бейтерминалдар жиынындағы α тізбесіне алмастыралады (**I.3.2.**).

Констексті бос грамматикалар (КБГ) тілдер теориясында маңызды орын алады және олар шығарым ережелерін тізбектей қолданудан туатын тізбенің синтаксистік құрылымын анықтауға қызмет етеді.

III.1.1.1-мысал. Мейлі $\boldsymbol{G_1} = <\boldsymbol{T},\boldsymbol{N},\boldsymbol{P},S>$, мұнда $\boldsymbol{N}=\{S,A,B\}$, $\boldsymbol{T}=\{a,b\}$, $\boldsymbol{P} = \{S\rightarrow aB, S\rightarrow bA, A\rightarrow aS, A\rightarrow bAA, B\rightarrow bS, A\rightarrow a, B\rightarrow aBB, B\rightarrow b\}$**.**

$\boldsymbol{G_1}$ грамматикасы контекстісіз болады, себебі оның әрбір шығарым ережелерінің сол жағы жалғыз ғана бейтерминалдан, ал оң жағы бос емес терминалдар мен бейтерминалдардың тізбесінен тұрады. $\boldsymbol{G_1}$ грамматикасындағы әдеттегі шығарымдар мынадай:

$$(S \Rightarrow aB \Rightarrow ab,$$
$$S \Rightarrow aB \Rightarrow abS \Rightarrow abbA \Rightarrow abba,$$
$$S \Rightarrow bA \Rightarrow ba,$$
$$S \Rightarrow bA \Rightarrow bbAA \Rightarrow bbaA \Rightarrow bbaa$$

Қос нүкте алдында a және b символдарының тең санынан тұратын тізбелердің барлық жиындарын тудыруға қолданылған шығарым ережелерінің нөмірлері көрсетілген.

КБГ-да бір орында бірдей ережелер әртүрлі ретпен қолданылатын бірнеше эквиваленттi шығарымдар болуы мүмкін. Кез келген КБГ үшін екі шығарымның эквиваленттілігін анықтау қиын, бірақ КБГ жағдайында эквиваленттi шығарымдар сыныбын шығарымдар дарағы деп аталатын графикалық түрде көрсетуге болады.

III.1.1.2-анықтамалар. Таңбалы реттелген $\boldsymbol{D}$ дарағы $\boldsymbol{G}(S) = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ КБГ-ндағы шығарым дарағы (немесе талдау дарағы) деп аталады, егер төменгі шарттар орындалса:

(1) $\boldsymbol{D}$ дарағының тамыры (ешбір доға кірмейтін төбе) S символымен таңбаланған;

(2) Егер $\boldsymbol{D}_1,....,\boldsymbol{D}_k$ – дарақ тамырының тікелей тармақтары басым болатын ішкі дарақтар болсын және $\boldsymbol{D}_i$ дарағының тамыры X_i таңбаланған, онда $A \rightarrow X_1 \ldots X_k$ өрнегі $\boldsymbol{P}$ жиынының ережелері.

(3) Егер X_i бейтерминал болып, $\boldsymbol{D}_i$ дарағы X_i арқылы белгіленген жалғыз ғана төбеден тұрса және X_i терминал болса, онда $\boldsymbol{D}_i$ кез келген i=1, 2, …, k үшін $\boldsymbol{G}(X_i) = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, X_i>$ грамматикасындағы шығарым дарағы болады.

(4) Егер дарақ тамырының ε арқылы таңбаланған жалғыз ұрпағы болса, онда осы ұрпақ жалғыз төбелі дарақты құрайды және $S \rightarrow \varepsilon$ өрнегі $\boldsymbol{P}$ жиынының ережесі болады.

Сонымен, шығарым дарағының әрбір төбесі $\boldsymbol{N} \cup \boldsymbol{T} \cup \{\varepsilon\}$ жиынының символымен таңбаланады. Мұнда, егер дарақтың ішкі төбесі A символымен, ал оның тікелей ұрпақтары $X_1, X_2, \ldots, X_n$ символдарымен таңбаланса, онда мына өрнек $A \rightarrow X_1 X_2 \ldots X_n$ грамматиканың шығарым ережесі болады.

$\boldsymbol{G} = <\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{P}, S>$ КБГ үшін шығарым дарағы былай құрылады:

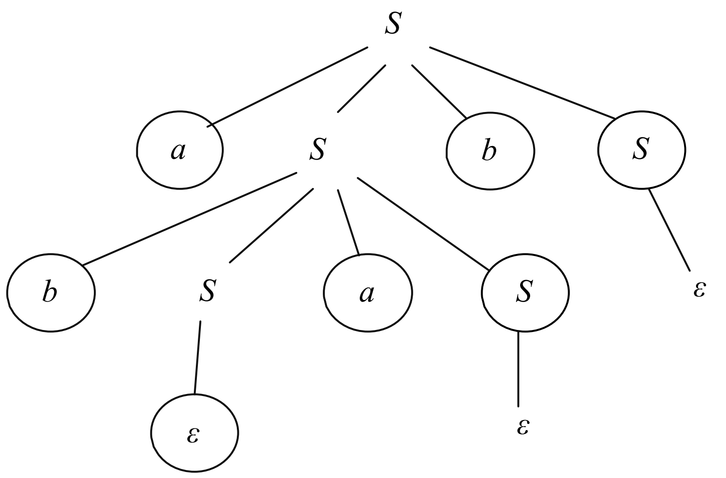
1. Дарақ төбелері қатал ретпен анықталған $\boldsymbol{T} \cup \boldsymbol{N}$ жиынының символдарымен таңбаланады.

2. Егер X тамғалы төбенің ең болмаса бір бағыныңқы төбесі болса, онда $X \in \boldsymbol{N}$. Мұнда дарақ тамыры $S \in \boldsymbol{N}$ таңбалады.

3. Егер $X_1, X_2, \ldots, X_k$ төбелері S төбесіне тіке бағыныштты болса, онда $S \rightarrow X_1, X_2, \ldots, X_k$ ережесі $\boldsymbol{P}$ жиынына кіру керек.

Реттелген дарақ төбесінің тіке ұрпақтары “солдан оңға” реттелетін төбелердің табиғи реті бар болатынын ескеру керек.

ІІІ.1.1-мысал. ІІІ.1.1.1-суретте ережелері $S \rightarrow aSbS|bSaS|\varepsilon$ бар $\boldsymbol{G_2} = \boldsymbol{G}(S)$ грамматикасындағы шығарым дарағының көрсетілген.



ІІІ.1.1.1-сурет. Шығарым дарағының мысалы.

X – төбе және X_1, ... , X_k – оның тікелей тармақтары болсын. Егер $i<j$ ($i=1,2, ..., k$, $j=1,2, ..., k$), болса, онда X_i төбесі мен оның барлық тармақтары X_j төбесі мен оның барлық тармақтарының сол жағына орналасқан деп есептеледі.

Мейлі $\boldsymbol{G} = <\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{P}, S>$ КБГ-нда $\boldsymbol{D}$ – шығарым дарағы берілсін. Сонда мынадай жаңа ұғымдарды енгізуге болады.

ІІІ.1.1.3-анықтамалар:

1.Шығарым дарағының *шыңы* деп жапырақтар тамғаларын солдан оңға қарай жазып шыққанда пайда болатын тізбені айтады.

2.Шығарым дарағының *қимасы* деп келесі шарттарды қанағаттандыратын осы дарақтың $\boldsymbol{C}$ төбелер жиынын айтамыз,

(1) $\boldsymbol{C}$ жиынының ешқандай да екі төбесі $\boldsymbol{D}$ дарағындағы бір жолда жатпайды;

(2) $\boldsymbol{D}$ дарағының ешқандай да төбесін (1) қасиетін бұзбай $\boldsymbol{C}$ жиынына қосуға болмайды.

$\boldsymbol{G}$ КБГ-дағы әрбір шығарылатын α тізбесі үшін $\boldsymbol{G}$-да шыңы α болатын шығарым дарағын салуға және керісінше болатындығын көрсетуге болады

III.1.1.4-анықтама*.** ***D дарағы *қимасының шыңы* деп қандай да бір қима құрайтын төбелер таңбаларын солдан оңға қарай тіркестіру арқылы жасалынатын тізбені айтады.

III.1.1.3-мысал. III.1.1.2-суретте көрсетілген шығарым дарағы қимасының шыңы *abaSbS* тізбесі болады.

Мейлі $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ КБГ болсын. Сонда $S \Rightarrow^* \tau$ болатындай ***G*-**да τ шыңды ***D*** шығарым дарағы бар болады.

Мейлі $C_0, C_1, C_2, \ldots, C_n$ − келесі шарттарға сай болатын ***D*** дарағының қималар тізбесі болсын:

(1) C_0 тек ***D*** дарағының тамырынан тұрады,

(2) $0 \le i < n$ үшін C_{i+1} қимасы C_i қимасынан бір бейтерминал төбені оның тіке ұрпағымен алмастырғаннан шығады;

(3) C_n – ***D*** дарағының шыңы.

Егер $S \Rightarrow^* \tau = \alpha_0, \alpha_1, \ldots, \alpha_n$ − терминалды τ тізбесінің сол жақ шығарымы болса, онда әрбір α_i $(0 \le i < n)$ тізбесі $x_i A_i \beta_i$ түрінде болады, мұндағы $x_i \in \boldsymbol{T}^*$, $A_i \in \boldsymbol{N}$ және $\beta_i \in (\boldsymbol{N} \cup \boldsymbol{T})^*$. Сол жақ шығарымның әрбір келесі α_{i+1} тізбесі алдыңғы α_i тізбесінің ең сол бейтерминал A_i, символын кейбір ережелердің оң жағымен алмастырғаннан пайда болады. Дәл осылай оң жақ шығарымда ең оң бейтерминал алмастырылады.

***III.1.1.5-анықтамалар*:**

1. Егер C_{i+1} қимасы C_i қимасынан ең сол бейтерминал төбесін оның тікелей тармақтарымен алмастыру арқылы алынса, онда сәйкес $\alpha_0, \alpha_1, \ldots, \alpha_n$ шығарымы ***G*** грамматикасындағы α_0 тізбесінен α_n тізбесінің *сол жақ шығарымы* деп аталады. Оң жақ шығарым да дәл осылай анықталады, тек "ең сол" дегенді "ең оң" деп оқу керек. Шығарым дарағы бойынша сол (оң) шығарым бірмәнді анықталатынын байқаймыз.

2. τ тізбесі ***G*** грамматикасында *солшығарымды* деп аталады, егер $S \Rightarrow^* \tau$ сол жақ шығарымы бар болса және ол былай $S \Rightarrow^*_{\boldsymbol{G}l} \tau$ (немесе $S \Rightarrow^*_l \tau$, егер қандай грамматика қолданылатыны белгілі болса) жазылады.

3. τ тізбесі $\boldsymbol{G}$ грамматикасында *оңшығарымды* деп аталады, егер $S \Rightarrow^* \tau$ оң жақ шығарымы бар болса және ол былай $S \Rightarrow^*_{G\mathrm{r}} \tau$ (немесе $S \Rightarrow^*_{\mathrm{r}} \tau$) жазылады.

Яғни сол жақ шығарымның бір қадамын $\Rightarrow_l$ арқылы, ал оң жақ шығарымның бір қадамын $\Rightarrow_{\mathrm{r}}$ арқылы белгілейміз.

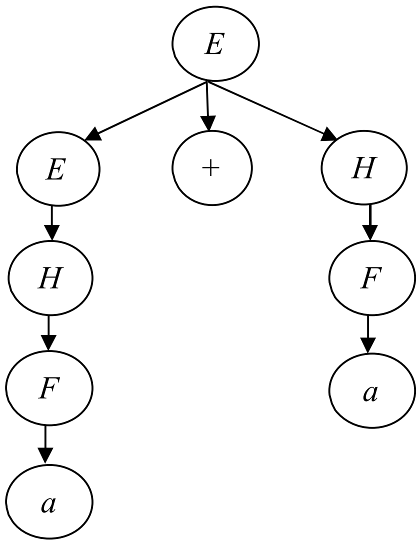
III.1.1.4-мысал. Мейлі $\boldsymbol{G}_a$ КБГ-ның ережелері төмендегідей

$$E \to E+H|H,\ H \to H * F|F,\ F \to (E)|a.$$

III.1.1.3-суретте көрсетілген шығарым дарағы $a+a$ тізбесінің $\boldsymbol{G}_a$ КБГ-ндағы бір-біріне эквивалентті екі шығарымын бейнелейді:

сол жақ шығарымы $E \Rightarrow E+H \Rightarrow H+H \Rightarrow F+H \Rightarrow a+H \Rightarrow a+F \Rightarrow a+a$,

оң жақ шығарымы $E \Rightarrow E+H \Rightarrow E+F \Rightarrow E+a \Rightarrow H+a \Rightarrow F+a \Rightarrow a+a$.



III.1.1.3-сурет. $a+a$ тізбесінің дарағы.

III.1.1.6-анықтама. $\boldsymbol{G}$ КБГ-ндағы екі немесе одан да көп шығарым дарақтарынының шыңы болып табылатын қайсыбір $\tau \in \boldsymbol{L}(\boldsymbol{G})$ тізбесі табылса, онда $\boldsymbol{G}$ *бірмәнді емес* деп аталады. Яғни қандай да бір $\tau \in \boldsymbol{L}(\boldsymbol{G})$ тізбесінің сол (оң) жақ шығарымдары болғаны. Кері жағдайда $\boldsymbol{G}$ КБГ *бірмәнді* деп аталады.

III.1.1.5-мысал. Жоғарыда III.1.1.5-мысалдағы КБГ $\boldsymbol{G}_5$ бір мәнді емес, себебі онда $a+a$ тізбесінің екі немесе одан да көп сол жақ (оң жақ) шығарымдары бар.

сол шығарым $E \Rightarrow E+H \Rightarrow H+H \Rightarrow F+H \Rightarrow a+H \Rightarrow a+F \Rightarrow a+a$,

оң шығарым $E \Rightarrow E+H \Rightarrow E+F \Rightarrow E+a \Rightarrow H+a \Rightarrow F+a \Rightarrow a+a$.

III.1.1-тапсырмалар:

1. Мынадай $S \to S0|S1|D0|D1$, $D \to H$, $H \to 0|1|H0|H1$ ережелері бар КБГ-нда 10.1001 тізбесі үшін шығарым дарағын тұрғызыңыз.

2. Мынадай $S \to$ *i*f *B* the*n* *S*|*B* = *E*, $E \to B|B+E$, $B \to a|b$ ережелері бар КБГ-нда *i*f *a* the*n* *b* = *a*+*b*+*b* тізбесі үшін шығарым дарағын тұрғызыңыз.

3. Мына тілді $\boldsymbol{L}= \{a^{2n}b^m c^{2k} : m=n+k,\ m>1\}$ тудыратын КБГ-н жазыңыз, шығарым дарағын және *aabbbcccc* тізбесі үшін сол жақ шығарымды тұрғызыңыз.

4. Мына тілді $\boldsymbol{L}= \{1^{3n+2}\ 0^n : n \geq 0\}$ тудыратын КБГ-н жазыңыз, шығарым дарағын және 1111111100 тізбесі үшін оң жақ шығарыммен сол жақ шығарымды тұрғызыңыз.

5. $\boldsymbol{T} = \{a, b\}$ әліпбиінде *b* әрпі *n* рет қайталанатын $\boldsymbol{L}_1$ және $\boldsymbol{L}_2$ тілдерін тудыратын КБГ-н құрыңыз $\boldsymbol{L}_1=\{ab^n : n \geq 0\}$, $\boldsymbol{L}_2=\{b^n a : n \geq 1\}$.

6. Конъюнция & және дизъюнкция ∨ амалдары арқылы құралған және >, <, = қатынастарымен бірегетін дұрыс логикалық өрнектерді тудыратын КБГ-н жазыңыз. Өрнектер үлгісі: $x=a$ & $x>b \vee x<c$.

7. $\boldsymbol{L}=\{a^{2n}b^m c^{2k} : m=n+k,\ m>1\}$ тудыратын және *aabbbcccc* тізбесі үшін шығарым дарағы мен сол жақ шығарымды тұрғызыңыз.

Мейлі КБГ-да мынадай $S \to AB$, $A \to Aa|bB$, $B \to a|Sb$ ережелер болсын. Мына *Baabaab*, *bBABb*, *baSb* тізбелердің шығарым дарақтарын тұрғызыңыз.

III.1.1. Сұрақтар

1. КБГ-нда ережелерге қандай шарттар бар?
2. Шығарым дарағы деген не?
3. КБГ-нда сол жақ (оң жақ) шығарым қалай анықталады?
4. КБГ-ның бір мәнді емес екендігі қалай анықталады?
5. Шығарым дарағының тамыры мен оның шыңының айрмашылығы неде?
6. Мына тілдің $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{a^n b^n c^n : n>1\}$ типі қандай?
7. Мына тілдің $\boldsymbol{L} = \{1^{3n+2}\ 0^n : n \geq 0\}$ типі қандай?
8. Мына $\{a+a*a\}$ жиыны қандай грамматикамен туындайды?

III.1.2. Контекстісіз грамматикаларды түрлендіру

Бұл параграфта контекстісіз грамматикаларды түрлендірудің әртүрлі әдістерін қарастырамыз, мысалдар ұсынылады, тапсырмалар беріледі және сұрақтар қойылады [1-9,11-13,15-22,24,25, 27-32].

Кейде берілген грамматиканы, ол тудыратын тіл қажетті құрылымға ие болу үшін жиі жетілдіру керек. Мысалы, $\boldsymbol{L}(\boldsymbol{G}_a)$ тілін қарастырайық. Бұл тіл $\boldsymbol{G}_a$ грамматикасында келесі ережелер $E \to E+E|E*E|(E)|a$ бойынша туындайды.

$\boldsymbol{G}_a$ грамматикасының екі кемшілігі бар. Оның біріншісі, грамматикада $E \to E+E|E*E|(E)|a$ ережесі болғандықтан ол бір мәнді емес. Бұл кемшіліктен $\boldsymbol{G}_a$ грамматикасының орнына ережелері мынадай $E \to E+H|E*H|H$, $\mathrm{H} \to (\mathrm{E})|a$ болатын $\boldsymbol{G}_2$ грамматикасын алу арқылы құтылуға болады:

Екінші кемшілік $\boldsymbol{G}_a$ грамматикасында және $\boldsymbol{G}_1$ грамматикасында да бар. Ол+және * амалдарының басымдылықтары бірдей екедігінде. Яғни, $\boldsymbol{G}_1$ грамматикасы беретін $a+a*a$ және $a*a+a$ өрнектерінің құрылымдары олардағы амалдардың орындалу ретін сәйкесінше $(a+a)*a$ және $(a*a)+a$ өрнектеріндегідей етіп береді.

Әуелі * амалы сонан соң+амалы орындалатындай яғни $a+a*a$ өрнегін $a+(a*a)$ ретінде түсінетіндей жағдайға жету үшін $\boldsymbol{G}_a$ грамматикасына көшу керек.

Берілген тілге қандай да бір құрылым беретіндей алгоритмдік әдіс жоқ. Дегенмен түрлендіру қатары көмегімен тудыратын тілді бүлдірмей грамматиканың түрін өзгертуге болады. Төменде осындай түрлендірулерді қарастырамыз.

Кейбір жағдайларда КБГ құрамында пайдасыз символдар мен ережелер болуы мүмкін. Мысалы, $\boldsymbol{G} = \langle\{S,A\}, \{a,b\}, \boldsymbol{P}, S\rangle$, мұнда $\boldsymbol{P}=\{S \to a, A \to b\}$ грамматикасында бейтерминал A мен терминал b ешқандай да шығарым тізбесінде кездеспейді. Бұл символдардың $\boldsymbol{L}(\boldsymbol{G})$ тіліне еш қатысы жоқ және оларды $\boldsymbol{G}$ грамматикасы анықтамасынан $\boldsymbol{L}(\boldsymbol{G})$ тіліне тиіспестен алып тастауға болады.

III.1.2.1-анықтама. Мейлі $\gamma\in\boldsymbol{T}^*$, $\delta\in\boldsymbol{T}^*$, $\omega\in\boldsymbol{T}^*$ және $X\in\boldsymbol{N}$ болсын. Егер X бейтерминал символы $S\Rightarrow^*\gamma X\delta\Rightarrow^*\omega$ түріндегі шығарымда қатыса алса, онда ол $\boldsymbol{G}$ = <$\boldsymbol{N}$, $\boldsymbol{T}$, $\boldsymbol{P}$, S> КБГ-нда *пайдалы* (*useful*) деп, әйтпесе ол *пайдасыз*(*useless*) деп аталады.

A бейтерминалының пайдасыз екендігін білу үшін бейтерминал қайсы бір терминал тізбелерін тудыра алатындығын анықтайтын алгоритм құрамыз, яғни біз мына $\{\tau\colon A\Rightarrow^*\tau,\ \tau\in\boldsymbol{T}^*\}$ жиынының бостық проблемасын шешетін алгоритм құрамыз. Осындай алгоритмнің болғандығынан КБГ үшін бостық проблемасының шешімді екендігі шығады.

III.1.2.1-алгоритм. L(G) тілі бос емес пе?

Кіру. КБГ $\boldsymbol{G}$ = <$\boldsymbol{N}$, $\boldsymbol{T}$, $\boldsymbol{P}$, S>.

Шығу. "ИЯ", егер $\boldsymbol{L}(\boldsymbol{G})\neq\varnothing$, "ЖОҚ" кері жағдайда.

Әдіс. Рекурсивті түрде $\boldsymbol{N}_0$, $\boldsymbol{N}_1$ ... жиындарын құрамыз:

(1) $\boldsymbol{N}_0=\varnothing$ және i=1 деп алу.

(2) $\boldsymbol{N}_i=\{A|A\rightarrow\alpha\in\boldsymbol{P}$ және $\alpha\in(\boldsymbol{N}_{i-1}\cup\boldsymbol{T})^*\}\cup\boldsymbol{N}_{i-1}$.

(3) Егер $\boldsymbol{N}_i=\boldsymbol{N}_{i-1}$, онда $i=i+1$ деп алып (2) қадамға көшу. Кері жағдайда $\boldsymbol{N}_e=\boldsymbol{N}_i$ деп алу.

(4)Егер $S\in\boldsymbol{N}_e$, онда "ИЯ", кері жағдайда "ЖОҚ".

Егер $\boldsymbol{N}$ құрамында n бейтерминалдар болса, $\boldsymbol{N}_e\subseteq\boldsymbol{N}$ болғандықтан III.1.2.1-алгоритмі ең көбі (2) қадамды n+1 рет қайталағаннан кейін тоқтауы керек.

III.1.2.2-анықтама. Егер $X\in\boldsymbol{N}\cup\boldsymbol{T}$ символы ешқандай да шығарым тізбесіде кездеспесе, X символын $\boldsymbol{G}$ = <$\boldsymbol{N}$, $\boldsymbol{T}$, $\boldsymbol{P}$, S> КБГ-нда *қол жетпейтін* символ деп атаймыз.

КБГ-нан қол жетпейтін символдарды келесі алгоритм көмегімен жоюға болады.

III.1.2.2-алгоритм. Қол жетпейтін символдарды жою.

Кіру. КБГ $\boldsymbol{G}$=<$\boldsymbol{N}$, $\boldsymbol{T}$, $\boldsymbol{P}$, S>.

Шығу. КБГ $\boldsymbol{G}'$ =<$\boldsymbol{N}'$, $\boldsymbol{T}'$, $\boldsymbol{P}'$, S >,

(i) $\boldsymbol{L}(\boldsymbol{G}')=\boldsymbol{L}(\boldsymbol{G})$,

(ii) барлық $X \in \boldsymbol{N'} \cup \boldsymbol{T'}$ үшін $S \Rightarrow^{*}_{G} \alpha X \beta$ орындалатындай $(\boldsymbol{N'} \cup \boldsymbol{T'})^*$ жиынының α және β тізбелері табылады.

Әдіс.

(1) $V_0 = \{S\}$ және $i = 1$ деп алу.

(2) $V_i = \{X$: $\boldsymbol{P}$ құрамында $A \rightarrow \alpha A \beta$ және $A \in V_{i-1}$ бар$\} \cup V_{i-1}$.

Егер $V_i \neq V_{i-1}$, онда $i=i+1$ деп алып (2) қадамға көшу. Кері жағдайда грамматика $\boldsymbol{G'} = <\boldsymbol{N'}, \boldsymbol{T'}, \boldsymbol{P'}, S>$, $\boldsymbol{N'} = V_i \cap \boldsymbol{N}$, $\boldsymbol{T'} = V_i \cap \boldsymbol{T}$, $\boldsymbol{P'}$ құрамында тек V_i символдары болатын $\boldsymbol{P}$ жиынының ережелерінен тұрады.

Енді КБГ-нан барлық пайдасыз символдарды жоюға дайын болдық.

III.1.2.3-алгоритм. Пайдасыз символдарды жою.

Кіру. КБ-грамматика $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ және $\boldsymbol{L}(\boldsymbol{G}) \neq \varnothing$.

Шығу. КБ-грамматика $\boldsymbol{G'}=<\boldsymbol{N'}, \boldsymbol{T'}, P', S>$ және $\boldsymbol{L}(\boldsymbol{G'})=\boldsymbol{L}(\boldsymbol{G})$, $\boldsymbol{N'} \cup \boldsymbol{T'}$ жиынында пайдасыз символдар жоқ.

Әдіс.

(1) $\boldsymbol{G}$-ге III.1.2.1 алгоритмін қолданып $\boldsymbol{N}_e$ жиынын алу. $\boldsymbol{G} = <\boldsymbol{N} \cap \boldsymbol{N}_e, \boldsymbol{T}, \boldsymbol{P}_1, S>$ деп алу, мұндағы, $\boldsymbol{P}_1$ жиыны құрамында тек $\boldsymbol{N}_e \cup \boldsymbol{T}$ жиынының символдары ғана болатын $\boldsymbol{P}$ жиынының ережелерінен тұрады.

(2) $\boldsymbol{G}_1$-ге III.1.2.2 алгоритмін қолданып, $\boldsymbol{G'}=<\boldsymbol{N'}, \boldsymbol{T'}, \boldsymbol{P'}, S>$ алу.

III.1.2.3 алгоритмінің (1) қадамында $\boldsymbol{G}$-дан терминалды тізбелер туғыза алмайтын барлық бейтерминал символдар жойылады. Сонан соң (2) қадамда барлық қол жетпейтін символдар жойылады. Нәтижедегі грамматикадағы әрбір X символы $S \Rightarrow^* \omega X y \Rightarrow^* \omega x y$ түріндегі шығарымдардың ең болмағанда бірінде кездесуі шарт. Әуелі III.1.2.2-алгоритмін, сонан соң III.1.2.1-алгоритмін қолдансақ, әр уақытта да құрамында пайдасыз символдар болмайтын грамматика алуға болмайтынын байқаймыз.

III.1.2.1-мысал. $\boldsymbol{G} = <\{S, A, B\}, \{a, b\}, \boldsymbol{P}, S>$ грамматикасын қарастырамыз, мұнда $\boldsymbol{P} = \{S \rightarrow a|A, A \rightarrow AB, B \rightarrow b\}$.

$\boldsymbol{G}$-ге III.1.2.3-алгоритмін қолданамыз. (1) қадамда $\boldsymbol{N}_e= \{S, B\}$, $\boldsymbol{G}_1=<\{S,B\}, \{a,b\}, \{S\to a, B\to b, S>$ аламыз. III.1.2.2-алгоритмін қолданып, $V_2=V_1=\{S, a\}$ аламыз. Демек, $\boldsymbol{G'}=<\{S\}, \{a\}, \{S\to a\}, S>$.

Егер $\boldsymbol{G}$-ға алдымен III.1.2.2-алгоритмін қолдансақ, онда барлық символдар қол жететін болады, бұл жағдайда грамматика өзгермейді. Сонан соң III.1.2.1-алгоритмін қолдану $\boldsymbol{N}_e=\{S, B\}$ береді және нәтижесінде $\boldsymbol{G'}$ грамматикасына ұқсамайтын $\boldsymbol{G}_1$ алынады.

Көбінесе КБГ-нан ε-ережені, яғни $A\to$e түріндегі ережені жою ыңғайлы. Дегенмен, e$\in\boldsymbol{L}(\boldsymbol{G})$ болған жағдайда $A\to$e ережесінсіз болу мүмкін емес екендігі анық.

III.1.2.3-анықтама. $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ КБГ-н ε-ережесінсіз грамматика (немесе қысқармаған) деп атаймыз, егер

(1) $\boldsymbol{P}$ құрамында ε-ереже болмайды, немесе

(2) Нақты бір ғана ε-ереже $S\to\varepsilon$ болады және $\boldsymbol{P}$ жиынының басқа ережелерінің оң жағында S кездеспейді.

III.1.2.4-алгоритм. ε-ережесіз грамматикадағы түрлендіру.

Кіріс. КБГ $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$.

Шығыс. $\boldsymbol{G} = <\boldsymbol{N'}, \boldsymbol{T}, \boldsymbol{P'}, S'>$ эквивалентті ε-ережесіз КБ – грамматика.

Әдіс.

$\boldsymbol{N}_e= \{A : A\in\boldsymbol{N}\ A\Rightarrow^*_{\boldsymbol{G}}\text{e}\}$ деп аламыз.

(1) $\boldsymbol{P'}$ жиынын былай құрамыз:

(а) Егер $A\to\alpha_0B_1\alpha_1B_2\alpha_2\ \ldots\ B_k\alpha_k$, $k\geq0$, $B_i\in\boldsymbol{N}_e$, $1\leq i\leq k$ ережесі $\boldsymbol{P}$ жиынында жатса, сондай-ақ, α_j $(0\leq j\leq k)$ тізбесінің бір де бір символы $\boldsymbol{N}_e$ жиыны құрамында болмаса, онда келесі түрдегі барлық ережелерді $\boldsymbol{P'}$ құрамына енгізу

$A\to\alpha_0X_1\alpha_1X_2\ \ldots\ \alpha_{k-1}X_k\alpha_k$

Мұндағы, X_i дегеніміз – B немесе e. $A\to$e ережесі енгізілмейді (бұл жағдай барлық α_i символдары e-ге тең болған жағдайда болуы мүмкін).

(б) Егер $S\in\boldsymbol{N}_e$, онда $\boldsymbol{P'}$ құрамына келесі ережені енгізу

$S' \to \varepsilon \mid S$, мұндағы S'– жаңа символ, және $\boldsymbol{N}'=\boldsymbol{N} \cup \{S'\}$ деп алу. Кері жағдайда $\boldsymbol{N}'= \boldsymbol{N}$ және $S'= S$ деп алу.

(3) $\boldsymbol{G}'= <\boldsymbol{N}', \boldsymbol{T}, \boldsymbol{P}', S'>$ деп аламыз.

III.1.2.2-мысал. Ережелері төмендегідей болатын III.1.2.13 мысалдағы грамматиканы қарастырамыз

$$S \to aSbS \mid bSaS \mid \varepsilon$$

Осы грамматикаға III.1.2.4 алгоритмін қолданып, келесі грамматиканы аламыз

$$S' \to S \mid \varepsilon$$
$$S \to aSbS \mid bSaS \mid aSb \mid abS \mid ab \mid bSa \mid baS \mid ba$$

III.1.2.1-теорема. III.1.2.4-алгоритмі бастапқы грамматикаға эквивалентті ε-ережесінсіз грамматиканы береді.

Грамматиканы түрлендірудің тағы бір түрі – тізбекті деп аталатын $A \to B$ түріндегі ережені жою.

III.1.2.5-алгоритм. Тізбекті ережелерді жою.

Кіру. Е-ережесінсіз $\boldsymbol{G}$ КБГ.

Шығу. $\boldsymbol{G}$ грамматикасына эквивалентті, ε-ережесінсіз және тізбекті ережелері жоқ $\boldsymbol{G}'$ КБГ.

Әдіс.

(1) Әрбір $A \in \boldsymbol{N}$ үшін келесі жолмен $\boldsymbol{N}_A=\{B: A \Rightarrow^* B\}$ құру:

(а) $\boldsymbol{N}_0 =\{A\}$ және i=1 деп алу.

(б) $\boldsymbol{N}_i= \{C: B \to C$ $\boldsymbol{P}$-да жатады және $B \in \boldsymbol{N}_{i-1}\} \cup \boldsymbol{N}_{i-1}$.

(в) Егер $\boldsymbol{N}_i \neq \boldsymbol{N}_{i-1}$ болса, $i=i+1$ деп алып (б) қадамын қайталау. Кері жағдайда $\boldsymbol{N}_A= \boldsymbol{N}_i$ деп алу.

(2) $\boldsymbol{P}'$ жиынын былай құру: егер $B \to \alpha$ ережесі $\boldsymbol{P}$-да жатса және тізбекті ереже болмаса, $\boldsymbol{P}'$ құрамына барлық A, $B \in \boldsymbol{N}_A$ үшін $A \to \alpha$ ережесін енгізу.

(3) $\boldsymbol{G}'=<\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}', S>$ алу.

III.1.2.3-мысал. Ережелері төмендегідей болатын $\boldsymbol{G}_0$ грамматикасына III.1.2.5 алгоритмін қолданамыз

$$E \to E+H \mid H,\ H \to H*F \mid F,\ F \to (E) \mid$$

(1) қадамда $N_E = \{E, H, F\}$, $N_T = \{H, F\}$, $N_F = \{F\}$. (2) қадамнан кейін P' жиыны мынадай түрге келеді:

$P' = \{E \to E+H|H*F|(E)|a, H \to H*F\ |(E)|a, F \to (E)|a\}$.

III.1.2.2-теорема. III.1.2.5 алгоритмі тұрғызатын G' грамматикасында тізбек ережелері болмайды және $L(G)= L(G')$.

Дәлелелдеу. III.1.2.5-алгоритмі тізбекті ережелері жоқ G' грамматикасын беретіні белгілі. Әуелі $L(G')\subseteq L(G)$ көрсетеміз. $\omega \in L(G')$ болсын. Онда G' грамматкасында $S \Rightarrow \alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_n = \omega$ шығарымы табылады. Егер α_i тізбесінен α_{i+1} тізбесіне ауысарда $A \to \beta$ ережесі қолданылса, онда $A \Rightarrow^*_G B$ және $B \Rightarrow_G \beta$ орындалатындай $B \in N$ символы ($B=A$ болуы мүмкін) табылады Сонымен, $A \Rightarrow^*_G \beta$ және $\alpha_i \Rightarrow^*_G \alpha_{i+1}$. Бұдан $S \Rightarrow^*_G \omega$ және $\omega \in L(G)$ шығады, яғни $L(G')\subseteq L(G)$.

Енді $L(G)\subseteq L(G')$ екендігін көрсетеміз. Мейлі $\omega \in L(G)$ және $S=\alpha_0 \Rightarrow_l \alpha_1 \Rightarrow_l \ldots \Rightarrow_l \alpha_n = \omega$ – G грамматикасындағы ω тізбесінің сол жақ шығарымы болсын. j-лердің ішіндегі $\alpha_{j-1} \Rightarrow_l \alpha_j$ қадамында тізбекті емес ережелер қолданатындардан ғана тұратын i_1, i_2, ..., i_k индекстерінің іштізбегін табуға болады. Терминалды тізбектің шығарымы тізбекті ережемен аяқталмайтындықтан, дербес жағдайда $i_k=n$ болады.

Біз сол жақ шығарымды қарастырғандықтан, тізбекті ережені қолдану шығарымның сәйкес бөлігі қамтитын сол жақ шығарымды тізбеде тек бір ғана орынды алатын символды алмастырады. Бұдан шығатыны, $S \Rightarrow_{G'} \alpha_{i1} \Rightarrow_{G'} \alpha_{i2} \Rightarrow_{G'} \ldots \Rightarrow_{G'} \alpha_{ik} = \omega$. Сонымен, $\omega \in L(G')$, яғни $L(G')= L(G)$.

***III.1.2.4-анықтама*:** Мейлі КБГ $G = \langle N, T, P, S \rangle$ берілсін. Сонда:

1. Егер G грамматикасында бейтерминал $A \in N$ үшін $A \Rightarrow \zeta A \xi$, $\zeta \in (T \cup N)^*$, $\xi \in (T \cup N)^*$ шығарымы болса, онда ол *қайталанымды символ* деп аталады.

2. Егер G грамматикасында қайталанымды символ $A \in N$ үшін $|\zeta A \xi|>1$ орындалса, онда ол *тиімді қайталанымды символ* деп аталады.

3. Егер $\boldsymbol{G}$ грамматикасында ең болмаса бір қайталанымды символ болса, онда ол *қайталанымды грамматика* деп аталады.

4. Егер $\boldsymbol{G}$ грамматикасында бейтерминал $A \in \boldsymbol{N}$ үшін $A \Rightarrow^{+} A$ шығарымдары болмаса, онда ол *қайталанымсыз грамматика* деп аталады.

5. Егер $\boldsymbol{G}$ грамматикасы қайталымсыз, ε-ережесінсіз және пайдасыз символдарсыз болса, онда ол келтірілген грамматика деп аталады.

Кейде ε-ережелі немес қайталанымды грамматикаларды талдау ε-ережесіз грамматикаларға қарағанада қиындық тудырады, себебі практикалық жағдаятта пайдасыз символдар қажетінсіз талдауыштың (анализатордың) көлемін үлкейтіп жібереді. Сондықтан да синтаксистік талдаудың кейбір алгоритмдерінде қолданылатын грамматикалардың келтірілген болғаны тиімді. Бұл талап барлық КБ тілдерді қайталанымсыз деп қарастыруға мүмкіндік береді.

III.1.2.3-теорема. Егер $\boldsymbol{L}$– КБ-тіл болса, онда келтірілген $\boldsymbol{G}$ КБ – грамматикасы үшін $\boldsymbol{L}=\boldsymbol{L}(\boldsymbol{G})$.

Дәлелдеу. $\boldsymbol{L}$ тілін анықтайтын КБ-грамматикаға III.1.2.2–III.1.2.5-алгоритмдерін қолдану керек.

III.1.2.5-анықтама. КБГ-ндағы $A \rightarrow \alpha$ түріндегі ереже A – ереже деп аталады (A – ережені $B \rightarrow$e түріндегі ε-ережемен шатыстырмау керек).

Грамматикадан $A \rightarrow \alpha B \beta$ түріндегі ережені жоятын тағы бір түрлендіру енгізейік. Бұл ережені жою үшін осы грамматикаға барлық B-ережелердің оң жағындағы B бейтерминал символдарына алмастырулар енгізу арқылы пайда болған жаңа ережелерді қосу керек.

III.1.2-лемма. $\boldsymbol{G}= <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ – КБ- грамматика және $\boldsymbol{P}$ құрамында $A \rightarrow \alpha B \beta$, $B \in \boldsymbol{N}$ ережесі бар болсын, ал α мен β $(\boldsymbol{N} \cup \boldsymbol{T})^*$ жиынында жатсын. $B \rightarrow \gamma_1 | \gamma_2 | \ldots | \gamma_k |$ – осы грамматиканың барлық B – ережелері болсын. Грамматика $\boldsymbol{G'}= <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$, мұнда

$\boldsymbol{P'} = <\boldsymbol{P} \setminus \{A \to \alpha B\beta\}) \cup \{A \to \alpha\gamma_1\beta | \alpha\gamma_2\beta | \ldots | \alpha\gamma_k\beta|$. Сонда $\boldsymbol{L}(\boldsymbol{G}) = \boldsymbol{L}(\boldsymbol{G'})$.

Бұл тұжырымың дәлелдеуін өздеріңіз тапсырма ретінде келтіріңіз.

III.1.2.4-мысал. $A \to aAA|b$ түріндегі екі ережесі бар $\boldsymbol{G}$ грамматикасынан $A \to aAA$ ережесін жоялық. $\alpha = a$ және $\beta = A$ деп алып, III.1.2.1-леммасын қолданамыз, ережелері төмендегідей болатын $\boldsymbol{G'}$ грамматикасын $A \to aAAA|abA|b$ аламыз.

$\boldsymbol{G}$ және $\boldsymbol{G'}$ грамматикаларындағы $aabbb$ тізбесінің шығарым дарақтары III.1.2.6-суретте берілген.

III.1.2-тапсырмалар:

1. Мына ережелері $S \to aABS|bCACd$, $A \to bAB|cSA|cCC$, $B \to bAB|cSB$, $C \to cS|c$ бар грамматикаға эквивалентті келтірелген грамматика тұрғызыңыз.
2. Мына ережелері $S \to aAB|E$, $A \to dDA|\varepsilon$, $B \to bE|f$, $C \to cAB|dSD|a$, $D \to eA$, $E \to fA|g$ бар грамматикаға эквивалентті келтірелген грамматика тұрғызыңыз.
3. Ең болмаса бір тиімді қайталымдық символы бар келтірілген қайталымды КБГ-мен туындаған тіл ақырсыз екендігін дәлелдеңіз.
4. Қайталанымсыз КБГ-мен туындаған тіл ақырлы екендігін дәлелдеңіз.
5. Әріптер мен цифрлардан тұратын айқындағыштарды тудыратын келтірілген грамматиканы құрыңыз.

III.1.2-сұрақтар:

1. Қайталанымсыз КБГ ақырлы тілді тудыра ма?
2. Жалғыз ғана тиімді қайталымдық символды қамтитын қайталымды келтірілген КБГ ақырсыз тілді тудыра ма?
6. Мына ережелері $A \to AA|\alpha$, $A \to A\alpha A|\beta$, $A \to \alpha A|A\beta|\gamma$ бар бір мәнді емес грамматиканы бір мәнді грамматикаға түрлендіруге бола ма?
7. ε-ережесінсіз грамматика ақырсыз тілді тудыра ма?
8. Грамматикадағы пайдасыз символдардан қалай құтылуға болады?

III.1.3. Хомскийдің нормалды пішімі

Бұл параграфта контекстісіз грамматикада Хомскийдің нормалды пішіміні анықталады, осы пішімге түрлендіретін алгоритм сипатталады, сонымен қатар тапсырмалар берілелі және сұрақтар қойылады [1-9,11-13,15-22,24,25, 27-32].

III.1.3.1-анықтама. КБГ $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S\rangle$ *Хомскийдің нормалды пішімінде* (немесе *бинарлы нормалды пішімде*) болады, егер ***P***-дағы әрбір ереже келесі түрдің біреуінде болса:

(1) $A \to BC$, мұндағы $A, B, C \in \boldsymbol{N}$,

(2) $A \to a$, мұндағы $a \in \boldsymbol{T}$,

(3) егер $\varepsilon \in \boldsymbol{L}(\boldsymbol{G})$, онда $S \to \varepsilon$, сондай-ақ S ережелердің оң жақ бөлігінде кездеспейді.

Әрбір КБ-тілі Хомскийдің нормалды пішіміндегі грамматикадан туындайтынын көрсетеміз.

III.1.3.1-алгоритм. Хомскийдің нормалды пішіміне түрлендіру.

Кіру. Келтірілген КБ – грамматика $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S\rangle$.

Шығу. ***G***-ға эквивалентті Хомскийдің нормалды пішіміндегі $\boldsymbol{G'}$ КБ – грамматикасы, яғни $\boldsymbol{L}(\boldsymbol{G'}) = \boldsymbol{L}(\boldsymbol{G})$.

Әдіс. ***G*** бойынша $\boldsymbol{G'}$ грамматикасы келесі жолмен құралады:

(1) $\boldsymbol{P'}$-ға ***P***-ның $A \to$а түріндегі әрбір ережесін енгізу.

(2) $\boldsymbol{P'}$-ға ***P***-ның $A \to BC$ түріндегі әрбір ережесін енгізу.

(3) $\boldsymbol{P'}$-ға $S \to \varepsilon$ ережесін енгізу, егер ол ***P***-да болса.

(4) $A \to X_1 \ldots X_k$, $k>2$ түріндегі ***P*** жиынының әрбір ережесі үшін $\boldsymbol{P'}$ жиынына келесі ережелерді қосу:

$$A \to X'_1\langle X_2 \ldots X_k\rangle,$$
$$\langle X_2 \ldots X_k\rangle \to X'_2\langle X_3 \ldots X_k\rangle,$$
$$\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$$
$$\langle X_{k-2}X_{k-1} \ldots X_k\rangle \to X'_{k-2}\langle X_{k-1}X_k\rangle,$$
$$\langle X_{k-1}X_k\rangle \to X'_{k-1}X'_k,$$

мұнда егер $X_i \in \boldsymbol{N}$, онда $X'_i = X_i$; егер $X_i \in \boldsymbol{T}$, онда X'_i – бейтерминал жаңа символ; $\langle X_i \ldots X_k\rangle$ - бейтерминал жаңа символ.

(5) Әрбір $A \to x_1x_2$ түріндегі ереже үшін, мұнда $x_1 \in \boldsymbol{T}$ және $x_2 \in \boldsymbol{T}$, $\boldsymbol{P}'$-ға $A \to x'_1x'_2$ ережесін енгізу.

(6) Қадам (4) және (5)-те енгізілген a' түріндегі әрбір бейтерминал символы үшін $\boldsymbol{P}'$ жиынына $a' \to a$ ережесін енгізу. Сонымен, $\boldsymbol{N}'$ - $\boldsymbol{P}'$ жиынын тұрғызғанда енгізілген барлық бейтерминал жаңа символдарымен қоса $\boldsymbol{N}$ жиынынан тұрады. Сонда ізделіп отырған грамматика $\boldsymbol{G} = <\boldsymbol{N}', \boldsymbol{T}, \boldsymbol{P}', S>$ болады.

III.1.3.1-теорема. $\boldsymbol{L}$ – КБ-тіл болсын. Онда Хомскийдің нормалды пішіміндегі $\boldsymbol{G}'$ КБГ үшін $\boldsymbol{L} = \boldsymbol{L}(\boldsymbol{G}')$ орындалады.

Дәлелдеу. III.1.2.3-теорема бойынша $\boldsymbol{L}$ тілі $\boldsymbol{G}$ келтірілген грамматикасымен анықталады. III.1.2.1 алгоритмі Хомскийдің нормалды пішімі бар $\boldsymbol{G}'$ грамматикасын құрайды. Енді $\boldsymbol{L}(\boldsymbol{G})=\boldsymbol{L}(\boldsymbol{G}')$ екендігін көрсету қалды. Бұл теңдік $\boldsymbol{G}'$ грамматикасындағы оң жақ құрамында а' болатын әрбір ережеге III.1.2.1-леммамен.

III.1.3.1-мысал. Келтірілген КБГ-ның ережелері $S \to aAB|BA$, $A \to BBB|a$, $B \to AS|b$. III.1.2.1-алгоритммен келесі ережелерді сақтап $\boldsymbol{P}'$ құрамыз: $S \to BA$, $A \to a$, $B \to AS$ және $B \to b$, $S \to aAB$ ережесін $S \to a'<AB>$ және $<AB> \to AB$ ережелерімен, ал $A \to BBB$ ережесін $A \to B<BB>$ және $<BB> \to BB$ ережелерімен алмастырамыз. Соңында $a' \to a$ ережесін қосамыз. Нәтижесінде $\boldsymbol{G} = <\boldsymbol{N}', \{a,b\}, \boldsymbol{P}', S>$ болады, мұндағы $\boldsymbol{N}' = \{S, A, B, <AB>, <BB>, a'\}$, ал $\boldsymbol{P}'$ ережелері: $S \to a'<AB>|BA$, $A \to B<BB>|a$, $B \to AS|b$, $<AB> \to AB$, $<BB> \to BB$, $a' \to a$.

III.1.3-тапсырма. Келесі келесі $\boldsymbol{G} = <\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{P}, S>$ КБГ-н Хомскийдің нормалды пішіміне түрлендіріңіз:

1. $\boldsymbol{T} = \{a, b\}$, $\boldsymbol{N} = \{S\}$, $\boldsymbol{P} = \{S \to ab, S \to aSb, S \to SS\}$;
2. $\boldsymbol{T} = \{x, a, p\}$, $\boldsymbol{N} = \{S, D\}$, $\boldsymbol{P}, \{A\}>$, $\boldsymbol{P} = \{S \to x|Dx, D \to Da|p\}$;
3. $\boldsymbol{T} = \{a, b\}$, $\boldsymbol{N} = \{A, B, S\}$, $\boldsymbol{P} = \{S \to Aa|a|b, A \to Ab|a\ |aB, B \to B\mathrm{a}|b\}$.

III.1.3-сұрақтар:

1. Кез келген КБГ үшін Хомскийдің нормалды пішімі бар ма?
2. КБГ-ның келтірілген пішімі қалай тұрғызылады?
3. III.1.3.1 алгоритмінің кірісі не болады?

III.1.4. Грейбахтың нормалды пішімі

Бұл параграфта контекстісіз грамматикада Грейбахтың нормалды пішіміні анықталады, осы пішімге түрлендіретін алгоритм сипатталады, сонымен қатар тапсырмалар берілелі және сұрақтар қойылады [1-9,11-13,15-22,24,25].

III.1.4.1-анықтама. Қандай да бір α және β үшін $A \Rightarrow^* \alpha A \beta$ орындалса, $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ КБГ-ның A бейтерминал символы рекурсивті деп аталады. Егер α=ε, онда A сол жақты рекурсивті деп аталады. Дәл осылай, егер β=ε, онда A оң жақты рекурсивті деп аталады. Құрамында ең болмағанда бір сол жақты рекурсивті бейтерминал символ болатын грамматика сол жақты рекурсивті грамматика деп аталады. Оң жақты рекурсивті грамматика дәл осылай анықталады. Бастапқы символдан басқа бейтерминал символдары рекурсивті болатын грамматика рекурсивті грамматика деп аталады.

Кейбір алгоритмдер сол жақты рекурсивті грамматикалармен жұмыс істей алмайды. Әрбір КБ-тілі ең болмағанда бір сол жақты рекурсивті емес грамматикамен анықталатындығын көрсетеміз. КБГ-дан сол жақты рекурсивтілікті тікелей жоюдан бастаймыз.

III.1.4.1-мысал. КБГ мына ережелермен анықталсын

$$E \to E+H|H,\ H \to H * F|F,\ F \to (E)|a$$

Егер оған III.1.2.9-леммасының құрастырымын қолдансақ, оған эквивалентті $\boldsymbol{G'}$ грамматиканың ережелері келісі

$$E \to H|HE',\ E' \to +\,H|+\,HE',\ H \to F|FH',\ H' \to *\,F|FH',\ F \to (E)|a$$

III.1.4.1-алгоритм. Сол жақты рекурсияны жою.

Кіру. Келтірілген КБ – грамматика $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$.

Шығу. Сол жақты рекурсиясы жоқ эквивалентті КС–грамматика.

Әдіс.

(1) $\boldsymbol{N}=\{A_1,...,A_n\}$ болсын. $A_i \to \alpha$ ережесенде α тізбесі терминал символдан немесе A_j, j>i символынан басталатындай етіп $\boldsymbol{G}$ грамматикасын түрлендіреміз. Осындай мақсатпен i=1 деп аламыз.

(2) $A_i \to A_i\alpha_1|...|A_i\alpha_m|\beta_1|...|\beta_p$ - A_i – ережелер жиыны болсын, мұнда $k \le i$ болғанда β_j тізбелерінің ешқайсысы A_k символынан басталмайды. A_i – ережелерін келесі ережелермен алмастырамыз

$A_i \to \beta_1|...|\ \beta_p|\beta_1\ A'_i|...|\ \beta_p A'_i$

$A'_i \to \alpha_1|...|\ \alpha_m|\alpha_1 A'_i|...|\ \alpha_m A'_i$

A'_i – бейтерминал жаңа символ. Енді барлық A_i – ережелерінің оң жағы терминалдан немесе A_k, k>i символынан басталады.

(3) Егер $i=n$, онда алынған $\boldsymbol{G'}$ грамматикасын нәтиже деп есептеп, тоқтау. Кері жағдайда $i= i+1$ және j=1 деп алу.

(4) A_j символы үшін әрбір $A_i \to A_j\alpha$ түріндегі ережені, мұнда $\alpha \in (\boldsymbol{T} \cup \boldsymbol{N})^*$, $A_i \to \beta_1\ \alpha|\ ...|\ \beta_m\ \alpha$ ережелерімен ауыстырамыз, ал $A_j \to \beta_1|...|\ \beta_m$ - барлығы A_j үшін ережелер. Себебі, әрбір $A_j \to \beta_1|...|\ \beta_m$ ережесінің оң жағы терминалдан немесе $k>j$ үшін A_k-дан басталғандықтан, оның оң жағы да осындай қасиетке ие болады.

(5) Егер $j=i-1$, (2) қадамға көшу. Кері жағдайда j= j+1 деп алып, (4) қадамға көшу.

III.1.4.1-теорема. Әрбір КБ-тіл сол жақты рекурсивті емес грамматикамен анықталады.

III.1.4.2-мысал. Мейлі $\boldsymbol{G}$ келесі ережелермен анықталсын

$$A \to BC|a,\ B \to CA|Ab,\ C \to AB|CC|a$$

$A_1=A$, $A_2=B$ және $A_3=C$ деп аламыз. III.1.4.1 алгоритміндегі (2) немесе (4) қадамдарын қолданған сайын мынадай грамматикалар пайда болады (тек жаңа ережелерді ғана көрсетеміз):

(2) қадам $i=1$ үшін: $\boldsymbol{G}$ өзгермейді

(4) қадам $i=2$, j=1 үшін: $B \to CA|BCb|ab$

(2) қадам $i=2$ үшін: $B \to CA|ab|CAB'\ |abB'$, $B' \to CbB'\ |Cb$

(4) қадам $i=3$, j=1 үшін: $C \to BCB|aB|CC|a$

(4) қадам $i=3$, j=2 үшін:

$$C \to CACB|abCB|CAB'CB\ |abB'CB|aB|CC|a$$

(2) қадам $i=3$ үшін:

$$C \to abCB|abB'CB\ |aB\ |a|abCBC'|abB'CBC'|aBC'|aC'$$

$$C' \to ACBC'|AB'CBC'|CC'|ACB|AB'CB|C$$

Сол жақты рекурсивті емес грамматиканың дербес жағдайы – Грейбахтың нормалды пішіміндегі грамматика.

III.1.4.2-анықтама. $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ КБ-грамматикасы *Грейбахтың* нормалды пішімінде деп аталады, егер онда ε-ереже болмаса және $\boldsymbol{P}$ жиынындағы $S \to \varepsilon$ бөлек әрбір ереже $A \to a\alpha$ түрінде болса, мұнда $A \in \boldsymbol{N}$, $a \in \boldsymbol{T}$ және $\alpha \in \boldsymbol{N}^*$.

Егер грамматика сол жақты рекурсивті болмаса, онда оның бейтерминал символдар жиынында табиғи аракідік ретті анықтауға болады. Осы жатылай ретті сызықты ретке кіргізуге болады. Бұл грамматиканы Грейбахтың нормалды пішіміне түрлендіру кезінде пайдалы болады.

III.1.4.1-лемма. $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ – сол жақты рекурсивті емес грамматика болсын. $A \to B\alpha$ ережесі $\boldsymbol{P}$ жиынында жатса $A < B$ орындалатындай $\boldsymbol{N}$ жиынынан $<$ сызықты реті табылады.

Дәлелдеу. R – $\boldsymbol{N}$ жиынындағы "$ARB \Leftrightarrow$ қандай да α үшін $A \Rightarrow^{+} B\alpha$ орындалады" деген қатынас болсын. $\boldsymbol{G}$ грамматикасы сол жақты рекурсивті болмағандықтан, R – дербес реттеу болады. R қатынасын сызықты $<$ реттеуіне дейін кеңейтуге болады.

III.1.4.2-алгоритм. Грейбахтың нормалды пішіміне түрлендіру.

Кіру. Сол жақты рекурсивті емес келтірілген $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ КБГ.

Шығу. Грейбахтың нормалды пішіміндегі $\boldsymbol{G}'$ эквивалентті грамматикасы.

Әдіс. Алты түрлі кіріс бар.

(1) III.1.2.10-лемма көмегімен $\boldsymbol{N}$ жиынында келесі шарт орындалатындай $<$ сызықты реттеу тұрғызу: әрбір A-ереже терминалдан немесе бейтерминал B, $A < B$ символынан басталады. $A_1 < A_2 < \ldots < A_n$ орындалатындай етіп $\boldsymbol{N} = \{A_1, \ldots, A_n\}$ реттеу.

(2) $i = n$-1 деп алу.

(3) Егер i=0, онда (5) қадамға көшу. Басқа жағдайда әрбір $A_i \to A_j\alpha$ мұнда $j > i$, түріндегі ережені $A_i \to \beta_1\alpha | \ldots | \beta_m\alpha$ ережелерімен

алмастырамыз, мұнда $A_j \to \beta_1|...|\beta_m$ - барлығы A_j үшін ережелер. Кейін әрбір $\beta_1, \ldots, \beta_m$, тізбе терминалмен басталатынына көз жеткіземіз.

(4) $i = i$-1 деп алып, (3) қадамға оралу.

(5) Қазір әрбір ереженің оң жағы ($S \to$e ережесінен басқа) терминалдан басталады. Әрбір $A \to aX_1 \ \ldots \ X_k$ ережесіндегі $X_j \in \boldsymbol{T}$ символын жаңа X'_j бейтерминал символмен ауыстыру.

(6) (5) қадамда енгізілген жаңа X'_j бейтерминал символдары үшін $X'_j \to X_j$ ережелерін қосу.

III.1.4.2-теорема. Егер $\boldsymbol{L}$ – КСА-тіл, онда Грейбахтың нормалды пішіміндегі $\boldsymbol{G}$ грамматикасы үшін $\boldsymbol{L}=\boldsymbol{L}(\boldsymbol{G})$ орындалады.

Дәлелдеу. $n-i$ бойынша (i бойынша, бірақ $i=n$-1–ден басталатын және i=1-ге аяқталатын кері бағытта) индукция арқылы III.1.4.2-алгоритмнің (3)-қадамын орындағаннан кейін i үшін әрбір A_i-ережесінің оң жағы терминалмен басталатынын көрсетуге болады. Мұның кілті – сызықтық рет < қолдану. (5) қадамнан кейін грамматика Грейбахтың нормалды пішіміне түрленеді және III.1.4.1 лемма бойынша ол тудыратын тіл өзгермейді.

III.1.4.3-мысал. $\boldsymbol{G}$ грамматикасының ережелері:

$$E \to H|HE',\ E' \to +H|+HE',\ H \to F|FH',\ H' \to *F|*FH',\ F \to (E)|a$$

Бейтерминал символдарды былай реттейміз: $E'<E<H'<H<F$.

Бұл реттеуде F – ең үлкен бейтерминал символ болғандықтан әрбір F – ережесінің оң жағы терминалдан басталады. Оның алдындағы H бейтерминал символының ережесі $H \to F|FH'$ түрінде болады, мұндағы F-ті алмастырып, $H \to (E)|a(E)H'|aH'$ ережесін аламыз. H' символына көше отырып мұнда ештемені өзгертудің қажеті жоқ екнін байқаймыз. Сонан соң E – ережені келесі ережелермен алмастырамыз

$$E \to (E)|a|(E)H'|aH'|(E)E'|aE'|(E)H'E'|aH'E'BE'$$

ережелерінде ешқандай өзгеріс жасудың қажеті жоқ.

(5) және (6) қадамдарда бейтерминал)′ жаңа символы және)′→) ережесі пайда болады, сондықтан алдыңғы ережелердегі барлық)

символын)′ символына ауыстыру қажет. Сонымен, нәтижесінде ережелері төмендегідей болатын Грейбахтың нормалды пішіміндегі грамматика шығады:

$$E\rightarrow(E)|a|(E)H'|aH'|(E)E'|aE'|(E)H'E'|aH'E',\ E'\rightarrow+H|+HE',$$
$$H\rightarrow(E)'|a|(E)'H'|aH',\ H'\rightarrow*F|*FH',\ F\rightarrow(E)'|a,\ )'\rightarrow).$$

III.1.4-тапсырмалар:

1. Мынадай ережелері бар:

$$S\rightarrow aAB,\ A\rightarrow aA|bB,\ B\rightarrow ACb|b,\ C\rightarrow bA|Cc.$$

КБ-грамматиканы Грейбахтың нормалды пішіміне түрлендіріңіз.

2. Ережелері мынадай $S\rightarrow SabA|ba$, $A\rightarrow AbA|bAa|c$ сол рекурсивті грамматикасы үшін эквивалентті оң рекурсивті грамматика табу.

3. Оң рекурсиядан құтылдыратын алгоритм тұрғызыңыз және осындай түрлендірудің эквивалентті екенін дәлелдеу керек?

4. Мына тізбелерді ғана шығаратын КБ-грамматиканы тұрғызу:

(а) 00..0011..11 (нөлдер саны бірліктер санына тең);

(б) 00..0011..11 (нөлдер саны бірліктер санынан екі есе көп);

(в) 00..0011..11 (нөлдер саны бірліктер санынан көп).

5. Мына жиынды тудыратын $\{a_1a_2...a_na_n...a_2a_1|a_i\in\{0,1\},1\leq i\leq n\}$ КБ-грамматикасын тұрғызыңыз.

III.1.4-сұрақтар:

1. Құрамында нөлдер, бірліктер, екіліктер саны тең мына түрдегі 00..0011..1122..22 тізбені ғана тудыратын КБГ бар ма?

2. Мына ережелері бар КБГ

$$S\rightarrow Aab|bSb|abB,\ A\rightarrow abS|ba|bbA,\ B\rightarrow bB|Da|Aa,\ D\rightarrow Dbb|aDa$$

оң рекурсивті немесе сол рекурсивті бола ма?

3. Мына $\{a^nb^nc^n : n>1\}$ тілді тудыратын контекстісіз тұрғызуға бола ма?

4. Контекстісіз грамматикасында Грейбахтың нормалды пішімі қалай тұрғызылады?

5. Грейбахтың нормалды пішіміне түрлендіретін алгоритмнің кірісі не?

III.2. Контекстісіз тілдерді танушы механизмдер

III.2.1. Стекті автоматтардың құрамы мен құрылымы

Бұл параграфта контекстісіз тілдерді танитын механизмдеріне жататын стекті автоматтар анықталады, мысалдар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1-13,15-21,25,31].

Стекті автомат – ол контексті тілді танитын бір жақты танығыш. Оның құрылғысы ақырлы автоматтардың құрылғысына жақын, бірақ ол тағыда қосымша жұмыс таспасымен – стекті жадпен жабдықталған.

Стектік жад"*соңында кіру*, *бірінші шығу – last input*, *first output* (*LIFO*)" прісіпімен ұйымдастырылған. Стектік жадқа қолжетім *стек шыңы* деп аталатын тек жоғарғы ұяшық арқылы жүзеге асырылады, яғни символдар тек стек шыңынан стекке кіргізілед енгізіледі (кіргізіледі) және стектен шығарылады (итеріледі). Ол үшін стектік автоматта ылғи стек шыңын қадағалайтын қосымша бастиек қарастырылған.

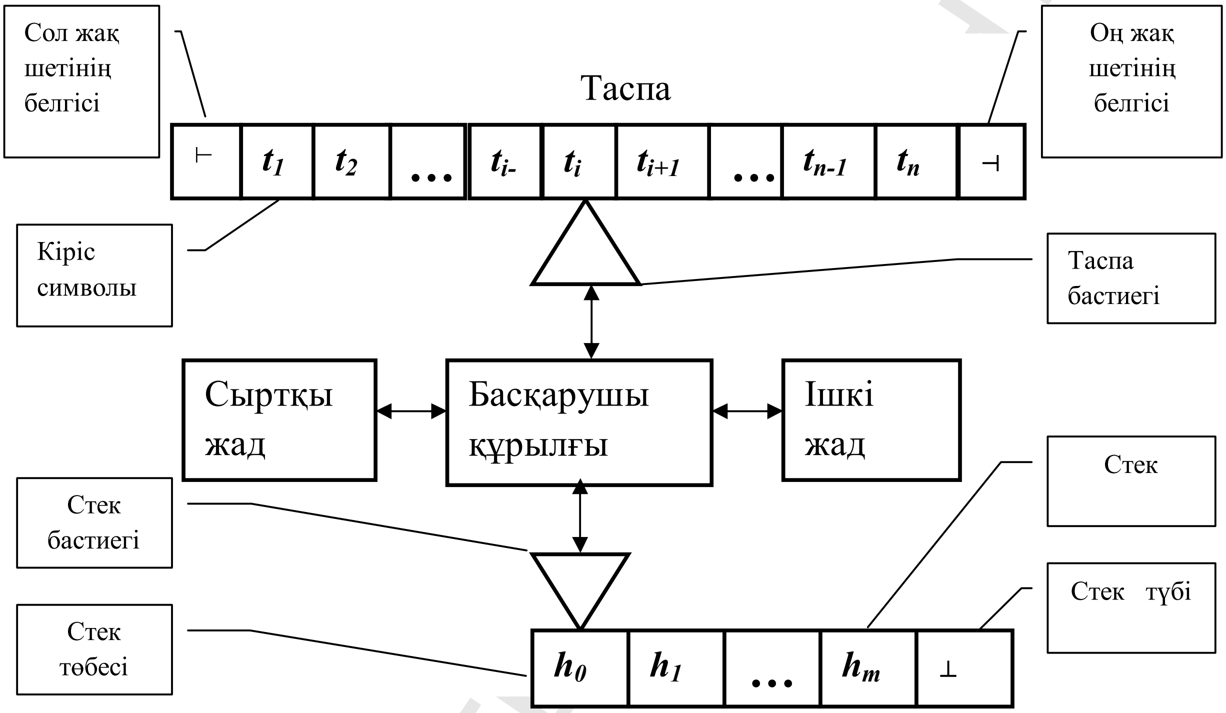
Стектегі деректерді $\boldsymbol{H}$ әліпбиінің h_i $(0 \leq i \leq n)$ символдарының ақырлы тізбесі түрінде бейнелеуге болады және оның төбесіндегі элементті тізбенің ең сол жақтағы символы деп есептейді. Мұнда, егер стекке соңғы жазылғаннан бастап *i-ші* символ оқылатын болса, онда алдымен i-1 аралық символдар өшірілуі қажет. Стектің ішіндегі ең соңғы символды шығарғаннан кейін, оның ішінде бос тізбе қалады, оны стектің түбін белгілегіш көрсетеді.

Стекті автомат құрамында *таспа*, *ішкі жад*, *сыртқы жад*, *таспа бастиегі*, *басқарушы құрылғы*, *стектік жад*, *стек бастиегі* бар. Стекті автомат өзінің жұмысын таспада кіріс символдарының тізбесі және стек бос болған алғашқы күйден бастайды. Келесі кескіндер оқылған кіріс символымен, басқару құрылғысының ішкі күйінің символымен және стектің төбесіндегі символмен анықталады. Автоматтың бір жұмыс тактісінде бастиек мынадай әрекеттер жасай алады:

1) стектің бір символын өшіреді, сонда стекте қалған барлық символдар бір ұяшыққа жоғары көтеріледі;

2) стектің бір символын өшіреді және стекке символдардың бос емес тізбесін жазады, сонда стекте қалған барлық символдар жазылатын тізбенің ұзындығына тең ұяшыққа төмен жылжиды.

Стекті автоматтың құрамы мен құрылымы III.2.1-суретте көрсетілген.



III.2.1-сурет. Стекті автоматтың құрамы мен

III.2.1.1-анықтама. Стекті автомат келесі ондықпен анықталады

$$\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, q_s, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta>,$$ мұнда:

$\boldsymbol{Q}$ – басқарушы құрылғының күйлерінің ақырлы жиыны;

$\boldsymbol{T}$ – кіріс символдарының ақырлы жиыны, $\boldsymbol{Q} \cap \boldsymbol{T} = \emptyset$, $\perp \notin \boldsymbol{T}$;

q_s – басқарушы құрылғының алғашқы күйі, $q_s \in \boldsymbol{Q}$;

$\boldsymbol{F}$ – басқарушы құрылғының соңғы күйлерінің жиыны, $\boldsymbol{F} \subseteq \boldsymbol{Q}$;

q_f – басқарушы құрылғының соңғы күйі, $q_f \in \boldsymbol{F}$;

$\boldsymbol{H}$ – стек символдарының ақырлы жиыны;

h_0 – стек төбесіндегі алғашқы символ, $h_0 \in \boldsymbol{H}$;

$\vdash, \dashv$ – таспаның сол жақ шеті мен оң жақ шетін белгілегіштер, $\vdash, \dashv \notin \boldsymbol{T}$;

ε – бос тізбе;

$\perp$ – стек түбінің белгілегіші;

δ – бейдетерминді стекті автоматтың (БСА) ауысулар функциясы $\boldsymbol{Q}\times(\boldsymbol{T}\cup\{\varepsilon\})\times\boldsymbol{H}$ жиынын $\boldsymbol{Q}\times\boldsymbol{H}^*$ жиынының барлық ішжиындарының жиынына бейнелейді, яғни

$$\delta: \boldsymbol{Q}\times(\boldsymbol{T}\cup\{\varepsilon\})\times\boldsymbol{H}\rightarrow\vartheta(\boldsymbol{Q}\times\boldsymbol{H}^*),$$

мұнда $\vartheta(\boldsymbol{Q}\times\boldsymbol{H}^*)$ – $\boldsymbol{Q}\times\boldsymbol{H}^*$ жиынының барлық ішжиындарының жиыны.

δ – детерминді стекті автоматтың (ДСА) ауысулар функциясы $\boldsymbol{Q}\times(\boldsymbol{T}\cup\{\varepsilon\})\times\boldsymbol{H}$ жиынын $\boldsymbol{Q}\times\boldsymbol{H}^*$ жиынының барлық ішжиындарының жиынына бейнелейді, яғни $\delta: \boldsymbol{Q}\times(\boldsymbol{T}\cup\{\varepsilon\})\times\boldsymbol{H}\rightarrow\boldsymbol{Q}\times\boldsymbol{H}^*$.

$\boldsymbol{\Delta}\subseteq(\boldsymbol{Q}\times\boldsymbol{T}^*\times\boldsymbol{H}^*)\times(\boldsymbol{Q}\times\boldsymbol{H}^*)$ – ауысулар жиыны.

III.2.1-ескертпе. Стекті автоматты күйлер диаграммасымен кескіндеуге болады. Диаграммада әрбір күй төбе болатын шеңбермен, ал ауысулар қабырға болатын бағыттамалармен бескілделеді. Әрбір алғашқы күй өзіне кіретін қысқа бағыттама арқылы танылады. Әрбір соңғы күй екі шеңбермен ерекшеленеді. Диаграммада x, δ: γ белгісі бар p-дан q-ге баратын бағыттамасы бар қабырға стектік автоматтың $<<p, x, \delta>, <q, \gamma>>$ ауысуы болады.

Мейлі СА $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, q_s, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta>$, мұнда $q\in\boldsymbol{Q}$, $t\in\boldsymbol{T}$, $\tau\in\boldsymbol{T}^*$, $h\in\boldsymbol{H}$, $\eta\in\boldsymbol{H}^*$, $q_i\in\boldsymbol{Q}$, $\eta_i\in\boldsymbol{H}^*$, $1\leq i\leq m$ болсын. Сонда $\boldsymbol{M}$ СА-ның жұмысын функция терминінде былай сипаттауға болады:

1) мынадай $\delta(q, t, h) = \{(q_1, \eta_1), (q_2, \eta_2),\ldots,(q_{\mathrm{m}}, \eta_{\mathrm{m}})\}$ жазбалар СА q күйінде стек шыңында h символын және кіріс таспасында t символын оқып, q_i күйіне өтеді, стекте h символын η_i тізбесіне ауыстырады және бастиекті бір ұяшыққа оңға жылжытатындығын білдіреді;

2) мынадай $\delta(q, \varepsilon, h) = \{(q_1,\eta_1), (q_2,\eta_2), \ldots, (q_m,\eta_m)\}$ жазбалар СА q күйінде стек шыңында h символын оқып және кіріс таспасында қандай символ оқуына байланыссыз q_i күйіне өтеді, стекте h символын η_i тізбесіне ауыстырады, ал бастиек жылжымайтындығын

білдіреді. Бұл амал стектің ішіндегісін өзгерту үшін пайдаланылады және ε-*қозғалу* деп аталады.

3) егер стек бос болса, яғни $\eta_i = \varepsilon$, онда келесі такт мүмкін емес.

Стектегі h символы η_i тізбесіне ауыстырылғаннан кейін, оның ең сол символы стек төбесінде, ал ең оң символы стек түбінде болады.

ІІІ.2.1.2-анықтама. Егер $q\in\boldsymbol{Q}$ – басқарушы құрылғының ағымдағы күйі; $q_s\in\boldsymbol{Q}$ – басқарушы құрылғының алғашқы күйі; $q_f\in\boldsymbol{F}$ – басқарушы құрылғының соңғы күйі; $\tau\in\boldsymbol{T}^*$ – кіріс таспасындағы қолданылмаған тізбе (оның ең сол жағындағы символды бастиек қарастырып тұр, егер $\tau=\varepsilon$, онда кіріс таспасы толық оқылды деп есептелінеді); $\eta\in\boldsymbol{H}^*$ – стектегі тізбе (оның ең сол жағындағы таңба стектің төбесіндегі таңба болып есептелінеді, егер $\eta=\varepsilon$, онда стек бос деп есептелінеді) және $h_0\in\boldsymbol{H}$ – стектің төбесіндегі таңба болса, онда мына үштік:

$(q_s, \tau, h_0)\in\boldsymbol{Q}\times\boldsymbol{T}^*\times\boldsymbol{H}^*$ – стекті автоматтың *алғашқы кескіні*;
$(q, \tau, \eta)\in\boldsymbol{Q}\times\boldsymbol{T}^*\times\boldsymbol{H}^*$ – стекті автоматтың *ағымдағы кескіні*;
$(q_f, \varepsilon, \eta)\in\boldsymbol{Q}\times\boldsymbol{T}^*\times\boldsymbol{H}^*$ – стекті автоматтың *соңғы кескіні*.

Алғашқы кескінде стекті СА алғашқы күйде, таспада кіріс символдарының тізбесі және стек бос болады. Автоматтың келесі кескіні оқылған басқарушы құрылғы күйінің символымен, таспадағы кіріс символымен және стек төбесіндегі символмен анықталады.

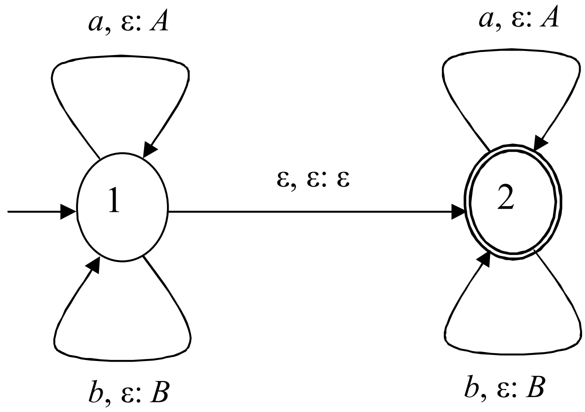
Автомат бір қадамда стек бастиегі келесі әрекеттерді жасайды:

1) төбедегі символды өшіреді, бұл жағдайда стекте қалған барлық символдар бір ұяшыққа жоғары көтеріледі;

2) төбедегі символды өшіріп, стекке бос емес символдар тізбесін жазады, бұл жағдайда стектің ішіндегісі, саны жазылатын тізбе ұзындығына тең болатындай, ұяшыққа төмен жылжиды.

$\boldsymbol{M}$ стекті автоматының *қадамы* оның $\boldsymbol{Q}\times\boldsymbol{T}^*\times\boldsymbol{H}^*$ жиынындағы кескіндерінде анықталған $\models_{\boldsymbol{M}}$ бинарлы қатынасы арқылы көрсетіледі, мұнда егер стекті автомат $\boldsymbol{M}$ алдын ала белгілі болса, онда $\models_{\boldsymbol{M}}$ бинарлы қатынасындағы көрсеткіш $\boldsymbol{M}$ көрсетілмейді.

III.2.1-мысал. Мейлі стектік автомат мынадай $\boldsymbol{Q}=\{1, 2\}$, $\boldsymbol{T}=\{a, b\}$, $\boldsymbol{H}=\{A, B\}$, $\boldsymbol{I}=1$, $\boldsymbol{F}=2$, $\boldsymbol{\Delta}=\{<<1,a,\varepsilon>, <1, A>>, <<1,b,\varepsilon>, <1,B>>, <<1,\varepsilon,\varepsilon>, <2,\varepsilon>>, <<2,a,A>, <2,\varepsilon>>, <2, b, B>, <2, \varepsilon>>\}$ параметрлерге ие болсын. Бұл стектік автоматтың диаграммасы III.2.1-суретте көрсетілген.



II.2.1.10-сурет. Стектік автоматтың диаграммасы

Мейлі стекті автомат $\boldsymbol{M}$ берілсін және $q\in\boldsymbol{Q}$, $q'\in\boldsymbol{Q}$, $t\in\boldsymbol{T}\cup\{\varepsilon\}$, $\tau\in\boldsymbol{T}^*$, $h\in\boldsymbol{H}$, $\eta\in\boldsymbol{H}^*$ болсын, сонда:

1) егер $\tau\neq\varepsilon$, $\tau=t\tau'$, $\tau'\in\boldsymbol{T}^*$, $\eta\neq\varepsilon$, $\eta=h\eta'$, $\eta'\in\boldsymbol{H}^*$ және $\xi\in\boldsymbol{H}^*$, онда стекті автоматтың қадамы $(q, t\tau', h\eta') \models (q', \tau', \xi\eta')$ қатынасы түрде жазылады және ол басқарушы құрылғының q күйінде, бастиек t кіріс символын қарастырып және стек төбесіндегі h символын ескеріп, бастиекті оңға қарай бір ұяшыққа жылжытатын, басқарушы құрылғыны q күйінен q' күйіне көшіретін, және стектің төбесіндегі h символын ξ тізбесіне ауыстыратын қадам жасайтындығын білдідіреді;

2) егер $\tau=\varepsilon$, $\eta\neq\varepsilon$, $\eta=h\eta'$, $\eta'\in\boldsymbol{H}^*$ және $\xi\in\boldsymbol{H}$, онда стекті автоматтың қадамы мына қатынас $(q, \varepsilon, h\eta')\models(q', \varepsilon, \xi\eta')$ түрінде жазылады және ол кіріс тізбесі толық оқылғандығын, басқарушы құрылғының q күйінде стек төбесіндегі h символын ескеріп, бастиекті жылжытпайтын, бірақ басқарушы құрылғыны q күйінен q' күйіне көшіретін және стектің төбесіндегі h символын γ тізбесіне ауыстыратын *ε-қадам* жасайтындығын білдідіреді;

3) егер $\eta=\varepsilon$, яғни стек бос болса, онда келесі қадам жоқ.

Әдеттегідей $\models$ қатынасының k-сыншы *дәрежесін* анықтауға болады, оны $\models^k$ арқылы белгілейміз: егер берілген кіріс тізбесі $t_1t_2. . .t_k\tau$ үшін $\boldsymbol{Q}$ жиынында q_1, q_2, ..., q_k, q_{k+1} күйлер тізбегі және $\boldsymbol{H}^*$ жиынында $\eta_1,\eta_2,...,\eta_k,\eta_{k+1}$ тізбелер тізбегі табылса, онда былай жазуға болады: $(q_1, t_1t_2. . .t_k\tau, \eta_1) \models^k (q_{k+1}, \tau, \eta_{k+1})$.

Енді, егер кез келген $i\geq1$ немесе $i\geq0$ үшін $(q_0, \tau, h_0) \models^i (q_i, \varepsilon, \eta)$ қатынасы орындалса, онда оны былай $(q_0, \tau, h_0) \models^+ (q_i, \varepsilon, \eta)$ немесе былай $(q_0, \tau, h_0) \models^* (q_i, \varepsilon, \eta)$ жазуға болады, мұнда $q_0 = q_s$, $q_i \in \boldsymbol{F}$, $\tau \in \boldsymbol{T}^*$, $h_0 \in \boldsymbol{H}$, $\eta \in \boldsymbol{H}^*$ және $\models$ қатынасы үшін $\models^+$ – транзитивтік тұйықталуы, ал $\models^*$ – рефлексивтік және транзитивтік тұйықталуы.

III.2.1-тапсырмалар:

1. Стекті автоматтардың ақырлы автоматтардан айырмашылығын анықтаңыз.
2. Стекті автоматтың элементтерін көрсетіңіз
3. Стекті автоматтың құрылымын сызыңыз.
4. Стекті автоматтың ішкі жадының қызметін сипаттаңыз.
5. Стекті автоматтың сыртқы жадының қызметін сипаттаңыз.
6. Стекті автоматтың формалды анықтамасын беріңіз.
7. Стекті автоматтың ауысу функциясын сипаттаңыз.
8. Стекті автоматтың жұмыс істеу қадамын сипаттаңыз.
9. Мына $\models$ қатынасының рефлексивтік және транзитивтік тұйықталуын анықтаңыз.

III.2.1-сұрақтар:

1. Стекті автоматтың жады қалай жұмыс істейді?
2. Стекті автомат қалай жұмыс істейді?
3. Стекті автоматтың тактісі деген не?
4. Стек бос тізбекті қамти ала ма?
5. Стек төбесі қалай белгіленеді?
6. Стек түбі қалай белгіленеді?
7. Стекті автоматтардың қандай түрлері бар?
8. Мына (q_f, ε, η) жазба нені білдіреді?

III.2.2 Стекті автомат танитын тізбелер мен тілдер

Бұл параграфта стекті автоматтар танитын тізбелер мен тілдер қарастырылады, оларға байланысты тапсырмалар берілелі және сұрақтар қойылады [1-13,15-21,25,31].

III.2.2.1-анықтама. Стекті автомат $\tau\in \boldsymbol{T}^*$ кіріс тізбесін екі әдіспен тани алады:

1) егер кейбір $q_i\in \boldsymbol{F}$ $(i\geq 1)$, $h_0\in \boldsymbol{H}$, $\eta\in \boldsymbol{H}^*$ үшін мына қатынас $(q_s,\tau, h_0)\models^*(q_i, \varepsilon, \eta)$ орындалса, онда соңғы күй арқылы;

2) егер кейбір $q\in \boldsymbol{Q}$ үшін мына қатынас $(q_s,\tau, h_0)\models^*(q, \varepsilon, \varepsilon)$ орындалса, онда стектің бос болуы арқылы.

***III.2.2.2-анықтама*.** $\boldsymbol{M}$ стекті автоматпен танылған тізбелер жиынын, осы автоматпен танылатын $\boldsymbol{L}(\boldsymbol{M})$ тілі деп атайды.

Стекті автомат $\boldsymbol{M}$ үшін танылатын тіл $\boldsymbol{L}(\boldsymbol{M})$ арқылы белгіленеді және ол екі әдіспен анықталады:

1) соңғы күй арқылы былай

$$\boldsymbol{L}(\boldsymbol{M})\rightleftharpoons\{\tau\colon \tau\in \boldsymbol{T}^* \ \&\ (q_s, \tau, h_0)\models^* (q_f, \varepsilon, \eta)\ \&\ \eta\in \boldsymbol{H}^*\}$$

2) стектің бос болуы арқылы былай

$$\boldsymbol{L}(\boldsymbol{M})\rightleftharpoons\{\tau\colon \tau\in \boldsymbol{T}^* \ \&\ (q_s, \tau, h_0)\models^* (q, \varepsilon, \varepsilon)\ \&\ q\in \boldsymbol{Q}\}$$

Егер тіл стектің бос болуы арқылы танылса, онда ол автоматтың соңғы күйлерінің жиыны пайдаланылмайды және бұл жағдайда әдетте $\boldsymbol{F} = \varnothing$ деп қабылданады.

Соңғы күй арқылы танылған тілдердің тобы стектің бос болуы арқылы танылған тілдердің тобына дәл келеді.

III.2.2.1-мысал. Мейлі $\boldsymbol{L}_1=\{0^n 1^n\colon n\geq 0\}$ тілі берілсін. $\boldsymbol{L}_1=\boldsymbol{L}(\boldsymbol{M}_1)$ болатындай стекті автомат $\boldsymbol{M}_1 = <\boldsymbol{Q}, \boldsymbol{T}, q_s, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta>$ құрастырамыз. Ол үшін $\boldsymbol{Q}=\{q_0, q_1, q_2\}$, $\boldsymbol{T} = \{0,1\}$, $q_s=q_0$, $\boldsymbol{F}=\{q_0\}$, $\boldsymbol{H}=\{h,0\}$, $h_0=h$ және

$$\delta(q_0, 0, h) = \{(q_1,0h)\},$$
$$\delta(q_1,0,0) = \{(q_1,00h)\},$$
$$\delta(q_1,1,0) = \{(q_2,0h)\},$$

$$\delta(q_2,1,0) = \{(q_2,h)\},$$
$$\delta(q_2,\varepsilon, h) = \{(q_0,\varepsilon)\}.$$

Бұл автомат жұмысы стекке кіріс тізбесінің 0-ден тұратын алғашқы бөлігін көшіріп, сонан кейін кіріс тізбесінде өзі көрген әрбір 1-ге сәйкес стектен бір 0-ді жоятындықтан тұрады. Сонымен қатар, күйлерге ауысу барлық 0-дер 1-лердің алдында тұрғандығына кепіл болады. мысалы, 0011 кіріс тізбесі үшін $\boldsymbol{M}_1$ автоматы келесі қадамдар тізбегін жасайды:

$$(q_0, 0011,h) \vDash (q_1,011,0h) \vDash$$
$$(q_1,11,00h) \vDash (q_2,1,0h) \vDash$$
$$(q_2,\varepsilon,h) \vDash (q_0,\varepsilon,\varepsilon)$$

1) $\boldsymbol{L}_1\subseteq\boldsymbol{L}(\boldsymbol{M}_1)$ екендігін дәлелдеу. Жалпы жағдайда келесі қадамдар тізбесін жазуға болады

$$(q_0, 0, h) \vDash (q_1, \varepsilon, 0h)\ ) \vDash (q_1, 0^i, 0h) \vDash^i (q_1, \varepsilon, 0^{i+1}h)\ ) \vDash$$
$$(q_1, 1, 0^{i+1}h) \vDash (q_2, \varepsilon, 0^ih)\ ) \vDash (q_2, 1^i, 0^ih) \vDash^i (q_2, \varepsilon, h)$$
$$(q_2, e, h) \vDash (q_0, \varepsilon, \varepsilon)$$

Осыларды біріктіре отырып, $n\geq 1$ үшін мыналарды аламыз

$$(q_0, 0^n1^n, h) \vDash^{2n+1} (q_0,\varepsilon,\varepsilon) \text{ және } (q_0,\varepsilon, h) \vDash (q_0,\varepsilon,h)$$

Сонымен, $\boldsymbol{L}_1\subseteq\boldsymbol{L}(\boldsymbol{M}_1)$.

2) $\boldsymbol{L}_1\supseteq\boldsymbol{L}(\boldsymbol{M}_1)$ екенін, яғни $\boldsymbol{M}_1$ автоматы тек 0^n1^n түріндегі тізбелерді танитынын көрсетеміз. Дәлелдеудің бұл бөлігі қиындау. Әдетте танығыш мынадай тізбелерді танитындығы оңай дәлелденеді, сондай-ақ, грамматика үшін, ол тізбелердің белгілі түрлерін тудыратындығы едәуір қиын дәлелденеді.

Осыдан, егер $\boldsymbol{M}_1$ автоматы бос емес тізбені таныса, онда ол q_0, q_1, q_2, q_0 күйлерін осы ретпен өтуі керектігі ескеріледі.

Егер $i\geq 1$ үшін $(q_0, \tau, h) \vDash^i (q_1, \varepsilon, \eta)$ орындалса, онда $\tau = 0^i$ жөне $\eta = 0^ih$. Дәл осылай, егер $(q_2, \tau, \eta) \vDash^i (q_2, \varepsilon, \beta)$ орындалса, онда $\tau = 1^i$ жөне $\eta = 0^i\beta$. Сондай-ақ, $(q_1, \tau, \eta) \vDash (q_2, \varepsilon, \beta)$ орындалады тек сонда ғана, егер $\tau =1$ және $\eta = 0\beta$ болса. Ал, $(q_2,\tau,h) \vDash^* (q_0, \varepsilon, \varepsilon)$ орындалады тек сонда ғана, егер $\tau = \varepsilon$ болса. Сонымен, егер қайсы бір $i\geq 0$ үшін $(q_0, \tau, h) \vDash^i (q_0, \varepsilon, \eta)$

орындалса, онда $\tau = \varepsilon$, $i = 0$ немесе $\tau = 0^n 1^n$, $i = 2n+1$, $\eta = \varepsilon$. Бұдан $\boldsymbol{L}_1 \supseteq \boldsymbol{L}(\boldsymbol{M}_1)$.

Дәлелденген екі қосымша тұжырымнан $\boldsymbol{L}_1 = \boldsymbol{L}(\boldsymbol{M}_1)$ шығады. Басқаша айтқанда, соңғы күй арқылы танылған тілдердің тобы стектің бос болуы арқылы танылған тілдердің тобына дәл келеді.

III.2.2.2-мысал. Бос стекті СА $\boldsymbol{M} = < \boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{H}, q_s, \boldsymbol{F}, h_0, \vdash, \dashv, \delta>$ мына тілді $\{xcx^{-1} : x \in \{0, 1\}^*\}$ таниды, мұнда:

$\boldsymbol{T} = \{0, 1, c\}$, $\boldsymbol{Q} = \{q_1, q_2\}$, $\boldsymbol{H} = \{R, B, D\}$, $q_0 = q_1$, $h_0 = R$, ал δ:

$\delta(q_1, 0, R) = \{(q_1, BR)\}$, $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$,

$\delta(q_1, 0, B) = \{(q_1, BB)\}$, $\delta(q_2, 1, D) = \{(q_2, \varepsilon)\}$,

$\delta(q_1, 0, D) = \{(q_1, BD)\}$, $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$,

$\delta(q_1, c, R) = \{(q_2, R)\}$, $\delta(q_1, 1, R) = \{(q_1, DR)\}$,

$\delta(q_1, c, B) = \{(q_2, B)\}$, $\delta(q_1, 1, B) = \{(q_1, DB)\}$,

$\delta(q_1, c, D) = \{(q_2, D)\}$, $\delta(q_1, 1, D) = \{(q_1, DD)\}$.

Мейлі x=001, сонда xcx^{-1} = 001c100. Кіріс тізбе 001c100 үшін ауысу тізбегі былай $(q_1, R) \models^0 (q_1, BR) \models^0 (q_1, BBR) \models^1 (q_1, DBBR) \models^c (q_2, DBBR) \models^1 (q_2, BBR) \models^0 (q_2, BR) \models^0 (q_2, R) \models^\varepsilon (q_2, \varepsilon)$.

Енді біз стекті автоматтардың кейбір түрлерін анықтап, олармен анықталатын тілдер және әдеттегі стекті автоматтармен танылатын тілдер арасында байланыс орнатамыз.

III.2.2.3-анықтама. *Кеңейтілген стекті автомат* деп келесі ондықты айтамыз $\boldsymbol{M} = < \boldsymbol{Q}, \boldsymbol{T}, q_s, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta>$, мұнда δ – $\boldsymbol{Q} \times (\boldsymbol{T} \cup \{\varepsilon\}) \times \boldsymbol{H}^*$ жиынының ақырлы ішжиынын $\boldsymbol{Q} \times \boldsymbol{H}^*$ жиынының барлық ақырлы ішжиынына бейнелеу, яғни δ: $\boldsymbol{Q} \times (\boldsymbol{T} \cup \{\varepsilon\}) \times \boldsymbol{H}^* \rightarrow \vartheta(\boldsymbol{Q} \times \boldsymbol{H}^*)$, ал қалған барлық символдар бұрынғы мәндеріне ие болады.

Кеңейтілген стекті автомат $\boldsymbol{M}$ үшін кескін және осы автоматпен танылатын тіл әдеттегі стекті автоматындай болады.

Кеңейтілген стекті автомат танитын тілдер тобы әдеттегі стекті автоматы танылатын тілдер тобымен сәйкес болады. Бірақ әдеттегі стекті автоматтан айырмашылығы - кеңейтілген стекті автомат өз жұмысын стек бос болғанда да жалғастыра береді.

Мейлі $\boldsymbol{M} = \boldsymbol{<} \boldsymbol{Q}, \boldsymbol{T}, q_s, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta>$ – бейдетерминді стекті автомат немесе кеңейтілген стекті автомат.

III.2.2.4-анықтама. M автоматы $\tau \in \boldsymbol{T}^*$ тізбесін стек бос болғанда таниды, егер кейбір $q \in \boldsymbol{Q}$ үшін $(q_s, \tau, h_0) \models^+ (q, \varepsilon, \varepsilon)$ орындалса.

III.2.2.5-анықтама. Стек бос болғанда танитын $\tau \in \boldsymbol{T}^*$ тізбелер жиыны стекті автомат $\boldsymbol{M}$ стек бос болғанда танитын $\boldsymbol{L}_\varepsilon(\boldsymbol{M})$ тілі деп аталады, яғни

$$\boldsymbol{L}_\varepsilon(\boldsymbol{M}) = \{\tau \colon \tau \in \boldsymbol{T}^* \ \& \ (q_s, \tau, h_0) \models^+ (q, \varepsilon, \varepsilon) \ \& \ q \in \boldsymbol{Q}\}$$

Стекті автомат стек бос болғанда танитын тілдер тобы әдеттегі бейдетерминді стекті автомат танитын тілдер тобымен сәйкес болады, яғни кез келген берілген стекті автомат $\boldsymbol{M'}$ үшін $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}_\varepsilon(\boldsymbol{M'})$ болатындай стекті автомат $\boldsymbol{M}$ тұрғызуға болады.

III.2.2-тапсырмалар:

1. $\{a^n b^n : n \geq 0\}$ тілін танитын бейдетерминді стектік автоматты тұрғызыңыз.
2. $\{a^n b^n c^n : n \geq 0\}$ тілін танитын бейдетерминді стектік автоматты тұрғызыңыз.
3. $\{a^i b^j c^k : i=j$ немесе $j=k\}$ тілін танитын бейдетерминді стектік автоматты тұрғызыңыз.
4. $\{\tau \in \{a,b\}^* : |\tau|_a = |\tau|_b\}$ тілін танитын бейдетерминді стектік автоматты тұрғызыңыз.
5. $\{\tau \in \{a,b,c\}^* : |\tau|_a = |\tau|_b, |\tau|_c = 1\}$ тілін танитын бейдетерминді стектік автоматты тұрғызыңыз.
6. Найти стековый автомат, распознающий язык:

$$\{\omega \in \{a, b\}^* : |\omega|_a = |\omega|_b\}.$$

III.2.2-сұрақтар:

1. $\{a^{m+k} b^m a^k b^m : m \geq 0, k \geq 0\}$ тілі стектік автоматпен таныла ма?
2. $\{a^k b^{m+k} a^k b^m : m \geq 0, k \geq 0\}$ тілі стектік автоматпен таныла ма?
3. $\{\tau \in \{a,b\}^* : |\tau|_a = |\tau|_b + 1\}$ тілі стектік автоматпен таныла ма?

III.2.3. Детерминді стекті автоматтар

Бұл параграфта біз ДСА қарастырылады, оларға байланысты тапсырмалар берілелі және сұрақтар қойылады [1-13,15-21,25,31].

Бейдетерминді стекті автомат тілді анықтаудың ыңғайлы абстракциясын қамтамасыз етеді, бірақ практикалық жүзеге асыру үшін оны детерминді түрге айнылдыру керек. Мұнда бейдетерминді және детерминді стекті автоматтардың функционалды мүмкіншіліктері әртүрлі екендігін ескеру қажет. Өкінішке орай ДСА қуаты БСА қуатына қарағанда аздау. ДСА тани алмайтын контекстісіз (контексті-бос) тіл бар.

III.2.3.1-анықтама. Берілген стекті автомат $\boldsymbol{M} = < \boldsymbol{Q}, \boldsymbol{T}, q_0, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta >$ детерминді стекті автомат (ДСА) деп аталады, егер әрбір $q \in \boldsymbol{Q}$ және $h \in \boldsymbol{H}$ үшін

(1)әрбір $t \in \boldsymbol{T}$ үшін $\delta(q,t,h)$ жиыны бірден көп емес элемент қамтиды және $\delta(q,\varepsilon,h)=\emptyset$, иә

(2) әрбір $t \in \boldsymbol{T}$ үшін $\delta(q,t,h)=\emptyset$ және $\delta(q,\varepsilon,h)$ бір ғана элемент қамтиды.

Осы екі шектеу бойынша ДСА әрбір кескінде бір ғана қадам жасайды. Практикада бейдетерминді автоматқа қарағанда детерминді автоматты моделдеу көп жеңіл болады.

III.2.3-келісім. ДСА үшін $\delta(q,t,h)$ бірден көп емес элемент қамтитын болғандықтан, $\delta(q,t,h) = \{(r,\eta)\}$ орнына $\delta(q,t,h) = (r,\eta)$ деп жазамыз.

III.2.3.2-анықтама. $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, q_0, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta>$ кеңейтілген стектік автомат детерминді деп аталады, егер мына шарттар орындалса:

(1) барлық $q \in \boldsymbol{Q}$, $t \in \boldsymbol{T} \cup \{e\}$ және $\eta \in \boldsymbol{H}^*$ үшін $\delta(q, t, \eta) \leq 1$;

(2) егер $\delta(q, t, \alpha) \neq \emptyset$, $\delta(q, t, \beta) \neq \emptyset$ және $\alpha \neq \beta$, онда α мен β тізбелерінің ешқайсысы басқаға жалғанбайды;

(3)егер $\delta(q, t, \alpha) \neq \varnothing$ және $\delta(q, \varepsilon, \beta) \neq \varnothing$, онда α мен β тізбелерінің ешқайсысы басқаға жалғанбайды;

Алдымен кіріс тізбесінің бөлігі оқылмай қалатын кез келген кескін үшін кезекті қадам болатындай етіп ДСА-ты түрлендіреміз. Очыны келесі лемма көрсетеді.

III.2.3.1-лемма. Мейлі $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, q_0, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta>$ – ДСА. Сонда оған эквивалентті $\boldsymbol{M}' = <\boldsymbol{Q}', \boldsymbol{T}', q'_0, \boldsymbol{F}', \boldsymbol{H}', h'_0, \vdash, \dashv, \perp, \delta'>$ ДСА былай тұрғызуға болады:

(1) барлық $t \in \boldsymbol{T}$, $q \in \boldsymbol{Q}'$ және $h \in \boldsymbol{H}'$ иә

(а) $\delta'(q, t, h)$ дәл бір элемент қамтиды және $\delta'(q, \varepsilon, h) = \varnothing$, иә

(б) $\delta'(q, t, h) = \varnothing$ және $\delta'(q, \varepsilon, h)$ дәл бір элемент қамтиды;

(2) егер кейбір $t \in \boldsymbol{T} \cup \{\varepsilon\}$ үшін $\delta'(q, t, h'_0) = (\mathrm{r}, \gamma)$, онда кейбір тізбе $\alpha \in \boldsymbol{H}^*$ үшін $\gamma = \alpha h'_0$.

III.2.3.3-анықтама. ДСА $\boldsymbol{M}$ үшін кескін (q, ω, α) қайталымды болады, егер әрбір $i \geq 1$ үшін $(q, \omega, \alpha) \models (p_i, \omega, \beta_1) \models (p_2, \omega, \beta_2) \models \ldots$ және $|\beta| \geq |\alpha|$ болатындай кескін (p_i, ω, β_i) табылса.

Сонымен, кескін қайталымды болады, егер $\boldsymbol{M}$ стекті қысқартпай (мұнда стек иә шексіз өсе алады иә қайталмалы бірнеше әртүрлі тізбелермен дәл келе алады) шексіз сан рет ε-қадам жасай алса.

Стекті қысқартушы бір қатар ε-қадамнан кейін қайталмалы кескінге өтетін қайталанбайтын кескіндер бар екендігін ескереміз. Кез келген берілген кескіннен ақырлы және есептелетін сан қадам арқылы қайталымды кескінге тап болмай шексіз ε-қадам жасауға болмайтындығын көрсетеміз.

Егер $\boldsymbol{M}$ кіріс тізбесінің ортасында қайталымды кескінге түссе, онда ол енді кіріс тізбесін пайдаланбайды. Біз берілген $\boldsymbol{M}$ ДСА-ын ешқашан қайталымды кескінге түспейтін $\boldsymbol{M}'$ ДСА-ына түрлендіміз.

III.2.3.1-алгоритм. Қайталымды кескінді табу.
Кіріс. ДСА $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, q_0, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta>$.
Шығыс.

(1) C_1 = {(q, A): (q, ε, A) – қайталымды кескін және кейбір $\alpha \in \boldsymbol{H}^*$ тізбесі үшін $(q,\varepsilon,A) \models^* (r,\varepsilon,\alpha)$ болатындай $r \in \boldsymbol{F}$ жоқ} және

(2) C_2 = {(q, A): (q, ε, A) – қайталымды кескін және кейбір $r \in \boldsymbol{F}$ және $\alpha \in \boldsymbol{H}^*$ үшін $(q,\varepsilon,A) \models^* (r,\varepsilon,\alpha)$}.

Әдіс. Мейлі $\#\boldsymbol{Q} = n_1$, $\#\boldsymbol{H} = n_2$ және l– $\boldsymbol{M}$ бір қадамда стекке жаза алатын ең ұзын тізбенің ұзындығы болсын. Мейлі $n_3 = n_1(n^{n_1 n_2^{1+l}} - n_2)/(n_2-1)$, егер $n_2>1$ және $n_3 = n_1$, егер $n_2=1$. n_3 – $\boldsymbol{M}$ қайталымсыз жасайтын ε-қадамдардың максималды саны.

(1) Кейбір $r \in \boldsymbol{Q}$ және $\alpha \in \boldsymbol{H}^+$ үшін $(q, \varepsilon, A) \models^{n_3} (r, \varepsilon, \alpha)$ орындалатындығын барлық $q \in \boldsymbol{Q}$ және $A \in \boldsymbol{H}$ үшін анықтау керек. Мұнда $\boldsymbol{M}$-нің тура моделдеуі пайдаланады. Егер орындалатын болса, онда (q, ε, A) – қайталымды кескін, себебі бұл жағдайда $(q, \varepsilon, A) \models^* (q', \varepsilon, A'\beta) \models^m (q', \varepsilon, A'\gamma\beta) \models^{m(j-1)} (q', \varepsilon, A'\gamma^j\beta)$, мұнда m>0, j>0 болатындай мынадай сыңар (q', A'), мұнда $q \in \boldsymbol{Q}$ және $A' \in \boldsymbol{H}$.

(2) Егер (q, ε, A) – қайталымды кескін болса, онда кейбір $0 \le j \le n_3$ үшін $(q, \varepsilon, A) \models^j (r, \varepsilon, \alpha)$ мынадай $r \in \boldsymbol{F}$ бар екендігін анықтау. Мұнда тағыда тура моделдеу пайдаланады. Егер иә, онда C_2-ге (q, A) қосу. Кері жағдайда $\mathbf{C}_1$-ге (q, A) қосу. Егер $\boldsymbol{M}$ (q, ε, A) кескінінен бастап соңғы кескінге жететін болса, онда ол n_3 немесе одан аз қадамда жүзеге асады.

III.2.3-алгоритмі C_1 және C_2 жиындарын дұрыс табады.

III.2.3.4-анықтама. $\boldsymbol{M} = \langle \boldsymbol{Q}, \boldsymbol{T}, q_0, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta \rangle$ ДСА оқуды жалғастырушы деп атайды, егер әрбір $\omega \in \boldsymbol{T}^*$ тізбе үшін мынау $(q_0, \omega, h_0) \models^* (p, \varepsilon, \gamma)$ орындалатындай $p \in \boldsymbol{Q}$ және $\alpha \in \boldsymbol{H}^*$ бар. Оқуды жалғастырушы ДСА – әрбір кіріс тізбесін соңына дейін оқитын автомат.

III.2.3.2-лемма. Егер $\boldsymbol{M} = \langle \boldsymbol{Q}, \boldsymbol{T}, q_0, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta \rangle$ ДСА болса, онда оған эквивалентті оқуды жалғастырушы ДСА $\boldsymbol{M}'$ бар.

III.2.3.1-теорема. Егер $\boldsymbol{L}=\boldsymbol{L}(\boldsymbol{M})$ кейбір ДСА $\boldsymbol{M}$ үшін, онда $\boldsymbol{L}=\boldsymbol{L}(\boldsymbol{M}')$ кейбір ДСА $\boldsymbol{M}'$ үшін.

Ақырлы автоматтар үшін детерминді және бейдетерминді моделдері танитын тілдер тобы эквивалентті екендігі дәлелденген. Осындай эквиваленттілік стекті автоматтар үшін жоқ. Детерминді стекті автоматтар контекстісіз тілдердің тек кейбір детерминді контекстісіз тілдер деп аталатын ішжиындарын ғана таниды.

III.2.3-мысал. $\{xx^T : x\in\{0,1\}^*\}$ тілі ***M*** бейдетерминді стектік автоматпен танылады. Мейлі $\boldsymbol{M} = \langle \boldsymbol{Q}, \boldsymbol{T}, q_0, \varnothing, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta \rangle$, мұнда $\boldsymbol{Q} = \{q_1, q_2\}$, $\boldsymbol{T} = \{0, 1\}$, $\boldsymbol{H} = \{A, B, C\}$, $q_0 = q_1$, $\mathrm{h}_0 = A$, $\lambda\in \boldsymbol{H}^*$ және δ:

$$\delta(q_1, 0, A) = \{(q_1, BA)\},\ \delta(q_1, 1, C) = \{(q_1, CC), (q_2, \lambda)\},$$
$$\delta(q_1, 1, A) = \{(q_1, CA)\},\ \delta(q_2, 0, B) = \{(q_2, \lambda)\},$$
$$\delta(q_1, 0, B) = \{(q_1, BB), (q_2, \lambda)\},\ \delta(q_2, 1, C) = \{(q_2, \lambda)\},$$
$$\delta(q_1, 0, C) = \{(q_1, BC)\},\ \delta(q_1, \lambda, A) = \{(q_2, \lambda)\},$$
$$\delta(q_1, 1, B) = \{(q_1, CB)\},\ \delta(q_2, \lambda, A) = \{(q_2, \lambda)\}.$$

Бірақ, $\{xx^{\mathrm{T}} : x\in\{0.1\}^*\}$ тілін ешбір ДСА тани алмайды.

III.2.3-тапсырмалар:

1. $\{(01)^n : n\geq 0\}$ тілін танитын ДСА құрыңыз. $\boldsymbol{M}=\langle\{q_0,q_1,q_2,q_3,q_4,q_5,q_6\},\{0,1\},q_0,\{q_0\},\{\mathrm{Z},0\},\mathrm{Z},\delta\rangle$, кіріс тізбе 000*111* болсын.
2. $\{a^nb^n : n\geq 0\}$ тілін танитын ДСА құрыңыз. Мейлі $\boldsymbol{M} = \langle\{q_0,q_1,q_2\}, \{a, b\}, q_0, \{q_0\}, \{h_0, a\}, h_0, \vdash, \dashv, \perp, \delta\rangle$, ал кіріс тізбе *aabb*.
3. $\{\tau$□$\{a, b, c\}^*: 2|\tau|a = |\tau|b+|\tau|c,\ |\tau|a\geq|\tau|b\}$ тілін танитын ДСА болсын. $\boldsymbol{M} = \langle\{q_0,q_1,q_2\}, \{a,b\},q_0,\{q_0\}, \{h_0, a\}, h_0, \vdash, \dashv, \perp, \delta\rangle$, ал кіріс тізбе *abbccc* болсын.
4. $\{\omega\omega r\ \varepsilon\{k, L\}^*\}$ тілін танитын ДСА құрыңыз.

III.2.3-сұрақтар:

1. $\{a^nb^na^n : n\geq 0\} \cup \{a^kb^la^m: k\geq 0, \mathrm{l}\geq 0, m\leq 20\}$ тілі детерминді стектік автоматпен танылa ма?

2. $\{a^nb^mc^{nm}: n\geq 1, m\geq 1\} \cap \{a^{nm}b^nc^m: n\geq 1, m\geq 1\}$ тілі детерминді стектік автоматпен танылa ма?

3. $\{\tau\in\{a,b,c\}^*: |\tau|_a \neq |\tau|_b$ немесе $|\tau|_b \neq |\tau|_c\}$ тілі детерминді стектік автоматпен танылa ма?

III.3. Контекстісіз тілдердің қасиеттері

III.3.1. Контекстісіз грамматикалар және стекті автоматтардың эквиваленттілігі

Бұл параграфта констекстісіз грамматикалар мен стекті автоматтардың эквиваленттілігі қарастыралады, оларға байланысты мысаладар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1-4, 6-9, 11-13, 15-21, 24, 27-34].

Контекстісіз грамматикалармен (КГ) тудырылған тілдер тобы стекті автоматтармен (СА) танылған тілдер тобына дәл келетінін көрсетуге болады. Бұл берілген КБ-грамматикаға сәйкес бейдетерминді СА тұрғызу және керісінше СА-қа сәйкес КБ-грамматика құру керек. Ол үшін келесі анықтамаларды берейік:

III.3.1.1-анықтама. $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ КБ-грамматикасы *ε-ережелерсіз* (немесе *қысқармайтын*) деп аталады, егер:

1) иә $\boldsymbol{P}$ жиыны ε-ережелерді қамтымайды,
2) иә тек бір ғана ε-ереже $S \to \varepsilon$ бар және S бейтерминалы $\boldsymbol{P}$ жиынындағы қалған ережелерде кездеспейді.

III.3.1.2-анықтама. *Кеңейтілген СА-автомат* деп мына ондықты айтады $\boldsymbol{M} = < \boldsymbol{Q}, \boldsymbol{T}, q_s, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta>$, мұнда δ – $\boldsymbol{Q}\times(\boldsymbol{T}\cup\{\varepsilon\})\times\boldsymbol{H}^*$ жиынының ақырлы ішжиынын $\boldsymbol{Q}\times\boldsymbol{H}^*$ жиынының барлық ішжиындарының ақырлы жиынына бейнелейді, яғни δ: $\boldsymbol{Q}\times(\boldsymbol{T}\cup\{\varepsilon\})\times\boldsymbol{H}^*\to\vartheta(\boldsymbol{Q}\times\boldsymbol{H}^*)$, $\vartheta(\boldsymbol{Q}\times\boldsymbol{H}^*)$ – $\boldsymbol{Q}\times\boldsymbol{H}^*$ жиынының барлық ішжиындарының жиыны, ал қалған басқа символдар бұрынғы мағынасына ие болады.

Кеңейтілген СА-автомат үшін кескін мен осы автомат танитын тіл әдеттегі СА-автоматтағы сияқты анықталады.

III.3.1.1-лемма. Егер $\boldsymbol{L}(\boldsymbol{G})$ – контексті-бос тіл *Грейбахтың* нормалды пішіміндегі $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ грамматикасымен тудырылса, онда $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}(\boldsymbol{G})$ болатындай бейдетерминді СА-автомат $\boldsymbol{M}$ бар болады.

Дәлелдеу. Айталық $\boldsymbol{M} = \langle \boldsymbol{Q}, \boldsymbol{T}, q_s, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \bot, \delta \rangle$ СА-автомат болсын, мұнда δ: $\boldsymbol{Q}\times(\boldsymbol{T}\cup\{\varepsilon\})\times\boldsymbol{H}\rightarrow\vartheta(\boldsymbol{Q}\times\boldsymbol{H}^*)$, $\boldsymbol{Q} = \{q\}$, $\boldsymbol{T}=\boldsymbol{T}$, $q_S = q$, $\boldsymbol{F} = \varnothing$, $\boldsymbol{H} = \boldsymbol{N}\cup\boldsymbol{T}$, $h_0 = S$, ал δ ауысу функциясы былай анықталады:

(1) $\delta(q, t, A) = \{(q, t)\}$ тек қана сол жағдайда, қашан $A\rightarrow t\tau\in\boldsymbol{P}$ кейбір $A\in\boldsymbol{N}$, $t\in\boldsymbol{T}$, $\tau\in\boldsymbol{N}^*$ үшін;

(2) $\delta(q, t, t) = \{(q, \varepsilon)\}$ барлық $t\in\boldsymbol{T}$ үшін.

Біз кейбір $m\geq1$ және $n\geq1$ үшін мынаны көрсеткіміз келеді: $A\Rightarrow^m\omega$ тек қана және тек қана сол жағдайда, қашан $(q, \omega, A)\models^n (q, \varepsilon, \varepsilon)$, яғни

$$\exists m(m\geq1 \;\&\; A\Rightarrow^m\omega) \Leftrightarrow \exists n(n\geq1 \;\&\; (q, \omega, A)\models^n (q, \varepsilon, \varepsilon)) \quad \text{(III.3.1.1)}$$

Жеткілікті жағдай m бойынша индукциямен дәлелденеді.

Индукция базисі. Мейлі $A\Rightarrow^m\omega$ болсын. Егер $m=1$ және $\omega= t_1\ldots t_k$ $(k\geq0)$, онда $(q, t_1\ldots t_k, A)\models(q, t_1\ldots t_k, t_1\ldots t_k)\models^k (q, \varepsilon, \varepsilon)$.

Индукциялық қадам. Айталық, кейбір $m > 1$ үшін $A\Rightarrow^m\omega$ болсын. Бұл шығарымның бірінші қадамы әрбір $m_i<m$, $1\leq i\leq k$ және $x_1 \ldots x_k = \omega$ үшін мына түрде $A\Rightarrow X_1 \ldots X_k$ болады, мұнда $X_i\Rightarrow^{m}{}_i\, x_i$.

Егер $X_i\in\boldsymbol{N}$, онда индукция болжамы бойынша $(q, x_i, X_i)\models^*(q, \varepsilon, \varepsilon)$. Егер $X_i = x_i\in\boldsymbol{T}$, онда $(q, x_i, X_i)\models(q, \varepsilon, \varepsilon)$.

Осы тактылар тізбегін біріктіріп мынаны $(q, \omega, A)\models^+(q, \varepsilon, \varepsilon)$ алуға болады.

Жеткіліктілік шарты n бойынша индукциямен дәлелденеді.

Индукция базисі. Мейлі $(q, \omega, A)\models^n (q, \varepsilon, \varepsilon)$ болсын. Егер $n=1$, онда $\omega=\varepsilon$ және $A\rightarrow\varepsilon\in\boldsymbol{P}$, яғни егер $(q, \omega, A)\models^+ (q, \varepsilon, \varepsilon)$, онда $A \Rightarrow^+\omega$ дәлелденді.

Индукциялық қадам. Айталық, барлық $n_i<n$ үшін мына қатынас $(q, \omega, A)\models{}^{n}{}_i(q, \varepsilon, \varepsilon)$ орындалса. Сонда СА-автомат $\boldsymbol{M}$ жасаған бірінші қадамы $q, \omega, A)\models(q, \omega, X_1 \ldots X_k)$ түрде болуы керек, сонымен бірге, $1\leq i\leq k$ және $\omega = x_1 \ldots x_k$ үшін $(q, x_i, X_i)\models^{n}{}_i(q, \varepsilon, \varepsilon)$. Ал $A\rightarrow X_1\ldots X_k\in\boldsymbol{P}$, индукциялық болжам бойынша $X_i\in\boldsymbol{N}$ үшін $X_i\Rightarrow^+x_i$ және $X_i\in\boldsymbol{T}$ үшін $X_i\Rightarrow^0x_i$. Бұдан $\boldsymbol{G}$ грамматикасындағы A бейтерминалдан ω тізбесінің шығарымы былайша болады:

$$A \Rightarrow X_1 \ldots X_k$$

$$\Rightarrow^* X_1 \ldots X_{k-1}X_k$$
$$\ldots\ldots$$
$$\Rightarrow^* X_1 \ldots X_{k-1}X_k$$
$$\Rightarrow^* x_1 \ldots x_{k-1}x_k = \omega$$

(III.1) тұжырымынан мынау шығады $S\Rightarrow^{+}\omega$ тек қана және тек қана сонда, қашан $(q,\omega,S) \models^{+} (q,\varepsilon,\varepsilon)$. Сондықтан, $\boldsymbol{L}(\boldsymbol{M})\subseteq\boldsymbol{L}(\boldsymbol{G})$.

III.3.1.2-лемма. Егер $\boldsymbol{L}(\boldsymbol{M})$ тілі $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, q_s, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta>$ СА-пен танылса, онда $\boldsymbol{L}(\boldsymbol{G}) = \boldsymbol{L}(\boldsymbol{M})$ болатындай КБ-грамматика $\boldsymbol{G}$ бар болады.

Дәлелдеу. $\boldsymbol{G} = <N, \boldsymbol{T}, \boldsymbol{P}, S>$ грамматикасын былай тұрғызайық:

1) $\boldsymbol{N} = \{(q, A, p)\}$, $q\in\boldsymbol{Q}$, $p\in\boldsymbol{Q}$, $A\in\boldsymbol{H}\cup\{S\}$;

2) $\boldsymbol{T} = \boldsymbol{T}$;

3) $S \rightarrow (q_0, x_0, q) \in \boldsymbol{P}$ кез келген $q\in\boldsymbol{Q}$;

4) q_1, B_1, $B_2,\ldots$, $B_m \in \delta(q, a, A)$ болатындай кез келген $q=q_1,q_2,\ldots,q_{m+1}=p\in\boldsymbol{Q}$, кез келген $a\in(\boldsymbol{T}\cup\{\varepsilon\})$ және кез келген $A,B_1,B_2,\ldots,B_m\in\boldsymbol{H}$ үшін $(q,A,p)\rightarrow a(q_1,B_1,q_2)(q_2,B_2,q_3)\ldots(q_m,B_m,q_{m+1})\in\boldsymbol{P}$ (Егер m=0, онда $q_1 = p(p\varepsilon)\in\delta(q, a, A)$ және $(q, A, p)\rightarrow a\in\boldsymbol{P}$).

III.3.1-теорема. Контексті-бос грамматикалармен танылатын тілдер тобы стекті автоматтармен танылатын тілдер тобымен дәл сәйкес келеді.

Дәлелдеу III.3.1.1-лемма мен III.3.1.2-леммадан шығады.

III.3.1.1-мысал. Мейлі КБ-грамматика $\boldsymbol{G}=<N, \boldsymbol{T}, \boldsymbol{P}, S>$ берілсін, мұнда $N=\{S,A,B\}$, $\boldsymbol{T} =\{a, b\}$ және $\boldsymbol{P}=\{S\rightarrow aB, S\rightarrow bA, A\rightarrow aS, A\rightarrow bAA, A\rightarrow a, B\rightarrow bS, B\rightarrow aBB, B\rightarrow b\}$.

Келесі бейдетерминді стектік автомат тұрғызуға болады

$$\boldsymbol{M} = < \boldsymbol{Q}, \boldsymbol{T}, q_s, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta>,$$

мұнда $\boldsymbol{Q} = \{q_1\}$, $\boldsymbol{T} = \{\boldsymbol{a}, b\}$, $\boldsymbol{F}=\varnothing$, $q_s = q_1$, $\boldsymbol{H} = \{S, A, B\}$, $h_0 = S$, ал ауысу функциясы δ былай беріледі:

$\delta(q_1, a, S) = \{(q_1, B)\}$, себебі $S\rightarrow aB \in \boldsymbol{P}$;

$\delta(q_1, b, S) = \{(q_1, A)\}$, себебі $S\rightarrow bA \in \boldsymbol{P}$;

$\delta(q_1, \boldsymbol{a}, A) = \{(q_1, S), (q_1\varepsilon)\}$, себебі $A\rightarrow aS \in \boldsymbol{P}$ және $A\rightarrow a \in \boldsymbol{P}$;

$\delta(q, b, A) = \{(q_1, AA)\}$, себебі $A \to bAA \in \boldsymbol{P}$;

$\delta(q_1, aB) = \{(q_1, BB)\}$, себебі $B \to aBB \in \boldsymbol{P}$;

$\delta(q_1, bB) = \{(q_1, S), (q_1, \varepsilon)\}$, себебі $B \to bS \in \boldsymbol{P}$ және $B \to b \in \boldsymbol{P}$;

1-теорема бойынша $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}(\boldsymbol{G})$.

III.3.1.3-лемма. Егер $\boldsymbol{L}(\boldsymbol{G})$ тілі $\boldsymbol{G} = \langle N, \boldsymbol{T}, \boldsymbol{P}, S\rangle$ КБ-грамматикасымен тудырылған болса, онда $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}(\boldsymbol{G})$ болатындай кеңейтілген СА-автомат $\boldsymbol{M}$ бар болады.

Дәлелдеу. Айталық $\boldsymbol{M} = \langle \boldsymbol{Q}, \boldsymbol{T}, q_s, \boldsymbol{F}, \boldsymbol{H}, h_0, \vdash, \dashv, \perp, \delta\rangle$ – СА, мұнда $\boldsymbol{Q} = \{q, p\}$, $\boldsymbol{T} = \boldsymbol{T}$, $q_S = q$, $\boldsymbol{F} = \{p\}$, $\boldsymbol{H} = \boldsymbol{N} \cup \boldsymbol{T} \cup \{\$\}$, $h_0 = \$$, ал δ ауысу функциясы былай анықталады:

(1) барлық $t \in \boldsymbol{T}$ үшін $\delta(q, t, \varepsilon) = \{(q, t)\}$ – бұл қадамдарда кіріс символдары стекке көшіріледі);

(2) кейбір бейтерминал $A \in \boldsymbol{N}$ және $\eta \in \boldsymbol{H}^*$ үшін егер $A \to \eta \in \boldsymbol{P}$, онда $\delta(q, \varepsilon, \eta) = \{(q, A)\}$;

(3) $\delta(q, \varepsilon, \$S) = \{(p, \varepsilon)\}$.

M автоматындағы есептеу үрдісі ***M*** автоматының кірісіндегі терминалдық тізбеден басталатын және S бастапқы бейтерминалмен аяқталатын ***G*** грамматикасында оң шығарымды тізбелерді құрудан тұрады. N бойынша индукция бойынша мынаны дәлелдейміз

$$S \Rightarrow^* p\eta A\, y \Rightarrow^n pxy \text{ ілестіреді } (q, xy, \$) \models^* (q, y, \$\eta A) \quad \text{(III.3.1.2)}$$

Индукция базисі. Егер $n=0$, онда автомат ***M*** бір қадам да жасамайды.

Индукциялық қадам. Айталық, n-нен кіші параметрлердің барлық мәндері үшін (III.3.1.2) орындалады. $\eta Ay \Rightarrow p\eta\beta y \Rightarrow^{n-1} pxy$ деп жазуға болады. $\eta\beta$ тізбесі тек терминал символдардан тұрады делік. Сонда $\eta\beta = x$ және $(q, xy, \$) \models^* (q, y, \$\eta\beta) \models (q, y, \$\eta A)$.

Егер $\eta\beta \notin \boldsymbol{T}^*$, онда $\eta\beta = \eta B\mathrm{h}$ деп жазуға болады, мұнда B – ең оң бейтерминал. (III.3.1.2) бойынша $S \Rightarrow^* p\eta B\mathrm{h}y \Rightarrow^{n-1} pxy$ шығарымдарынан $(q, xy\$) \models^* (q, hy, \$\eta B)$ туындайды. Сонымен бірге, мүмкін болатын тактілер тізбегі мынау да болады:

$$(q, hy, \$\eta B) \vdash^* (q, y, \$\eta Bh) \vdash (q, y, \$\eta A).$$

Барлық n үшін (III.3.1.2) ақиқат деп қорытындылаймыз. $(q,\varepsilon,\$S) \vdash (p,\varepsilon,\varepsilon)$ болғандықтан, $\boldsymbol{L}(\boldsymbol{G}) \subseteq \boldsymbol{L}(\boldsymbol{M})$ орындалады.

Кері тұжырымды, яғни $\boldsymbol{L}(\boldsymbol{M}) \subseteq \boldsymbol{L}(\boldsymbol{G})$ екендігін дәлелдеу үшін

$$(q, xy, \$) \vdash^n (q, y, \$\eta A) \text{ ілестіреді } \eta Ay \Rightarrow^* xy \qquad \text{(III.3.1.2)}$$

Бұл тұжырым n бойынша индукциямен дәлелденеді

Индукция базисі. n=0, тұжырым автоматты орындалады.

Индукциялық қадам. Мейлі (III.3.1.2) тұжырымы барлық $n<m$ үшін орындалады делік. Егер $\boldsymbol{M}$ автоматындағы стектің төбесіндегі символ бейтерминал болса, онда соңғы такт δ ауысу функциясы анықтамасының (2) ереже бойынша орындалады. Сондықтан ол былай $(q, xy, \$) \vdash^{m-1} (q, y, \$\eta\beta) \vdash (q, y, \$\eta A)$ болады, мұнда $A \rightarrow \beta \in \boldsymbol{P}$. Егер $\eta\beta$ тізбесі бейтерминалды қамтыса, онда индукциялық болжам бойынша $\eta\beta y \Rightarrow^* xy$. Осыдан $\eta \mathrm{A} y \Rightarrow \eta\beta y \Rightarrow^* xy$, тұжырымдалған осы.

(III.3.1.2) тұжырымының дербес жағдай ретінде мына қатынас $(q, \omega, \$) \vdash^* (q, \varepsilon, \$S)$ ілестіреді $S \Rightarrow^* \omega$. $\boldsymbol{M}$ автоматы ω тізбесін танитын болады тек қана және тек қана сонда, қашан мынау орындалса $(q, \omega, \$) \vdash^* (q, \varepsilon, \$S) \vdash (p, \varepsilon, \varepsilon)$, осыдан мынау шығады $\boldsymbol{L}(\boldsymbol{M}) \subseteq \boldsymbol{L}(\boldsymbol{G})$. Сонымен, $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}(\boldsymbol{G})$.

III.3.1.2-мысал. Грамматика $\boldsymbol{G}$ үшін шықпалы анализатор тұрғызамыз. Мейлі $\boldsymbol{M} = \langle\{q, r\}, T, q, \{r\}, \$, H, \vdash, \dashv, \perp, \delta\rangle$, кеңейтілген стектік автомат, мұнда δ былай анықталады:

(1) $\delta(q, b, \varepsilon) = \{(q, b)\}$ барлық $b \in \{T, +, *, (,)\}$ үшін;

(2) $\delta(q, \varepsilon, T{*}F) = \{(q, E)\}$; $\delta(q, \varepsilon, T) = \{(q, E)\}$;

$\delta(q, \varepsilon, T{*}F) = \{(q, T)\}$; $\delta(q, \varepsilon, F) = \{(q, T)\}$;

$\delta(q, \varepsilon, (E)) = \{(q, F)\}$; $\delta(q, \varepsilon, \boldsymbol{a}) = \{(q, F)\}$;

(3) $\delta(q, \varepsilon, \$E) = \{(r, \varepsilon)\}$.

Кіріс $a{+}T{*}T$ үшін $\boldsymbol{M}$ тактылардың мына тізбегін орындайды:

$$(q, \boldsymbol{a} + T{*}T, \$) \vdash (q, +T{*}T, \$\boldsymbol{a}) \vdash$$
$$(q, +T{*}T, \$F) \vdash (q, +T{*}T, \$T) \vdash$$
$$(q, +T{*}T, \$E) \vdash (q, +T{*}T, \$E^{+}) \vdash$$

$(q, +T{*}T, \$E^{+}a) \models (q, +T{*}T, \$E^{+}F) \models$
$(q,+T{*}T, \$E^{+}T) \models (q, +T{*}T, \$E^{+}T{*}) \models$
$(q,+T{*}T, \$E^{+}T{*}a) \models (q, +T{*}T, \$E^{+}T{*}F) \models$
$(q, +T{*}T, \$E^{+}T) \models (q, +T{*}T, \$E) \models (q, \varepsilon, \varepsilon).$

Кіріс $T^{+}T{*}T$ үшін ***M*** танығышы үшін тактылардың әртүрлі көп тізбектерін жасай алады. Бірақ жазылған тізбек – бастапқы кескінді қорытынды кескінге аударатын жалғыз тізбек.

III.3.1-тапсырма. Келісі КГ үшін СА тұрғызыңыз:

1. $\boldsymbol{G} = <\{\boldsymbol{a}, b, c\}, \{A, B, C, S\}, \boldsymbol{P}, S>$,
   $\boldsymbol{P} = \{S\to \boldsymbol{a}AB, A\to aA|\ bB, B\to ACb|b, C\to bA|Cc\}$
2. $\boldsymbol{G} =<\{x, \boldsymbol{a}, p\}, \{A, D\}, \boldsymbol{P}, \{A\}>$, $\boldsymbol{P} = \{A\to x|Dx, D\to D\boldsymbol{a}|p\}$
3. $\boldsymbol{G} = <\{\boldsymbol{a}, b\}, \{A, B, S\}, \boldsymbol{P}, S>$,
   $\boldsymbol{P} = \{S \to A\boldsymbol{a}|\boldsymbol{a}|b, A\to Ab|\boldsymbol{a}|aB, B\to Ba|b\}$
4. $\boldsymbol{G}=<\{\boldsymbol{a}, b\},\{A,B,C\}, \boldsymbol{P},\{C\}>$, $\boldsymbol{P}=\{A\to b|Cb, B\to Aa|b, C\to Bb|\boldsymbol{a}\}$.
5. $\boldsymbol{G}=<\{x,y,z\},\{X,Y,Z,S\},\boldsymbol{P},S>$, $\boldsymbol{P}=\{S\to Xx|Yy,X\to Zz|z,Y\to B\boldsymbol{a}|b,Z\to y\}$

III.3.1-сұрақтар:

1. ε-ереже деген не?
2. Грейбахтың нормалды пішімі қалай тұрғызылады?
3. Эквивалентті СА тұрғызу үшін КГ қандай пішімде болу керек?
4. КГ тудыратын тілдер СА танитын тілдер тобына жата ма?
5. Берілген КГ ***G*** үшін кеңейтілген СА ***M*** мына $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}(\boldsymbol{G})$ орындалатындай табыла ма?
6. Мына ережермен берілген контекстісіз грамматика бір мәнді ме:
   1) $S\to aSbS, S\to bSaS, F\to\varepsilon$?
   2) $S\to ScS, S\to SdS, S\to aSb, S\to i$?
   3) $S\to ScT, S\to F , F\to FdF, F\to F, F\to aSb, F\to i$?
   4) $S\to F , S\to V , F\to aabF a, F\to aF aa, F\to daaF db, F\to c,$
   $V\to aV a, V\to bV b, V\to dV d, V\to aca, V\to bcb, V\to dcd$?
   5) $S\to FV , S\to V F, F\to aV a, F\to bV b, V\to aF a, V\to bF b, F\to\varepsilon$?
   6) $S\to FV , S\to V F, F\to aV a, F\to bV b, V\to aF a, V\to bF b, F\to c$?

III.3.2. Контекстісіз грамматикалардың алгоритмдік проблемалары

Бұл параграфта контекстісіз тілдердің алгоритмдік проблемалары талқыланылады, мысаладар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1-4, 6-9, 11-13, 15-21, 24, 27-34].

1. Алгоритмдік шешілімді проблемалар

Контекстісіз тілдер үшін бостық проблемасы және ақырсызсыздық проблемасы алгоритмдік шешілімді.

Бостық проблемасының шешілімдігі контекстісіз грамматикаларды келтіру алгоритмінен шығады, атап айтқанда: $\boldsymbol{L}(\boldsymbol{G})$ тілі бос емес сонда және тек қана сонда, қашан $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ грамматикасында бастапқы бейтерминал S туындайтын болса, яғни $S\rightarrow\tau$, $\tau\in\boldsymbol{T}^+$ түрінде ең болмаса бір ереже бар болса.

Детерминді стекті автоматтардың эквиваленттілік проблемасы шешілімді, яғни келесі теорема ақиқат.

III.3.2.1-теорема. Қандай да екі детерминді стектік автомат $\boldsymbol{M_1}$ және $\boldsymbol{M_2}$ бойынша $\boldsymbol{L}(\boldsymbol{M_1}) = \boldsymbol{L}(\boldsymbol{M_2})$ рас немесе жоқ болады.

Контекстісіз грамматикалардың ақырсыздық проблемасы алгоритмді шешілімді, яғни келесі теорема ақиқат.

III.3.2.2-теорема. Қандай да бір контекстісіз грамматика $\boldsymbol{G}$ бойынша $\boldsymbol{L}(\boldsymbol{G})$ ақырсыз немесе ақырлы болады.

Констекстісіз тілдердің тұйықтылық қасиеттері келесі III.3.2.3, III.3.2.4, III.3.2.5 теоремалар арқылы дәлелденеді:

III.3.2.3-теорема. Егер $\boldsymbol{L}$ – контекстісіз тіл болса, онда $\boldsymbol{L^*}$ тілі де контекстісіз тіл болады.

Дәлелдеу. Мейлі контекстісіз $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ грамматикасы $\boldsymbol{L}$ тілін тудырсын деп алаыйқ. Сонда $\boldsymbol{L^*}$ тілін контекстісіз грамматика $\boldsymbol{G_i} = <\boldsymbol{N} \cup \{R\}, \boldsymbol{T}, \boldsymbol{P} \cup \{R\rightarrow SR, R\rightarrow\varepsilon\}, S>$ тудырады, мұнда $S\notin\boldsymbol{N}\cup\boldsymbol{T}$.

III.3.2.4-теорема. Егер $\boldsymbol{L}_1$ және $\boldsymbol{L}_2$ – $\boldsymbol{T}$ әліпбиіндегі контекстісіз тілдер болса, онда $\boldsymbol{L}_1 \cdot \boldsymbol{L}_2$ тілі де контекстісіз.

Дәлелдеу. $\boldsymbol{L}_1$ тілін контекстісіз грамматика $\boldsymbol{G_1} = <\boldsymbol{N}_1, \boldsymbol{T}, \boldsymbol{P}_1, S_1>$ тудырсын, ал $\boldsymbol{L}_2$ тілін контекстісіз грамматика $\boldsymbol{G_2} = <\boldsymbol{N}_2, \boldsymbol{T}, \boldsymbol{P}_2, S_2>$ тудырсын, мұнда $\boldsymbol{N}_1 \cap \boldsymbol{N}_2 = \emptyset$. Сонда $\boldsymbol{L}_1 \cdot \boldsymbol{L}_2$ тілін мына грамматика $\boldsymbol{G_k} = <\boldsymbol{N}_1 \cup \boldsymbol{N}_2 \cup \{R\}, \boldsymbol{T}, P_1 \cup P_2 \cup \{R \to S_1 S_2\}, R>$ тудырады, $R \notin \boldsymbol{N}_1 \cup \boldsymbol{N}_2 \cup \boldsymbol{T}$.

III.3.2.5-теорема. Егер $\boldsymbol{L}_1$ және $\boldsymbol{L}_2$ — $\boldsymbol{T}$ әліпбиіндегі контекстісіз тіл болса, онда $\boldsymbol{L}_1 \cup \boldsymbol{L}_2$ де контекстісіз тіл.

Дәлелдеу. Мейлі $\boldsymbol{L_1}$ тілін контекстісіз грамматика $\boldsymbol{G_1} = <\boldsymbol{N_1}, \boldsymbol{T}, \boldsymbol{P_1}, S_1>$ тудырсын, ал $\boldsymbol{L_2}$ тілін контекстісіз грамматика $\boldsymbol{G_2} = <\boldsymbol{N_2}, \boldsymbol{T}, \boldsymbol{P_2}, S_2>$ тудырсын, мұнда $\boldsymbol{N_1} \cap \boldsymbol{N_2} = \emptyset$. Сонда $\boldsymbol{L_1} \cup \boldsymbol{L_2}$ тілін мына грамматика $G_k = <\boldsymbol{N_1} \cup \boldsymbol{N_2} \cup \{R\}, \boldsymbol{T}, \boldsymbol{P_1} \cup \boldsymbol{P_2} \cup \{R \to S_1, R \to S_2\}, R>$ тудырады, мұнда $R \notin \boldsymbol{N_1} \cup \boldsymbol{N_2} \cup \boldsymbol{T}$.

2. Алгоритмдік шешілімсіз проблемалар

Контекстісіз грамматикалар үшін бірнеше алгоритмдік шешілілімсіз проблемалар бар:

1. *Бірмәнділік проблемасы*: КГ-ның бір мәнділігін білдіретін алгоритмнің бар болуы.

2. *Тұйықтылық проблемасы*: тілдің қиылысуға (толықтыруға) байланысты тұйықтылығын білдіретін, яғни екі контекстісіз тілдің қиылысуы (контекстісіз тілдің толықтыруы) контекстісіз тіл болатындығын білдіретін алгоритмнің бар болуы.

3. *Толықтырудың бостығы* (*тұрпайылығы*) *проблемасы*: тіл толықтыруының бостығын білдіретін алгоритмнің бар болуы, яғни $\boldsymbol{T}^* - \boldsymbol{L}(\boldsymbol{G}) = \emptyset$?

4. *Қиылысудың бостығы проблемасы*: екі КГ $\boldsymbol{G_1}$ және $\boldsymbol{G_2}$ арқылы туындаған екі тіл қиылысуының бостығын білдіретін алгоритмнің бар болуы, яғни $\boldsymbol{L}(\boldsymbol{G_1}) \cap \boldsymbol{L}(\boldsymbol{G_2}) = \emptyset$?

5. *Эквиваленттік проблемасы*: екі $\boldsymbol{G_1}$ және $\boldsymbol{G_2}$ КГ бір тілді тудыратындығын білдіретін алгоритмнің бар болуы, яғни $\boldsymbol{L}(\boldsymbol{G_1}) = \boldsymbol{L}(\boldsymbol{G_2})$? Сондай-ақ, егер грамматикалардың біреуі оң сызықты грамматика болса да бұл проблема шешілімсіз екендігі дәлелденген.

ІІІ.3.2-тапсырмалар:

1. $L_1 \cup L_2$ тілі үшін контекстісіз грамматика табыңыздар, мұнда L_1 тілі $A \to a$, $A \to bA$, $A \to cAA$ грамматикасымен, ал L_2 тілі $A \to aB$, $A \to cB$, $B \to aA$, $B \to bA$, $B \to \varepsilon$ грамматикасымен туындайды.

2. Констекстісіз грамматика $N = \{A, B, C\}$, $S = A$, $T = \{a, b, c\}$, $P = \{A \to aB,\ B \to aB,\ B \to b,\ B \to bD,\ D \to d,\ D \to dD,\ A \to aD,\ A \to a\}$> мен эквиваленттi стектi автомат тұрғызыңыз.

3. Констекстісіз грамматика $G=\{N,T,P,S\}$ үшін $L=\{A \in N : A \Rightarrow \varepsilon\}$ жиынын тұрғызатын алгоритм жазыңыз.

4. $L= \{i+n*(i+n)\}$ тілін констекстісіз грамматика $N=\{E, F, H\}$, $T =\{i, n, (,), +,*\}$, $S=E$, $P=\{E \to E+F, F \to F*H,\ H \to I,\ E \to F,\ F \to H,\ H \to n, H \to (E)\}$ мен эквиваленттi стектi автомат тұрғызыңыз.

5. Кез келген контекстісіз грамматика G үшін $L(G)$ бос екендігін анықтайтын алгоритм құрыңыз.

6. Мына $S \to AB$, $A \to Aa|bB$, $B \to a|Sb$ ережелермен туындайтын контекстісіз тілді жазыңыз.

7. $(ab)^n(ca)^*b^n+2(abc)^*$, $n>0$ тілін тудыратын контекстісіз грамматиканы тұрғызыңыз.

ІІІ.3.2-сұрақтар:

1. Мына $S \to T|T+S|T\text{-}S$, $T \to F|F*T$, $F \to a|b$ ежелер қандай грамматикаға жатады?

2. Мына $S \to A+B|B+A$, $A \to a$, $B \to b$ ежелер қандай грамматикаға жатады?

3. $G = \langle\{A, B\}, \{0,1\}, \{S \to 0A|1S|\varepsilon,\ A \to 0B|1A,\ B \to 0S|1B\}, S\rangle$ грамматикасына эквиваленттi автомат қалай тұрғызылады?

4. Мына ережелермен $E \to E+E|E*E|(E)|i$ берілген грамматикаға эквиваленттi стектiк мавтомат тұрғызуға бола ма?

5. $S \to AB$, $A \to Aa|bB$, $B \to a|\ Sb$ ережелерімен берілген грамматикаға эквиваленттi стектiк автомат тұрғызуға бола ма?

6. $G=\langle\{A,B\}, \{0,1\}, \{X \to 0A|1S|\varepsilon\}, \{A \to 0B|1A\}, \{B \to 0S|1B\}, S\rangle$ грамматикасына эквиваленттi стектiк автомат қалай тұрғызылады?

7. Мына ережелері $E \to E+E|E*E|(E)|i$ бар грамматикаға эквиваленттi стектiк автомат тұрғызуға бола ма?

IV. КОНТЕКСТІ ТІЛДЕР

IV.1. Контексті тілдерді тудырушы механизмдер

IV.1.1. Контексті грамматикалар

Бұл бөлімде контексті тілдер қарастырылады, оларды тудыру және тану механизмдері беріледі, осы механизмдердің эквиваленттілігі көрсетіледі және контексті тілдердің алгоритмдік проблемалары талқыланады, мысаладар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1-4, 6-9, 11-13, 15-21, 24, 27-34].

Контексті тілдердің тудырушы механизмдері ретінде 1-ші типтегі грамматикалар болады және олар контексті-тәуелді (контексті) грамматикалар деп аталады. Оларға тікелей құраушылар грамматикалары және қысқармайтын грамматикалар жатады. Контексті тілдердің таныған механизмдері ретінде сызықты шенеуленген автоматтар болады.

IV.1.1.1-анықтама. Контексті грамматика $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ тікелей құраушылар грамматикасы болады, егер оның әрбір ережесінің түрі мынадай $\eta A\theta\to\eta\alpha\theta\in\boldsymbol{P}$, мұнда $A\in\boldsymbol{N}$; $\eta, \theta\in(\boldsymbol{T}\cup\boldsymbol{N})^*$; $\alpha\in(\boldsymbol{T}\cup\boldsymbol{N})^+$ болса.

IV.1.1.1-мысал. Мейлі грамматика $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ берілсін, $\boldsymbol{N} =\{S, A, B,\}$, $\boldsymbol{T} = \{a, b\}$ және ережелер жиыны $\boldsymbol{P}$ мыналар болсын:

$$S\to ASBA,\ S\to AbA,\ A\to a,\ bB\to bb,\ AB\to BA$$

Бұл $\boldsymbol{G}$ грамматикасы тікелей құраушылар грамматикасы болады.

IV.1.1.2-анықтама. Контексті грамматика $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ қысқармайтын (*noncontracting*) грамматикасы болады, егер оның әрбір $\alpha\to\beta\in\boldsymbol{P}$ түріндегі ережесіне мынадай шектеулік қойылса $|\alpha| \leq |\beta|$, мұнда $\alpha, \beta\in(\boldsymbol{T}\cup\boldsymbol{N})^+$.

IV.1.1.2-мысал. Мына ережелермен грамматика берілсін:

$$\begin{array}{ll} S\to TS, & A\to a, \\ S\to US, & TA\to AAT, \\ S\to b, & UAb\to b, \\ Tb\to Ab, & UAAA\to AAU, \end{array}$$

Бұл грамматика қысқармайтын грамматика емес, себебі соңғы үш ереже қойылған талаптағы түрде емес.

Мейлі контексті грамматика $\boldsymbol{G}=\langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S\rangle$ берілсін, сонда:

1) Егер $\boldsymbol{G}$ оң жағы бос ережені, яғни мына $A\to\varepsilon$ түрдегі ережені қамтымаса, онда ол қысқармайтын грамматика болады.

2) Егер $\boldsymbol{G}$ оң жағы бос ережені, яғни мына $A\to\varepsilon$ түрдегі ережені қамтыса, онда ол қысқаратын грамматика болады. Мұнда шығарылатын ережелер тізбесінің ұзындығы k қадамынан $(k+1)$ қадамына өткен кезде қысқартылады.

IV.1.1.3-анықтама. Контексті грамматикалармен туындаған тілдер контексті тілдер деп аталады.

IV.1.1.3-мысал. Мейлі КГ $\boldsymbol{G}=\langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S\rangle$ берілсін, мұнда:

1. $\boldsymbol{P}=\{S\to aSBC|abc, cB\to BC, bB\to bb, bC\to bc, cC\to cc\}$, $\boldsymbol{N}=\{B, C, S\}$, $\boldsymbol{T}=\{a, b, c\}$. Сонда $\boldsymbol{G}$ грамматикасы келесі шығарым $S\to aSBC\to aabcBC\to aabBCC\to aabbCC\to aabbcC\to aabbcc$ арқылы $\boldsymbol{L}(\boldsymbol{G})\rightleftharpoons\{a^n b^n c^n : n\geq1\}$ контексті тілін тудырады.

2. $\boldsymbol{P}=\{A\to aAB, A\to Abc, bB\to bbc, cB\to Bc\}$, $N=\{A,B,C\}$, $\boldsymbol{T}=\{a, b, c\}$ болсын. Сонда $\boldsymbol{G}$ грамматикасы келесі шығарым тізбегі $A\to aAB\to aabcB\to aabBc\to aabbcc$ арқылы $\boldsymbol{L}(\boldsymbol{G})\rightleftharpoons\{a^n b^n c^n : n\geq1\}$ контексті тілін тудырады.

3. $\boldsymbol{P}=\{S\to SbaS, S\to ab, aS\to b\}$, $\boldsymbol{N}=\{S\}$, $\boldsymbol{T}=\{a, b\}$ болсын. Сонда $\boldsymbol{G}$ грамматикасы келесі шығарымдар тізбегі $S\to SbaS\to abbaS\to abbb$, $S\to SbaS\to abbaab$ арқылы мынадай $\boldsymbol{L}(\boldsymbol{G})\rightleftharpoons\{a^n b^n c^n : n\geq1\}$ контексті тілін тудырады.

4. $\boldsymbol{P}=\{S\to ABA, B\to ABCA, B\to b, bC\to bb, AC\to DC, DC\to DA, DA\to CA, A\to a\}$, $\boldsymbol{N}=\{A, B, C, D\}$, $\boldsymbol{T}=\{a, b, c\}$. Сонда мына туындаған $\boldsymbol{L}(\boldsymbol{G})\rightleftharpoons\{a^n b^n c^n : n\geq1\}$ тілі контексті болады.

Бос тізбесіз тілді тудыратын қайсы бір контексті грамматика үшін эквивалентті қысқармайтын граматика тұрғызуға болатындығын көрсетейік.

Мейлі контексті грамматика $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ берілсін. Грамматиканың бос тізбе шығарылатын барлық бейтерминал символдарының жиынын келесі

$$W_1 = \{A: A\to\varepsilon \in \boldsymbol{P}\},$$
$$W_{n+1} = W_n \cup \{B: B\to\phi \in \boldsymbol{P}, \phi \in W_n{}^*\}$$

жиындарын ерекшелеп тұрғызайық.

Сонан кейін эквивалентті грамматиканың ережелерін екі этапта табамыз:

а) алғашқы грамматикасының $\boldsymbol{P}$ жиынынан оң жағы бос ережелерді жоямыз: $\boldsymbol{P}_1 = \boldsymbol{P} \setminus \{A\to\varepsilon: A\to\varepsilon \in \boldsymbol{P}\}$;

б) алғашқы грамматиканың әрбір $A\to\phi\in\boldsymbol{P}$ ережесінен W_n жиынына мына $\boldsymbol{P}_2 = \{A\to\phi' \mid A\to\phi\in\boldsymbol{P}_1;\ \phi=\phi_1B\phi_2 | B\in W_n;\ \phi'=\phi_1B\phi_2\}$ ережесі арқылы енген бейтерминалды жойғаннан кейін жаңа $A\to\phi'$ ережесін аламыз.

W_n жиынына жататын әрбір бейтерминал үшін б) пунктісін қайталап, $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}_2, S>$ грамматикасына эквивалентті грамматика аламыз.

Сонымен, келесі теорема орын алады.

IV.1.1.1-теорема. Әрбір контексті грамматика қысқармайтын грамматика болады.

IV.1.1.4-мысал. Мейлі $\boldsymbol{G}$ грамматика мына ережелерімен берілсін:

$$S\to AbA|cAb|Bb$$
$$A\to aAb|\varepsilon$$
$$B\to AA|a$$

Сонда мыналарды тұрғызу керек:

1) өзінен бос тізбе ε шығарылған бейтерминалдар жиынын;

2) алғашқы грамматикаға эквивалентті грамматиканы.

Грамматикадағы өзінен бос тізбе шығатын барлық бейтерминалдардың жиынын тұрғызу үшін келесі жиындарды ерекшелейік:

$$W_1 = \{A : A\to\varepsilon \in P\};$$
$$W_{m+1} = W_m \cup \{B : B\to\phi \in P, \phi \in W_m{}^*\}.$$

Біз $A \to \varepsilon \in \boldsymbol{P}$ ережесіне ие болу себебті, A бейтерминалын қамтитын $W_1 = \{A\}$ жиынын тұрғызуға болады.

W_2 жиынын тұрғызайық. A бейтерминалымен B бейтерминалы байланысқан, яғни $B \to AA$ және $A \in W_1$ ережелері бар. Сондықтан $W_2 = \{A, B\}$.

$W_3 = W_2$, себебі $W_3 = \{B{:}B \to \phi \in \boldsymbol{P}, \phi \in W_m{}^*\}$ жиыны бос болады.

Оң жағында бос тізбені қамтитын ережені жойып, мына түрдегі қысқармайтын $\boldsymbol{G_1}$ грамматикасын аламыз:

$$S \to AbA|cAb|Bb|bA|Ab|cb|b;$$
$$A \to aAb|ab;$$
$$B \to AA|A|a.$$

IV.1.1.4-анықтама. Контексті грамматика $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ Куродының нормалды пішімінде болады, егер оның әрбір шығарым ережесі мына пішімдердің бірінде бейнеленсе: $A \to BB'$, $AB \to A'B$, $A \to a$, $A \to AB'$, $A \to A'$, мұнда $A, A', B, B' \in \boldsymbol{N}$, $a \in \boldsymbol{T}$.

Кез келген берілген контексті грамматика $\boldsymbol{G_1}$ бойынша оған эквивалентті контексті $\boldsymbol{G_2}$ грамматикасын, олардың тудыратын тілдері тең болатындай, Куродының нормалды пішімінде тұрғызуға болады, яғни $\boldsymbol{L}(\boldsymbol{G_2}) = \boldsymbol{L}(\boldsymbol{G_1})$.

IV.1.1.4-ескертпе. Берілген грамматикада бір ережелер бір орында, бірақ басқа ретпен қолданылатын бірнеше эквивалентті шығарым болады. Дегенмен де, кез келген түрдегі грамматика үшін екі шығарымның эквивалентті екенін дәлелдеу өте қиын.

IV.1.1.5-мысалдар:

1. Мейлі контексті грамматика $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ берілсін, мұнда $\boldsymbol{P} = \{S \to aSBC|abc, cB \to BC, bB \to bb, bC \to bc, cC \to cc\}$, $\boldsymbol{N} = \{B, C, S\}$, $\boldsymbol{T} = \{a, b, c\}$. Сонда грамматика $\boldsymbol{G}$ мына контексті $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{a^n b^n c^n \mid n \geq 1\}$ тілін келесі шығарым арқылы тудырады:

$$S \to aSBC \to aabcBC \to aabBCC \to aabbCC \to aabbcC \to aabbcc.$$

2. Мейлі контексті грамматика $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ берілсін, мұнда $\boldsymbol{P} = \{A \to aAB, A \to Abc, bB \to bbc, cB \to Bc\}$, $\boldsymbol{N} = \{A, B, C\}$, $\boldsymbol{T} = \{a, b, c\}$.

Сонда $\boldsymbol{G}$ грамматикасы контексті $\boldsymbol{L}(\boldsymbol{G})\rightleftharpoons\{a^n b^n c^n : n\geq 1\}$ тілін келесі шығарым арқылы тудырады: $A\to aAB\to aabcB\to aabBc\to aabbcc$.

3. Мейлі контексті грамматика $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ берілсін, мұнда $\boldsymbol{N} = \{S\}$, $\boldsymbol{T} = \{a, b\}$, ал $\boldsymbol{P}$ мына ережелерден тұрады:

$S\to SbaS$

$S\to ab$

$aS\to$

Сонда $\boldsymbol{G}$ грамматикасы мына контексті $\boldsymbol{L}(\boldsymbol{G}) = \{ab^{n+1}\}$ тілін келесі шығарымдар арқылы тудырады:

$S\to SbaS\to abbaS\to abbb$

$S\to SbaS\to abbaab$

IV.1.1-тапсырмалар:

1. Ережелері мынадай грамматикамен туындайтын тілді табу:

$S\to A1$

$A\to 1A1|0A0|B$

$B\to 021|120C$

$C1\to 0C$

$C0\to 1$

$R1\to 11$

2. Мына тілді тудыратын қысқармайтын грмматиканы табу:

$\{\omega\in\{a,b,c,\}^+ : |\omega|_a = |\omega|_b = |\omega|_c\}$;

$\{\omega\in\{a,b,c,\}^* : |\omega|_a < |\omega|_b < |\omega|_c\}$;

$\{\omega c\omega : \omega\in\{a,b,c,\}^*\}$;

$\{c^n dc^n dc^n : n\geq 0\}$;

$\{a^n b^n c^n : n\geq 1\}$.

3. Берілген $\boldsymbol{L} = \{a^n b^n c^n : n \geq 1\}$ тілін мына ережелері бар грамматика тудыратынын дәлелдеу:

$S\to aSBC|abC$

$CB\to BC$

$bB\to bb$

$bC\to bc$

$cC\to cc$

4. Мына ережелері бар грамматика тудыратын тілді табу:

$$S \to 0A1$$
$$0A \to 00A1$$
$$A \to \varepsilon$$

5. $\boldsymbol{L} = \{\omega \mid \omega \in \{a, b, c\}$ және ω тізбесінде a әріпінің саны b әріпінің санына және c әріпінің санына тең$\}$ тілін тудыратын контексті грамматиканы тұрғызу.

Ережелері мынадай контексті грамматика тудыратын тілді табу: $S \to CD$, $C \to aCA$, $AD \to aD$, $BD \to bD$, $Aa \to aA$, $Ab \to bA$, $Ba \to aB$, $Bb \to bB$, $C \to e$, $D \to e$.

IV.1.1-сұрақтар:

1. Мына ережелері бар грамматика қай топқа жатады?

a) $S \to a|Ba$, $B \to Bb|b$

b) $S \to Ab$, $A \to Aa|ba$

c) $S \to 0A1|01$, $0A \to 00A1$, $A \to 01$

d) $S \to AB$, $AB \to BA$, $A \to a$, $B \to b$

e) $S \to aAb$, $aA \to aaAb$, $A \to \varepsilon$

f) $S \to AB|ABS$, $AB \to BA$, $BA \to AB$, $A \to a$, $B \to b$

h) $S \to abC|aB$, $B \to bc$, $bC \to bc$

2. Мейлі жалғыз S бейтерминалы, үш 1, 2, 3 терминалы және мынадай ережелері бар грамматика берілсін:

$$S \to 0,\ S \to S1,\ S \to SS2,\ S \to SSS3.$$

Солдан оңға қарай оқи отырып (бір әріпті оқыған кезде әрекет саны шектеулі болу керек) сөздің грамматикада шығарулуын қалай тексереміз?

3. Мына тіл қандай топқа жатады:

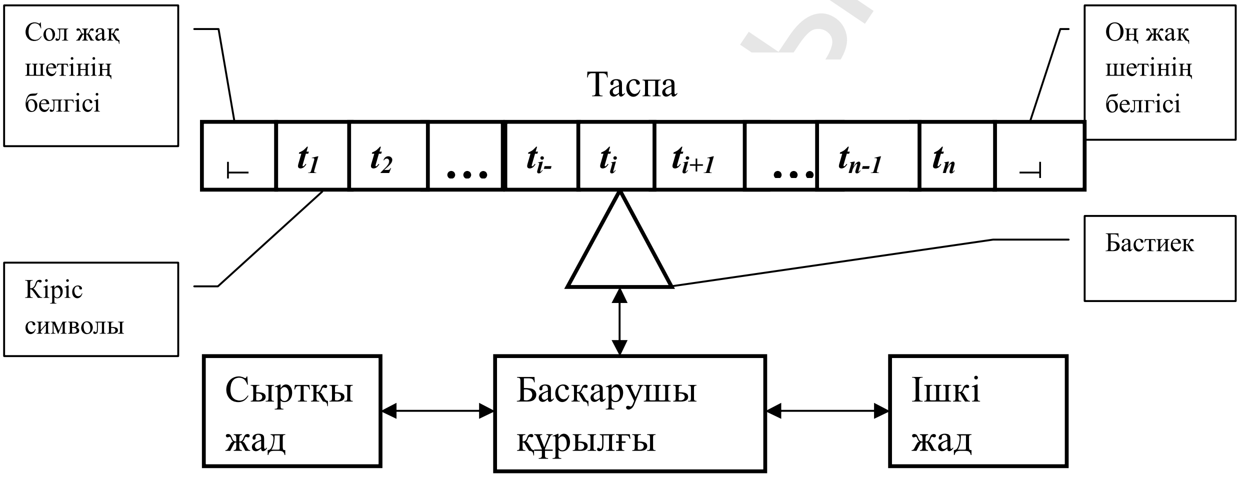
$$\boldsymbol{L} = \{1^{3n+2}\, 0^n : n \geq 0\}?$$
$$\boldsymbol{L} = \{a^m b^n a^m b^n : m,n \geq 1\};$$
$$\boldsymbol{L} = \{\omega\omega : \omega \in \{a, b\}\};$$
$$\boldsymbol{L} = \{0^i 1^j : i \neq j;\ i, j \geq 0\};$$
$$\boldsymbol{L}(\boldsymbol{G}) = \{a^n b^n c^n : n \geq 1\};$$
$$\boldsymbol{L}(\boldsymbol{G}) = \{ab^{n+1}\}.$$

IV.2. Контексті тілдерді танушы механизмдер

IV.2.1. Сызықты – шенеуленген автоматтар

Бұл бөлімде сызықты-шенеуленген автоматтардың (США) құрамы, құрылымы және функциясы қарастырылады, олардың формалды анықтамасы, кескіні, тактысы және тілдерді тану әдістері көрсетіледі, мысаладар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1-4,6-9,11-13,15-21,24-32].

Сызықты-шенеуленген автоматтар (l*in*ear-bou*n*ded automato*n*) бейдетерминді ақырлы автоматтарға ұқсайды, бірақ олардың бастиегі кіріс таспасына символдарды жаза алады және екі жаққа бір ұяшық оңға немесе бір ұяшық солға жылжи алады. Сызықты-шенеуленген автоматтың құрылымы IV.2.1-суретте көрсетілген.



IV.2.1-сурет. Сызықты шенеуленген автомат

IV.2.1.1-анықтама. Сызықты шенеуленген автомат мына ондық $\boldsymbol{M} = < \boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{U}, q_s, \boldsymbol{F}, \vdash, \dashv, \leftarrow, \rightarrow, \delta >$ болады, мұнда:

$\boldsymbol{Q}$ – басқарушы құрылғы күйлерінің ақырлы жиыны;

$\boldsymbol{U}$ – таспалық символдардың ақырлы жиыны, $\boldsymbol{Q} \cap \boldsymbol{U} = \varnothing$;

$\boldsymbol{T}$ – кіріс символдарының ақырлы жиыны, $\boldsymbol{T} \subseteq \boldsymbol{U}$;

q_s – басқарушы құрылғының алғашқы күйі, $q_s \in \boldsymbol{Q}$;

$\boldsymbol{F}$ – басқарушы құрылғының соңғы күйілерінің жиыны, $\boldsymbol{F} \subseteq \boldsymbol{Q}$;

$\vdash, \dashv$ – таспа басы мен соңын белгілегіштер, $\vdash, \dashv \in \boldsymbol{U}$, $\vdash, \dashv \notin \boldsymbol{T}$;

$\leftarrow$ – бастиектің солға жылжитындығын көрсететін арнаулы символ;

$\rightarrow\!\!|$ – бастиектің оңға жылжитындығын көрсететін арнаулы символ;

δ: $\boldsymbol{Q}\times\boldsymbol{T}\rightarrow\boldsymbol{Q}\times\boldsymbol{T}\times\{|\!\!\leftarrow, \rightarrow\!\!|\}$ – детерминді сызықты шенеуленген автомат үшін.

δ: $\boldsymbol{Q}\times\boldsymbol{T}\rightarrow\vartheta(\boldsymbol{Q}\times\boldsymbol{T}\times\{|\!\!\leftarrow, \rightarrow\!\!|\})$ – бейдетерминді сызықты шенеуленген автомат үшін, мұндағы $\vartheta(\boldsymbol{Q}\times\boldsymbol{T}\times\{|\!\!\leftarrow, \rightarrow\!\!|\})$ жиыны $\boldsymbol{Q}\times\boldsymbol{T}\times\{|\!\!\leftarrow, \rightarrow\!\!|\}$ жиынының барлық ішжиындарының жиыны.

Осындан, егер кез келген $q\in\boldsymbol{Q}$ және $t\in\boldsymbol{U}$ үшін $\delta(q, t)$ бейнелеуінде бірден көп емес элемент болса, онда сызықты шенеуленген автомат *детерминді* болатындығы көрінеді.

IV.2.1.2-анықтама. Сызықты шенеуленген автоматтың кескіні деп мына (q, χ, i) үштікті айтады, мұнда:

(1) $q\in\boldsymbol{Q}$ – басқарушы құрылғының ағымдағы күйі;

(2) $\chi = \vdash t_1 t_2 \ldots t_n \dashv$ – таспадағы тізбе $t_j\in\boldsymbol{T}$, $1\leq j\leq n$, егер $\chi = \vdash\varepsilon\dashv$, онда кіріс таспасы толық оқылды деп есептелінеді;

(3) $i\in\boldsymbol{N}$ – бастиектің τ тізбесінің сол жақ шетінен ара қашықтығын көрсететін натурал сан.

Егер $(p, t, |\!\!\leftarrow)\in\delta(q, t_i)$, $t\in\boldsymbol{T}$, $i>1$, онда $\boldsymbol{M}$ автоматының қадамы мынадай бинарлы қатынас арқылы жазылады:

$$(q, \vdash t_1 t_2 \ldots t_{i-1}\, t_i\, t_{i+1} \ldots t_n \dashv, i) \models_{\boldsymbol{M}} (p, \vdash t_1 t_2 \ldots t_{i-1}\, t\, t_{i+1} \ldots t_n \dashv, i-1).$$

Егер $(p, t, \rightarrow\!\!|)\in\delta(q, t_i)$, $t\in\boldsymbol{T}$, $i<n$, онда $\boldsymbol{M}$ автоматының қадамы мынадай бинарлы қатынас арқылы жазылады:

$$(q, \vdash t_1 t_2 \ldots t_{i-1}\, t_i\, t_{i+1} \ldots t_n \dashv, i) \models_{\boldsymbol{M}} (p, \vdash t_1 t_2 \ldots t_{i-1}\, t\, t_{i+1} \ldots t_n \dashv, i+1).$$

$\models_{\boldsymbol{M}}$ қатынасы $\boldsymbol{M}$ автоматының q күйі p күйіне ауысып және бастиегі i-ші ұяшығындағы t_i символының орнына t символын жазып, оның оң жағындағы бір ұяшыққа, кіріс тізбесі алғаш орналасқан аумақтан шықпайтындай болып, солға ($|\!\!\leftarrow$) немесе оңға ($\rightarrow\!\!|$) жылжығанын көрсетеді. Мұнда, егер автомат белгілі болса, онда $\models_{\boldsymbol{M}}$ қатынасындағы $\boldsymbol{M}$ әрпін жазбауға болады.

Әдеттегідей, $\models$ қатынасының k-шы *дәрежесін* $\models^k$ арқылы, ал оның рефлексивтік және транзитивтік тұйықталуын $\models^*$ арқылы белгілейміз.

США $\boldsymbol{M}$ танитын $\boldsymbol{L}(\boldsymbol{M})$ тілін былай анықтауға болады:

$$\boldsymbol{L}(\boldsymbol{M}) \rightleftharpoons \{\tau{:}\tau\in \boldsymbol{T}^*\&(q_s,\vdash\tau\dashv,1) \models^* (q,\vdash\alpha\dashv,i)\&q\in \boldsymbol{F}\&\alpha\in \boldsymbol{T}^*\&1\le i\le n, n=|\tau|+2\}$$

Жалпы айтқанда, тіл танылады деп аталады, егер ақырлы сан қадам ішінде кез келген тізбенің тілге жататыны туралы сұраққа жауап беретін алгоритм болса.

IV.2.1-ескертпе. Егер моделі сызықты-шенеуленген автомат болатын программа жазу керек болса, онда ол бейдетерминді автомат болады дем есептеу керек.

IV.2.1-мысал. Контексті $\boldsymbol{L}(\boldsymbol{G}_1)=\{a^n b^n a^n\}$ тілін тану үшін сызықты-шенеуленген автомат (США) тұрғызу, егер контексті грамматика $\boldsymbol{G}_1$ мына шығарым ережелерін қамтитын болса $S\to ACA$, $C\to ACBA$, $C\to b$, $AB\to KB$, $KB\to KA$, $KA\to BA$, $bB\to bb$, $A\to a$.

США жұмысын осы грамматикада төмендегідей туындайтын *aaabbbaaa* терминалды тізбесін талдау кезінде қарау керек: $S\to ACA\to AACBAA\to AAACBABAA\to AAACBKBAA\to AAACBKAAA\to AAACBBAAA\to AAAbBBAAA\to AAAbbBAAA\to AAAbbbAAA\to aAAbbbAAA\to aaAbbbAAA\to aaabbbAAA\to aaabbbaAA\to aaabbbaaA\to aaabbbaaa$.

Мына $a^3b^3a^3\in\boldsymbol{L}(\boldsymbol{G}_1)$ тізбесінің келесі ережелері бар эквивалентті грамматикада шығарымы мүмкін болатыны ескеру керек $S\to aCa$, $C\to aCBa$, $C\to b$, $aB\to Ba$ $bB\to bb$.

Мына тізбелерді таныған кезде США кескіндерін бейнелейміз:

$a_0q_1\ aaabbbaaa\ a_0 \models a_0D\ q_1aabbbaaa\ a_0C\ a_0Daabbbaaaq_1 \models\ldots\models$
$q_2DaabbbaaaA \models\ldots\models Daabq_2bbaaA \models\ldots\models q_2DaabbBaaA \models\ldots\models$
$Daaq_2bbBaaA \models\ldots\models q_2DaabBBaaA \models\ldots\models Daaq_2bBBaaA \models\ldots\models$
$q_2DaaCBBaaA \models\ldots\models DaaCBq_2BaaA \models\ldots\models q_2DaaCBaBaA \models\ldots\models$
$Daq_2aCBaBaA \models\ldots\models q_2DaC\text{ннн}BaA \models\ldots\models Dq_2aC\text{ннн}BaA \models\ldots\models$
$q_2Dc\text{нннннн}A \models q_2{}^* \models q_2aC\text{нннннн}a \models\ldots\models q_2S\text{ннннннн} \models\ldots\models S\text{ннннннн}q_z$

IV.2.2-ескертпе. Егер талданатын тізбе $\boldsymbol{L}(\boldsymbol{G}_1)=\{a^n b^n a^n\}$ тіліне енбесе, онда США талдаудың кез келген баламасында тұйықталу жағдаятына көшеді. Мысалы, *abaa* тізбесінің талдауының бір варианты келесідегідей: $q_1abaa \models\ldots\models q_2DbaA \models\ldots\models Dq_2baA \models\ldots\models q_2DcaA \models\ldots\models q_2aCaa \models q_2S\text{нн}a \models\ldots\models S\text{нн}q_2a$ – тұйық.

Инициалды $\boldsymbol{M}$=<$\boldsymbol{Q}$, $\boldsymbol{T}$, $\boldsymbol{U}$, q_s , $\boldsymbol{F}$, $\vdash$,$\dashv$, $\leftarrowtail$, $\rightarrowtail$, δ> США-мен кіріс тізбесін тану алгоритмі δ: $\boldsymbol{Q}\times\boldsymbol{U}\rightarrow\boldsymbol{Q}\times\boldsymbol{U}\times\{\leftarrowtail, \rightarrowtail\}$, $\tau\in\boldsymbol{U}^*$, $|\tau| = l$:

– ұзындығы $\boldsymbol{L}+1$ ($\boldsymbol{L}$– алғашқы $\tau\in\boldsymbol{U}^*$ тізбесінің ұзындығы, оған бір қосылады – күй символы q_i) США-ның барлық мүмкін кескіні құрылады. Мұндай кескіндер жиын ақырлы;

– осы кескіндерден қайталаусыз тізбектер құрылады (әрбір кескін кортежге бір рет кіреді). Мұндай тізбектер жиыны ақырлы.

(q_s, $\vdash\tau\dashv$, 1) кескінінен басталып (q, $\vdash\alpha\dashv$, i)&$q\in\boldsymbol{F}$&$\alpha\in\boldsymbol{T}^*$& $1\le i\le n$ & $n=|\tau|+2$) кескініне аяқталатын тізбектердің ішжиынын ерекшеленеді.

Егер осы тізбектердің ішінде США-ың жұмыс істеу вариантын анықтайтындар табылса, онда $\tau\in\boldsymbol{U}^*$ тізбесі автоматпен танылады, кері жағдайда – танылмайды.

IV.2.1-тапсырмалар. Мына тілді танитын бейдетерминді сызықты-шенеуленген автомат тұрғызыңыз:

1. $\{a^n b^n c^n: n\ge 1\}$;
2. $\{a^n b^n c^n d^n: n\ge 1\}$;
3. $\{(ab)^n(ba)^n(ab)^n: n\ge 1\}$;
4. $\{\tau c\tau: \tau\in\{a,b\}^*\}$;
5. $\{\tau\in\{a,b,c\}^*: |\tau|_a<|\tau|_b<|\tau|_c\}$;
6. $\{\omega\omega^R: \omega \in \{a,b\}$ & $\omega = ab, \omega^R= ba\}$;
7. $\{\omega^n\omega^m: \omega \in \{a,b\}, n\ge 1, m\ge 1\}$.

IV.2.1-сұрақтар: Мына тілді бейдетерминді сызықты-шенеуленген автоматпен танила ма:

1. $\{\tau c\tau : \tau\in\{a,b,c\}^*\}$?
2. $\{\tau\in\{a,b,c,d\}^* : |\tau|_a\ne|\tau|_b, |\tau|_c \ne|\tau|_d\}$?
3. $\{\tau\in\{a,b,c\}^* : |\tau|_a<|\tau|_b< |\tau|_c\}$?
4. $\{\omega\omega^R : \omega \in \{a,b\}$ & $\omega = ab, \omega^R= ba\}$?
5. $\{\omega^n\omega^m: \omega \in \{a,b\}, n\ge 1, m\ge 1\}$?
6. $\{a^n b^n c^n : n\ge 1\}$?
7. $\{a^m b^n a^m b^n : m,n \ge 1\}$?

III.3. Контексті тілдердің қасиеттері

III.3.1. Контексті грамматикалар мен сызықты-шенеуленген автоматтар эквиваленттігі

Бұл бөлімде контексті тілдердің қасиеттері қарастырылады, контексті грамматикалар мен сызықты шенеуленген автоматтардың эквиваленттігі көрсетіледі, контексті тілдердің алгоритмдік проблемалары талқыланады, мысаладар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1-4,6-9,11-13,15-21,24-32].

Контексті-тәуелді грамматикалармен (КТГ) тудырылған тілдер тобы сызықты шенеуленген автоматтармен (США) танылған тілдер тобына дәл келетінін көрсетуге болады. Бұл берілген КТГ сәйкес бейдетерминді США тұрғызу және керісінше США сәйкес КТГ құру керек.

III.3.1-лемма. Мейлі $\boldsymbol{L}(\boldsymbol{G})$ тілі КТГ $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$-мен тудырылсын, онда $\boldsymbol{L}(\boldsymbol{M}) \subseteq \boldsymbol{L}(\boldsymbol{G})$ болатындай бейдетерминді США $\boldsymbol{M}$ бар болады.

Дәлелдеу. Бізге $\boldsymbol{L}(\boldsymbol{M})$ тілін танитын $\boldsymbol{M}$ бейтерминді США тұрғызу керек. $\boldsymbol{M}$ автоматын тұрғызу мейлінше қиын болғандықтан, оның егжей-тегжейіне берілмей, автомат жұмысының сұлбасын қарастырайық.

Айталық автомат таспасы екі жолды болсын. Бірінші жолда соңғылық белгілегішпен бірге кіріс тізбесі болады, яғни бірінші жолдың ішінде $\vdash\tau\dashv$ болады, мұнда $\vdash$ және $\dashv$ таспаның сол жақ және оң жақ шетін белгілегіштер, $\tau = t_1\ t_2\ \ldots\ t_n$ – кіріс тізбесі, $t_i \in \boldsymbol{T}$, $1 \leq i \leq n$. Екінші жол есептеуге пайдаланылады және ол мына $\upsilon = u_1\ u_2\ \ldots\ u_n$, $u_i \in \boldsymbol{N} \cup \boldsymbol{T} \cup \{\$\}$ тізбені қамтиды, мұнда \$ – бос ұяшықты толтырғыш, $1 \leq i \leq n$. Сонымен, таспада мына тізбелер қосағы (τ, υ) болады. Алғашқы кескінде келесі қосақтар тізбесі $(\vdash,\$)$, $(t_1, \$)$,…,$(t_n, \$)$, $(\dashv,\$)$ болады.

США $\boldsymbol{M}$ келесі тудыру процедурасын орындайды:

1. Бастапқы бейтерминалдық символ S екінші жолдың ең сол жағындағы ұяшыққа орналасады;

2. Екінші жолдан $\alpha\rightarrow\beta \in \boldsymbol{P}$ болатындай іштізбе α таңдалынады;

3. α іштізбесі β тізбесіне ауыстырылады;

4. Егер $|\beta| > |\alpha|$, онда екінші жолда α-ның оң жағындағы символдар оң жаққа қарай жылжытылады, мұнда, егер бұл таспаның соңын белгілегіштен асып кететін болса, онда тудыру процедурасы тоқтайды (аралық сентенциалдық пішімі кірістен ұзын болмауы керек).

5. 2-ші қадамға көшу немесе тудыру процедурасының жұмысын аяқтау бейдетерминді таңдалынады.

Тудыру процедурасының шығысында бірінші жол әлі τ тізбесін қамтыса, ал екінші жол $S \Rightarrow_{\boldsymbol{G}}^{*} \upsilon$ болатындай υ тізбесін қамтиды. США бірінші жолдың символдарын екінші жолдың сәйкес символдарымен салыстырады. Осыған орай:

1) егер бірінші және екінші тізбелердің символдары бірдей болмаса, яғни $\tau \neq \upsilon$, онда США $\boldsymbol{M}$ τ кіріс тізбесін танымай тоқтайды деп есептелінеді;

2) егер бірінші және екінші тізбелердің символдары бірдей болса, яғни $\tau=\upsilon$, онда США τ кіріс тізбесін танып тоқтайды;

Егер $\tau\in\boldsymbol{L}(\boldsymbol{G})$, онда екінші жолда τ тізбесін құратын және кірісті танитын США $\boldsymbol{M}$-ның қадамдар тізбегі бар болады. Дәл солай, США $\boldsymbol{M}$ τ тізбесін тану үшін екінші жолда τ тізбесін құратын қадамдар тізбегі бар болуы керек. Сонымен, КТГ $\boldsymbol{G}$-да S бастапқы бейтерминал символдан τ тізбесін шығаратын шығарым болуы керек.

III.3.2-лемма. Егер $\boldsymbol{L}$ тілі США $\boldsymbol{M}$-мен танылса, яғни $\boldsymbol{L} = \boldsymbol{L}(\boldsymbol{M})$, онда $\boldsymbol{L}$ тілін тудыратын КЗГ $\boldsymbol{G}$ бар, яғни $\boldsymbol{L} = \boldsymbol{L}(\boldsymbol{G})$.

Дәлелдеу. КТГ бейтерминалдары США ұяшықтардың тек алғашқы және ағымдағы қамтуын ғана емес, сол жақ немесе оң жақ көрші ұяшық таспаның шетін белгілегіштері болатын-болмайтындығын да көрсетеді. Сонымен қатар, КТГ-ның таспа шеттерін белгілегіштері мен США күйлері үшін бөлек символдары болмайтындықтан, США күйлері бейтерминалдың белгілеуінде бастиек астындағы символмен қосылуы керек. Қашан тізбе терминалды болғанда, ал ε-*туындауға* тиім салынғанда, бұл символдар бос тізбе ε ауыстырылуы керек.

Мейлі, $\boldsymbol{L}$ тілін танитын США $\boldsymbol{M} = \langle \boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{U}, q_s, \boldsymbol{F}, \vdash, \dashv, |\!\!\leftarrow, \rightarrow\!\!|, \delta \rangle$ берілсін және $\varepsilon \notin \boldsymbol{L}$.

Енді алдымен $\boldsymbol{T}^*$ жиынындағы қандай-да бір тізбенің екі көшірмесін бейдетерминді тудырып, сонан кейін осы көшірменің бірінде США $\boldsymbol{M}$ моделденеді. Егер США тізбені танитын болса, онда $\boldsymbol{G}$ грамматикасы екінші көшірмені терминалдыға айналдырады. Егер США тізбені танымайтын болса, онда $\boldsymbol{G}$ грамматикасындағы шығарым ешқашан терминалды тізбеге алып келмейді.

Айталық, $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, A_1 \rangle$ – КТГ, мұнда

$\boldsymbol{N} = \{A_1, A_2\} \cup \{[q, \vdash, x, t, \dashv], [\vdash, q, x, t, \dashv], [\vdash, x, t, q, \dashv], [q, x, t],$
$[q, x, t, \dashv], [x, t, q, \dashv], [\vdash, x, t], [x, t], [x, t, \dashv] \mid t \in \boldsymbol{T}, x \in \boldsymbol{U} \setminus \{\vdash, \dashv\}, q \in \boldsymbol{Q}\}$;

$\boldsymbol{T} = \boldsymbol{T}$; ал $\boldsymbol{P}$ жиыны келесі ережелерді қамтиды:

(1) $(q_0, \vdash t \dashv, 1)$ түрінлегі алғашқы кескінді моделдеу:

$$A_1 \rightarrow [q_0, \vdash, t, t, \dashv];$$

(2) $q \in \boldsymbol{Q} \setminus \boldsymbol{F}$ болғанда бір символды тізбедегі қозғалысты моделдеу:

$$[q, \vdash, x, t, \dashv] \rightarrow [\vdash, p, x, t, \dashv], \text{ егер } (p, \vdash, \rightarrow\!\!|) \in \delta(q, \vdash);$$
$$[\vdash, q, x, t, \dashv] \rightarrow [p, \vdash, y, t, \dashv], \text{ егер } (p, y, |\!\!\leftarrow) \in \delta(q, x);$$
$$[\vdash, q, x, t, \dashv] \rightarrow [\vdash, y, t, p, \dashv], \text{ егер } (p, y, \rightarrow\!\!|) \in \delta(q, x);$$
$$[\vdash, x, t, q, \dashv] \rightarrow [\vdash, p, x, t, \dashv], \text{ егер } (p, \dashv, |\!\!\leftarrow) \in \delta(q, \dashv);$$

(3) $q \in \boldsymbol{F}$ болғанда бір символды кіріс тізбесін қайта тұрғызу:

$$[q, \vdash, x, t, \dashv] \rightarrow t; \; [\vdash, q, x, t, \dashv] \rightarrow t; \; [\vdash, x, t, q, \dashv] \rightarrow t;$$

(4) $(q_0, \vdash \omega \dashv, 1)$ түріндегі, мұнда $|\omega| > 1$, алғашқы кескінді моделдеу:

$$A_1 \rightarrow [q_0, \vdash, t, t] A_2; \; A_2 \rightarrow [t, t] A_2; \; A_2 \rightarrow [t, t, \dashv];$$

(5) $q \in \boldsymbol{Q} \setminus \boldsymbol{F}$ болғанда тізбенің сол жақ шетіндегі қозғалысты моделдеу:

$$[q, \vdash, x, t] \rightarrow [\vdash, p, x, t], \text{ егер } (p, \vdash, \rightarrow\!\!|) \in \delta(q, \vdash);$$
$$[\vdash, q, x, t] \rightarrow [p, \vdash, y, t], \text{ егер } (p, y, |\!\!\leftarrow) \in \delta(q, x);$$
$$[\vdash, q, x, t]\,[z, u] \rightarrow [\vdash, y, t]\,[p, z, u], \text{ егер } (p, y, \rightarrow\!\!|) \in \delta(q, x);$$

(6) $q \in \boldsymbol{Q} \setminus \boldsymbol{F}$ болғанда тізбенің ортасындағы қозғалысты моделдеу:

$$[q, x, t]\,[z, u] \rightarrow [y, t]\,[p, z, u], \text{ егер } (p, y, \rightarrow\!\!|) \in \delta(q, x);$$
$$[z, u]\,[q, x, t] \rightarrow [p, z, u]\,[y, t], \text{ егер } (p, y, |\!\!\leftarrow) \in \delta(q, x);$$

$$[q, x, t]\ [z, u, \dashv] \to [y, t]\ [p, z, u, \dashv], \text{ егер } (p, y, \to\!\mid) \in \delta(q, x);$$

(7) $q \in \boldsymbol{Q} \setminus \boldsymbol{F}$ болғанда тізбенің оң жақ шетіндегі қозғалысты моделдеу:

$$[q, x, t, \dashv] \to [y, t, p, \dashv], \text{ егер } (p, y, \to\!\mid) \in \delta(q, x);$$

$$[x, t, q, \dashv] \to [p, x, t, \dashv], \text{ егер } (p, \dashv, \mid\!\leftarrow) \in \delta(q, \dashv);$$

$$[z, u]\ [q, x, t, \dashv] \to [p, z, u]\ [y, t, \dashv], \text{ егер } (p, y, \mid\!\leftarrow) \in \delta(q, x);$$

(8) тізбенің сол жақ шетінде $q \in \boldsymbol{F}$ күйіне өту кезінде кіріс символын қайта тұрғызу:

$$[q, \vdash, x, t] \to t;\ [\vdash, q, x, t] \to t;$$

(9) тізбенің ортасында $q \in \boldsymbol{F}$ күйіне өту кезінде кіріс символын қайта тұрғызу:

$$[q, x, t] \to t;$$

(10) тізбенің оң жақ шетінде $q \in \boldsymbol{F}$ күйіне өту кезінде кіріс символын қайта тұрғызу:

$$[q, x, t, \dashv] \to t;\ [x, t, q, \dashv] \to t;$$

(11) тізбені солдан оңға қайта тұрғызу:

$$t[x, u] \to tu;\ t[x, u, \dashv] \to tu;$$

(12) тізбені оңнан солға қайта тұрғызу:

$$[x, t]\ u \to tu;\ [\vdash, x, t]\ u \to tu;$$

Мұнда $p \in \boldsymbol{Q}$; $u \in \boldsymbol{T}$; $y \in \boldsymbol{U} \setminus \{\vdash, \dashv\}$; $z \in \boldsymbol{U} \setminus \{\vdash, \dashv\}$.

Енді егер бос емес тізбе τ СЩА ***M***-мен танылса, онда τ КТГ ***G***-мен тудырылатындығын СЩА ***M*** қозғалысының саны бойынша индукция арқылы дәлелдуге болады. Керісінше, егер тізбе τ КТГ ***G***-мен тудырылса, онда τ СЩА ***M***-мен танылатындығын шығарым ұзындығы бойынша индукция арқылы дәлелденеді. Сондықтан, егер $\tau \in \boldsymbol{L}(\boldsymbol{G})$, онда $\tau \in \boldsymbol{L}(\boldsymbol{M})$ және керісінше, егер $\tau \in \boldsymbol{L}(\boldsymbol{M})$, онда $\tau \in \boldsymbol{L}(\boldsymbol{G})$.

III.3.1- лемма және III.3.2- леммадан келесі теорема туады.

III.3.1-теорема. Бос тізбені қамтымайтын ***L*** тілі қандай-да бір грамматикамен туындайды тек қана сонда және тек қана сонда, қашан ***L*** тілі БСЩА танылса.

Язык $\boldsymbol{L}$, не содержащий пустого слова, порождается некоторой неукорачивающей грамматикой тогда және только тогда, когда существует НЛОА, допускающий язык $\boldsymbol{L}$.

III.3.1-тапсырмалар: Контекстісіз $\boldsymbol{G} = <N, \boldsymbol{T}, \boldsymbol{P}, S>$ грамматикасына эквивалентті сызықты шенеуленген автомат тұрғызыңыз, мұнда:

1. $\boldsymbol{N}=\{A, B, S\}$, $\boldsymbol{T}=\{a, b, c\}$, $\boldsymbol{P}=\{S\to aAB, A\to abc, bB\to bbc, cB\to Bc\}$.
2. $\boldsymbol{N}=\{A, B, S\}$, $\boldsymbol{T}=\{a, b, c\}$, $\boldsymbol{P}=\{S\to AB, A\to a, A\to ac, B\to b, B\to cb\}$.
3. $\boldsymbol{N}=\{S\}$, $\boldsymbol{T}=\{a, b\}$, $\boldsymbol{P}=\{S\to SbaS, S\to ab, aS\to b\}$.
4. $\boldsymbol{N}=\{A, B, C, D, S\}$, $\mathrm{T}=\{a, b, c\}$, $\boldsymbol{P}=\{S\to CD, C\to aCA, C\to bCB, AD\to aD, BD\to bD, Aa\to aA, Ab\to bA, Ba\to aB, Bb\to bB, C\to c, D\to c\}$.
5. $\boldsymbol{N}=\{A, B, C, S\}$, $\boldsymbol{T}=\{0, 1, 2\}$, $\boldsymbol{P}=\{S\to A1, A\to 1A1|0A0|B, B\to 021|$
$120C, C1\to 0C, C0\to 1, R1\to 11\}$.

6. Мына тілдерді тудыратын қысқармайтын грамматика және танитын сызықты шенеуленген автомат тұрғызыңыз:

1) $\{\omega\in\{a, b, c\}^*: |\omega|_a < |\omega|_b < |\omega|_c\}$.
2) $\{a^n b^n a^n: n\geq 1\}$.
3) $\{(ab)^n(ba)^n(ab)^n: n\geq 1$.

III.3.1-сұрақтар: Келесі $\boldsymbol{G} = <N, \boldsymbol{T}, \boldsymbol{P}, S>$ грамматикасымен, мұнда
1. $\boldsymbol{N}=\{B, S\}$, $\boldsymbol{T}=\{a, b\}$, $\boldsymbol{P}=\{S\to a|Ba, B\to Bb|b\}$;
2. $\boldsymbol{N}=\{A, B, S\}$, $\boldsymbol{T}=\{a, b\}$, $\boldsymbol{P}=\{S\to Ab, A\to Aa|ba\}$;
3. $\boldsymbol{N}=\{A, B, S\}$, $\boldsymbol{T}=\{a, b, c\}$, $\boldsymbol{P}=\{S\to aAb|ab, aA\to aaAb, A\to ab\}$;
4. $\boldsymbol{N}=\{A, B, S\}$, $\boldsymbol{T}=\{a, b\}$, $\boldsymbol{P}=\{S\to AB, AB\to BA, A\to a, B\to b\}$;
5. $\boldsymbol{N}=\{B, C, S\}$, $\boldsymbol{T}=\{a, b, c\}$, $\boldsymbol{P}=\{S\to abC|aB, B\to bc, bC\to bc\}$;
6. $\boldsymbol{N}=\{A,S\}$, $\boldsymbol{T}=\{0, 1\}$, $\boldsymbol{P}=\{S\to 0A1, 0A\to 00A1, A\to\varepsilon\}$

туындайтын тілдерді танитын сызықты шенеуленген автоматтар бар бола ма?

7. Мына тілдерді тудыратын қысқармайтын грамматика және танитын сызықты шенеуленген автомат тұрғызуға болады ма:

1) $\{\omega\in\{a, b, c\}^+: |\omega|_a = |\omega|_b = |\omega|_c\}$?
2) $\{c^n dc^n dc^n: n\geq 0\}$?
3) $\{\omega c\omega: \omega \in \{a, b\}*\}$?

III.3.2. Контексті тілдердің алгоритмдік проблемалары

Бұл параграфта контексті тілдердің алгоритмдік проблемалары талқыланады, мысаладар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1-4,8,9,11-13,15-21,24-32].

Олардың ішінде әліде болса ашық проблемалар бар, яғни олардың алгоритмді шешілімдігі туралы осы күнге шейін жауап жоқ. Бізді көбінесе алгоритмді шешілімді проблемалар қызықтырады. Практикалық есептерді шешкенде бұл проблемаларды білу өте маңызды. Төменде сондай проблемалардың кейбіреулері туралы айтылады:

Мүше болу проблемасы қысқармайтын грамматикада алгоритмді шешілімді, яғни келесі теорема ақиқат.

III.3.2.1-теорема. Қайсы бір қысқармайтын грамматика $\boldsymbol{G}$ және тізбе τ бойынша $\tau \in \boldsymbol{L}(\boldsymbol{G})$ рас па жоқ па білуге болады.

Тұйықтылық проблемасы контексті тілдер мүше болу, бірігу, қиылысу және толықтыру бойынша тұйықтылығы алгоритмді шешілімді, яғни келесі теорема ақиқат.

III.3.2.2-теорема. Қысқармайтын грамматикалармен туындайтын тілдер тобы, яғни контексті тілдер мүше болу, бірігу, қиылысу және толықтыру бойынша тұйық.

Толықтыру бойынша тұйықтылықты 1987 жылы (бір біріне тәуелсіз) Нил Иммерман (Neil Immerman) и Роберт Селепчени (Robert Szelepcheni) дәлелдеді. Бірігу бойынша толықтылық айқын, ал қиылысу бірігу мен толықтыру арқылы өрнектеледі.

Бостық проблемасы қысқартылмайтын грамматикада алгоритмді шешілімді, яғни келесі теорема ақиқат.

III.3.2.3-теорема. Қайсы бір қысқармайтын грамматика $\boldsymbol{G}$ бойынша $\boldsymbol{L}(\boldsymbol{G}) = \varnothing$ рас па жоқ па білуге болады

Дәлелдеу. ***G*** грамматикасынан барлық пайдасыз символдарды, алғашқы символдан басқа, жоямыз. Егер алынған грамматика ең болмаса бір ережені қамтыса, онда $\boldsymbol{L}(\boldsymbol{G}) \neq \varnothing$.

III.3.2.1-мысалдар:

1. Контексті тіл $\boldsymbol{L}(\boldsymbol{G}) = \{a^n b^n c^n \colon n \geq 1\}$ мына $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ грамматикасымен туындайды, мұнда $\boldsymbol{N}=\{B, S\}$, $\boldsymbol{T}=\{a, b, c\}$, $\boldsymbol{P}=\{S \to aSB, S \to abc, bB \to bbc, cB \to Bc\}$ себебі, мынадай шығарымдар: $S \to aSB \to aabcB \to aabBc \to aabbcc$ тұрғызуға болады.

2. Контексті тіл $\boldsymbol{L}(\boldsymbol{G}) = \{a^n b^n c^n \colon n \geq 1\}$ мына $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ грамматикасымен туындайды, мұнда $\boldsymbol{N}=\{B, C, S\}$, $\boldsymbol{T}=\{a, b, c\}$, $\boldsymbol{P}=\{S \to aSBC|abC, CB \to BC, bB \to bb, bC \to bc, cC \to cc\}$, себебі, мынадай шығарымдар: $S \to aSBC \to aabCBC \to aabBCC \to aabbCC \to aabbcC \to aabbcc$ тұрғызуға болады.

3. Контексті тіл $\boldsymbol{L}(\boldsymbol{G}) = \{(ab)^n \colon n \geq 1\}$ мына $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ грамматикасымен туындайды, мұнда $\boldsymbol{N}=\{A, B, C, D, S\}$, $\boldsymbol{T}=\{a, b\}$, $\boldsymbol{P}=\{S \to CD, C \to aCA, C \to bCB, AD \to aD, BD \to bD, Aa \to aA, Ab \to bA, Ba \to aB, Bb \to bB, C \to \varepsilon, D \to \varepsilon\}$, себебі, мынадай шығарымдар: $abCBaD \to abCaBD \to abCabD \to ab\varepsilon abD \to ab\varepsilon ab\varepsilon \to abab$ тұрғызуға болады.

4. Контексті тіл $\boldsymbol{L}(\boldsymbol{G}) = \{ab^{n+1} \colon n \geq 1\}$ мына $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ грамматикасымен туындайды, мұнда $\boldsymbol{N}=\{S\}$, $\boldsymbol{T}=\{a, b\}$, $\boldsymbol{P}=\{S \to SbaS, S \to ab, aS \to b\}$, себебі, мынадай шығарымдар: $S \to SbaS \to abbaS \to abbb$ немесе $S \to SbaS \to abbaab$ тұрғызуға болады.

5. Контексті тіл $\boldsymbol{L}(\boldsymbol{G}) = \{(abc)^n \colon n \geq 1\}$ мына $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ грамматикасымен туындайды, мұнда $\boldsymbol{N}=\{B,C,S\}$, $\boldsymbol{T}=\{a, b,c\}$, $\boldsymbol{P}=\{S \to abC|aB, B \to bc, bC \to bcS\}$, себебі, мынадай шығарымдар: $S \to abC \to abcS \to abcaB \to abcabc$ немесе$S \to aB \to abc$ тұрғызуға болады.

6. Контексті тіл $\boldsymbol{L}(\boldsymbol{G}) = \{(01)^n \colon n \geq 1\}$ мына $\boldsymbol{G} = \langle \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S \rangle$ грамматикасымен туындайды, мұнда $\boldsymbol{N}=\{A,B,C\}$, $\boldsymbol{T}=\{a,b,c\}$, $S=A$, $\boldsymbol{P}=\{A \to 01BA, A \to 01C, 1B \to 101, C \to \varepsilon\}$ себебі, мынадай шығарымдар: $A \to 01BA \to 01B01C \to 010101\varepsilon \to 010101$ тұрғызуға болады.

III.3.2.1-тапсырмалар:

1. Берілген $\boldsymbol{G} = \langle N, T, P, S \rangle$, мұнда $\boldsymbol{N} = \{A, B, C\}$, $\boldsymbol{T} = \{a, b, c\}$, $\boldsymbol{P} = \{A \rightarrow aAB,\ A \rightarrow Abc,\ bB \rightarrow bbc,\ cB \rightarrow Bc\}$ контексті грамматикасымен туатын тізбенің мына $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{a^n b^n c^n : n \geq 1\}$ тілге жататынын дәлелдеңіз.

2. Мына тілді $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{\omega \in \{a, b, c\}^+ : |\omega|_a = |\omega|_b = |\omega|_c\}$ тудыратын контексті $\boldsymbol{G}$ грамматикасын табыңыз.

Мына тілді $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{a^n b^n a^n : n \geq 1\}$ тудыратын контексті $\boldsymbol{G}$ грамматикасын табыңыз.

3. Мына тілді $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{c^n dc^n dc^n : n \geq 0\}$ тудыратын контексті $\boldsymbol{G}$ грамматикасын табыңыз.

4. Мына тілді $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{(ab)^n (ba)^n (ab)^{\mathrm{n}} : n \geq 1\}$ тудыратын контексті $\boldsymbol{G}$ грамматикасын табыңыз.

5. Мына тілді $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{a^n b^n c^n d^n : n \geq 1\}$ тудыратын контексті $\boldsymbol{G}$ грамматикасын табыңыз.

6. Мына тілді $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{a^n (ba)^n ba^n : n \geq 1\}$ тудыратын контексті $\boldsymbol{G}$ грамматикасын табыңыз.

7. Мына тілді $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{\omega c \omega : \omega \in \{a, b\}^*\}$ тудыратын контексті $\boldsymbol{G}$ грамматикасын табыңыз.

8. Мына тілді $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{a^{2n} : n \geq 0\}$ тудыратын контексті $\boldsymbol{G}$ грамматикасын табыңыз.

III.3.2.1-сұрақтар:

1. Мына $S \rightarrow aaCM$, $S \rightarrow aaaKT$, $M \rightarrow aMb$, $M \rightarrow bMa$, $M \rightarrow \varepsilon$, $C \rightarrow aCa$, $C \rightarrow bCb$, $K \rightarrow bB$, $K \rightarrow aB$, $B \rightarrow bKa$, $B \rightarrow ab$ ережелері бар контексті грамматикамен туындайтын тіл бос бола ма?

2. Мына тіл $\{a^m b^n a^m b^n : m \geq, n \geq 1\}$ қандай топтағы грамматикамен туындайды?

3. Контексті грамматиканың мына $S \rightarrow aF$, $S \rightarrow aaCM$, $S \rightarrow aaaKF$, $F \rightarrow J$, $F \rightarrow a$, $J \rightarrow F$, $J \rightarrow b$, $M \rightarrow aMb$, $M \rightarrow bMa$, $M \rightarrow \varepsilon$, $C \rightarrow aCa$, $C \rightarrow bCb$, $K \rightarrow bT$, $K \rightarrow aT$, $T \rightarrow bKa$, $T \rightarrow ab$ ережелерімен туындаған тіл шексіз болады ма?

4. Мына тілді $\{0^i 1^j : i \neq j, i \geq 0, j \geq 0\}$ тудыратын контексті грамматиканың ережелері қандай болады?

V. ШЕНЕУЛЕНБЕГЕН ТІЛДЕР

V.1. Шенеуленбеген тілдерді тудырушы механизмдер

V.1.1. Шенеуленбеген грамматика

Бұл бөлімде шенеуленбеген тілдер қарастырылады, олар 0-типтегі грамматикалармен туындайды және Тьюринг машиналарымен танылады. 0-типтегі грамматикаға жататын шенеуленбеген грамматикалар және Тьюринг машиналары сипатталады. Шенеуленбеген грамматикалар мен Тьюринг машиналарының эквиваленттігі көрсетіледі. Шенеуленбеген тілдердің алгоритмдік проблемалары талқыланады. Мысаладар ұсынылады, тапсырмалар берілелі және сұрақтар қойылады [1-4,8,9,11-13,15-21,24-32].

I.3.1-бөлімде грамматиканың формалды анықтамасы келесі төрттік $\boldsymbol{G} = <\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{P}, S>$ түрінде берілген, мұнда:

$\boldsymbol{T}$ – терминал символдардың бос емес ақырлы жиыны;

$\boldsymbol{N}$ – бейтерминал символдардың бос емес ақырлы жиыны, $\boldsymbol{T}\cap\boldsymbol{N}=\varnothing$, $\varnothing$ – бос жиын;

$\boldsymbol{P}$ – $\alpha\to\beta$ түріндегі шығарым ережелерінің бос емес ақырлы жиыны, мұнда $\alpha\in(\boldsymbol{T}\cup\boldsymbol{N})^*\times\boldsymbol{N}\times(\boldsymbol{T}\cup\boldsymbol{N})^*$, $\beta\in(\boldsymbol{T}\cup\boldsymbol{N})^*$, яғни

$$\boldsymbol{P}\subseteq\{(\alpha,\beta) : \alpha\in(\boldsymbol{T}\cup\boldsymbol{N})^*\times\boldsymbol{N}\times(\boldsymbol{T}\cup\boldsymbol{N})^*\&\beta\in(\boldsymbol{T}\cup\boldsymbol{N})^*\};$$

S – бастапқы бейтермина символы, $S\in\boldsymbol{N}$.

Мынаны еске алсақ:

1) ω терминал тізбесі $\boldsymbol{G}$ грамматикасымен туындайды, егер осы тізбенің S бастапқы бейтерминалдан туындаса, яғни $S\Rightarrow^*\omega$.

2) $\boldsymbol{G}$ грамматикасымен туындайтын тіл деп бастапқы S бейтерминал символынан шығарымды болатын терминалдар тізбесінінің жиыны, яғни

$$\boldsymbol{L}(\boldsymbol{G})\rightleftharpoons\{\omega : \omega\in\boldsymbol{T}^*, S\Rightarrow^*\omega\}.$$

V.1.1-мысалдар:

1. Мейлі терминал символдары $\boldsymbol{T} = \{a, b\}$, бейтерминал символдары $\boldsymbol{N} = \{S, A, B\}$ және ережелері $\boldsymbol{P} = \{S\to Sb, S\to SA, S\to SB, S\to\varepsilon, Aa\to aaA, Ab\to ab, Baaa\to aaB, Bab\to b\}$ бар $\boldsymbol{G}$ грамматикасында

aab тізбесі туындайтын болсын. Сонда оның мынадай шығарымы $S \Rightarrow^* BBTTTb \Rightarrow^* BBaaaaaaab \Rightarrow^* aab$ бар болады.

2. Шенеуленбеген грамматика мынадай төрттіктен құралады $\boldsymbol{G} = <\{0, 1\}, \{A, S\}, \boldsymbol{P}, S>$, мұнда $\boldsymbol{P} = \{S \to 0A, 0A \to 00A1, A \to \varepsilon\}$. Бұл $\boldsymbol{G}_1$ грамматикасы құрамында нөлдер мен бірліктердің саны тең болатын тізбелерді тудырады, яғни $\boldsymbol{L}(\boldsymbol{G}) = \{0^n1^n : n>1\}$.

3. 0-типтегі тіл 0 $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{a^2b^{n^2-1} : n\geq 1\}$ мына ережелері бар шенеуленбеген грамматикамен туындайды: $S \to aaCFD$, $F \to AFB|AB$, $AB \to bBA$, $Ab \to bA$, $AD \to D$, $Cb \to bCғ$ $CB \to C$, $bCD \to \varepsilon$.

V.1.1-тапсырмалар: Берілген шенеуленбеген $\boldsymbol{G} = < \boldsymbol{T}, N, \boldsymbol{P}, S>$ грамматикасы үшін осы грамматикамен туындайтын $\boldsymbol{L}(\boldsymbol{G})$ тілін тұрғызу керек, мұнда:

1. $\boldsymbol{T} = \{a, b, d\}$, $\boldsymbol{N} = \{A, B, D\}$, $S = A$, $\boldsymbol{P} = \{A \to aB, B \to aB, B \to b, B \to bD, D \to d, D \to dD, A \to aD, A \to a\}$.

2. $\boldsymbol{T} = \{a, b, e\}$, $\boldsymbol{N} = \{A, B, C, D, S\}$, $\boldsymbol{P} = \{S \to CD, C \to aCA, C \to bCB, AD \to aD, BD \to bD, Aa \to aA, Ab \to bA, Ba \to aB, Bb \to bB, C \to e, D \to e\}$.

3. $\boldsymbol{T} = \{a, b, c\}$, $\boldsymbol{N} = \{A, B, C, D, S\}$, $\boldsymbol{P} = \{S \to CD, C \to aCA, C \to bCB, AD \to aD, BD \to bD, Aa \to aA, Ab \to bA, Ba \to aB, Bb \to bB, C \to c, D \to c\}$.

4. $\boldsymbol{T} = \{a,b,c\}$, $\boldsymbol{N} = \{B,C,S\}$, $\boldsymbol{P} = \{S \to abC|aB, B \to bc, bC \to bc\}$.

5. $\boldsymbol{T} = \{a,b,c\}$, $\boldsymbol{N} = \{A, B, C, D, S\}$, $\boldsymbol{P} = \{S \to CD, C \to aCA, C \to bCB, AD \to aD, BD \to bD, Aa \to aA, Ab \to bA, Ba \to aB, Bb \to bB, C \to c, D \to c\}$.

6. $\boldsymbol{T} = \{a,b,c\}$, $\boldsymbol{N} = \{B,C,S\}$, $\boldsymbol{P} = \{S \to aSBC|abC, CB \to BC, bB \to bb, bC \to bc, cC \to cc\}$.

7. $\boldsymbol{T} = \{a,b\}$, $\boldsymbol{N} = \{A,B,S\}$, $\boldsymbol{P} = \{S \to AB, AB \to BA, A \to a, B \to b\}$.

V.1.1-сұрақтар: Мына тілді тудыратын шенеуленбеген грамматика бола ма:

1. $\{a^mb^na^mb^n : m\geq 1 \;\&\; n\geq 1\}$?

2. $\{a^kb^mc^n : k\geq 1 \;\&\; m\geq 1 \;\&\; n\geq 1\}$?

3. $\{a^kb^ka^nb^n : k\geq 1 \;\&\; n\geq 1\}$?

4. $\{a^{k+n}b^{k-n}: k\geq 1 \;\&\; n\geq 1\}$?

5. $\{(a^kb^m)c^n : k\geq 1 \;\&\; m\geq 1 \;\&\; n\geq 1\}$?

6. $\{a^{k-m}b^mc^{n-m} : k\geq 1 \;\&\; m\geq 1 \;\&\; n\geq 1\}$?

V.2. Шенеуленбеген тілдерді танушы механизмдер

V.2.1. Тьюринг машиналары

Бұл бөлімде шенеуленбеген тілдердің танушы механизмдері қарастырылады, олар Тьюринг машиналары (ТМ) болады. Олардың құрамы, құрылымы және функциясы сипатталады. ТМ-ның кескіні, тактысы және программасының, сондай-ақ, онымен танылатын тілдің формалды анықтамасы беріледі. Мысаладар ұсынылады, тапсырмалар және сұрақтар қойылады [1-4,8,9,11-13,15-21,24-32].

ТМ деп алгоритм ұғымын айқындау үшін 1936 жылы ағылшын математигі Алан Тьюринг ұсынған формализмді айтады.

ТМ құрамына сол жағынан шектелген ақырсыз *таспа*; оқушы және жазушы *бастиек*; *басқарушы құрылғы*; ішкі және сыртқы *жад* кіреді.

Таспа әрқайсына ақырлы кіріс-шығыс әліпби символын, өзінің сол жақ шетін белгілейтін символды немесе бос ұяшықтарды толтыруыш символды жазуға болатын ұяшықтарға бөлінген.

Бастиек әрбір уақыт мезгілінде таспа ұяшықтарының бірін ғана шолу жасайды және осы ұяшықтағы символ мен басқарушы құрылғының күіне байланысты оның ішіндегісін өзгерте (жаза немесе өшіре) алады және бір ұяшыққа солға немесе оңға жылжиды, иә қозғалмай орнында қалады.

Басқарушы құрылғы қандай да бір күйде, соның ішінде ерекшелеуге болатын бастапқы (стартты) күйде және соңғы (финалды) күйде бола алады.

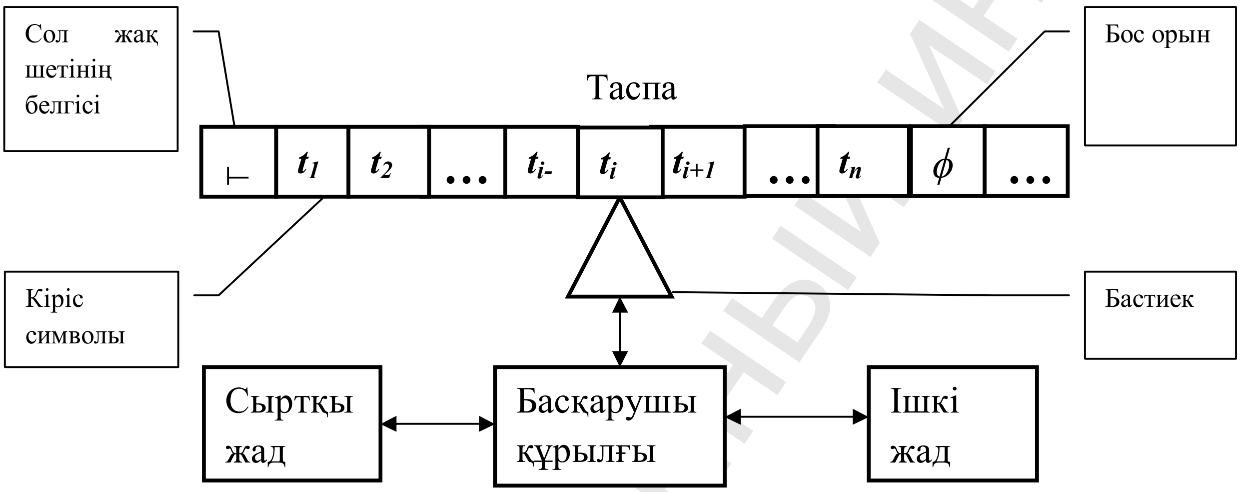
ТМ бастапқы күйде өз жұмысын бастайды, ал соңғы күйде з жұмысын тоқтатады.

Жад екі бөліктен тұрады: *ішкі жад* басқарушы құрылғының барлық мүмкін болатын күйлерінің ақырлы жиынын, ал *сыртқы жад* барлық кіріс және шығыс әліпбиінің символдарын қамтиды.

ТМ-ның *деректері* кіріс және шығыс әліпбиінің символдарынан құрылған тізбелер болады.

ТМ жұмысының алдында басқарушы құрылғы алғашқы күйде болады, таспаның ең сол жағындағы ұяшыққа арнаулы символ ⊢ жазылады, онан кейінгі алғашқы n ұяшық кіріс және шығыс әліпби символдарынан құралған $t_1t_2\ldots t_n$ тізбесін қамтиды, ал қалған ұяшықтарға арнаулы толтыруыш символ ϕ толтырылады. Мұндағы арнаулар символдар ⊢ және ϕ кіріс және шығыс әліпбиінің элементтері емес.

ТМ-ның құрылымы V.2.1-суретте берілген.



V.2.1-сурет. Тьюринг машинасының құрылымы

Енді ТМ-ның формалды (математикалық) анықтамасын берейік.

V.2.1.1-анықтама. Тьюринг машинасы (ТМ) деп мына ондықты айтады:

$$\boldsymbol{M} = < \boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{U}, q_s, \boldsymbol{F}, \vdash, \leftarrowtail, \|, \rightarrowtail, \delta>, \text{ мұнда}$$

$\boldsymbol{Q}$ – басқарушы құрылғы күйлерінің ақырлы жиыны;

$\boldsymbol{U}$ – таспалық символдардың ақырлы жиыны, $\boldsymbol{Q} \cap \boldsymbol{U} = \varnothing$;

$\boldsymbol{T}$ – кіріс символдарының ақырлы жиыны, $\boldsymbol{T} \subseteq \boldsymbol{U}$;

q_s – басқарушы құрылғының алғашқы күйі, $q_s \in \boldsymbol{Q}$;

$\boldsymbol{F}$ – басқарушы құрылғының соңғы күйілерінің жиыны, $\boldsymbol{F} \subseteq \boldsymbol{Q}$;

⊢ – таспаның басын белгілегіш, $\vdash \in \boldsymbol{U}$, $\vdash \notin \boldsymbol{T}$;

ϕ – бос ұяшықты толтырғыш арнаулы символ, $\phi \in \boldsymbol{U}$, $\phi \notin \boldsymbol{T}$;

⇤ – бастиектің солға жылжитындығын көрсететін арнаулы символ;

⇥ – бастиектің оңға жылжитындығын көрсететін арнаулы символ;

$\Vert$ – бастиектің жылжымайтындығын көрсететін арнаулы символ;

Детерминді ТМ-нда ауысулар функциясы δ кейбір аргументтер үшін анықталмауы мүмкін және $\boldsymbol{Q}\times\boldsymbol{U}$ жиынын $\boldsymbol{Q}\times(\boldsymbol{U}\backslash\{\phi\})\times\{⇤,\Vert,⇥\}$ жиынына бейнелейді, яғни

$$\delta\colon \boldsymbol{Q}\times\boldsymbol{U}\rightarrow(\boldsymbol{U}\setminus\{\phi\})\times\{⇤, \Vert, ⇥\}$$

Бейдетерминді ТМ-нда ауысулар функциясы $\boldsymbol{Q}\times\boldsymbol{U}$ жиынын $\boldsymbol{Q}\times(\boldsymbol{U}\backslash\{\phi\})\times\{⇤, \Vert, ⇥\}$ жиынының барлық ішжиындарының жиынына бейнелейді, яғни

$$\delta\colon \boldsymbol{Q}\times\boldsymbol{U}\rightarrow\vartheta(\boldsymbol{Q}\times(\boldsymbol{U}\backslash\{\phi\})\times\{⇤, \Vert, ⇥\}),\ \text{мұнда}$$

$\vartheta(\boldsymbol{Q}\times(\boldsymbol{U}\backslash\{\phi\})\times\{⇤, \Vert, ⇥\})$ – $\boldsymbol{Q}\times(\boldsymbol{U}\backslash\{\phi\})\times\{⇤, \Vert, ⇥\}$ жиынының барлық ішжиындарының жиыны.

ТМ жұмыс істегенде қарапайым қадам жасайды: символды оқу және жазу, бір ұяшыққа солға немесе оңға бастиекті жылжыту, басқарушы құрылғы жаңа күй немесе ескі күй бола алатын келесі күйге көшеді. ТМ өзінің бір қадамында келесі әрекеттерді жасайды:

- басқарушы құрылғы ағымдағы күйді қабылдайды;
- бастиек ағымдағы ұяшықтағы символды оқиды;
- ағымдағы ұяшықтағы символ мен басқарушы құрылғының ағымдағы күйіне байланысты бастиек осы ұяшыққа (мүмкін бұрынғымен бірдей немесе «бос») символ жазады;
- бастиек бір ұяшыққа солға немесе оңға жылжиды, иә болмаса орнынан қозғалмайды.
- басқарушы құрылғы жаңа күйге көшеді немесе өзінің ескі күйінде қалады;
- егер басқарушы құрылғы соңғы күйге түссе, онда ТМ өзінің жұмысын аяқтап, тоқтайды.

Айталық, ТМ $\boldsymbol{M}$ және оның $q\in\boldsymbol{Q}$ – ағымдағы күйі, $q_s\in\boldsymbol{Q}$ – алғашқы күйі, $q_f\in\boldsymbol{Q}$ – соңғы күйі (мұнда s және f сандық айнымалы емес, бастапқы және соңғы мнемоникалық таңба ретінде қарастырылады), $\tau\in(\boldsymbol{U}\setminus\{\phi\})^*$ – таспаның бос емес бөлігі, ал $i\in\boldsymbol{N}$ – бастиектің τ тізбесінің сол жақ шетінен ара қашықтығын көрсететін натурал сан берілсін. Сонда келесі үштікті (q, τ, i) осы Тьюринг машиналарының *кескіні* деп атайды**.**

Ары қарай, мейлі ТМ $\boldsymbol{M}$ үшін $t_1\, t_2 \ldots t_n \in \boldsymbol{T}^*$ – кіріс тізбесі, $(q, t_1\, t_2 \ldots t_n, i)$, $1 \le i \le n+1$ – кескін болсын. Сонда егер:

1) $\delta(q, t_i) = (q', z, \mid\!\leftarrow)$, $z \in \boldsymbol{T}$, $2 \le i \le n$ болса, онда $\boldsymbol{M}$ машиналарының қадамы мына қатынас $(q, t_1\, t_2 \ldots t_n, i) \models_M (q', t_1\, t_2 \ldots t_{i-1}\, z\, t_{i+1} \ldots t_n, i-1)$ ретінде жазуға болады, мұнда бастиек i ұяшығына z символын жазып, бір ұяшыққа солға, бірақ сол жақ шетіндегі $\vdash$ символынан шықпайтындай, жылжиды деп есептелінеді;

2) $\delta(q, t_i) = (q', z, \rightarrow\!\mid)$, $z \in \boldsymbol{T}$, $1 \le i \le n$ болса, онда $\boldsymbol{M}$ машиналарының қадамы мына қатынас $(q, t_1\, t_2 \ldots t_n, i) \models_A (q', t_1\, t_2 \ldots t_{i-1}\, z\, t_{i+1} \ldots t_n, i+1)$ ретінде жазуға болады, мұнда бастиек i ұяшығына z символын жазып, бір ұяшыққа оңға жылжиды деп есептелінеді;

3) $(q, t_i) = (q', z, \|)$, $z \in \boldsymbol{T}$, $1 \le i \le n$ болса, онда $\boldsymbol{M}$ ТМ-ның қадамы мына қатынас $(q, t_1\, t_2 \ldots t_n, i) \models_M (q', t_1\, t_2 \ldots t_{i-1}\, z\, t_{i+1} \ldots t_n, i)$ ретінде жазуға болады, мұнда бастиек i ұяшығына z символын жазып, бұрынғы орнында жылжымай қалады деп есептелінеді.

Егер $i = n+1$, онда $\boldsymbol{M}$ машиналарының бастиегі арнаулы ϕ символын оқиды. Мұнда егер:

1) $\delta(q, \phi) = (q', z, \mid\!\leftarrow)$ болса, онда $\boldsymbol{M}$ ТМ-ның қадамын мына қатынас $(q, t_1\, t_2 \ldots t_n, n+1) \models_M (q', t_1\, t_2 \ldots t_{i-1}\, z\, t_{i+1} \ldots t_n, n)$ ретінде жазуға болады;

2) $\delta(q, \phi) = (q', z, \rightarrow\!\mid)$ болса, онда $\boldsymbol{M}$ ТМ-ның қадамын мына қатынас $(q, t_1\, t_2 \ldots t_n, n+1) \models_M (q', t_1\, t_2 \ldots t_{i-1}\, z\, t_{i+1} \ldots t_n, n+2)$ ретінде жазуға болады;

3) $\delta(q, \phi) = (q', z, \|)$ болса, онда $\boldsymbol{M}$ ТМ-ның қадамын мына қатынас $(q, t_1\, t_2 \ldots t_n, n+1) \models_M (q', t_1\, t_2 \ldots t_{i-1}\, z\, t_{i+1} \ldots t_n, n+1)$ ретінде жазуға болады.

Егер қандай ТМ туралы сөз болып жатқаны белгілі болса, онда $\models_A$ қатынасындағы $\boldsymbol{M}$ әріпін жазбауға болады.

Әдеттегідей, $\models$ қатынасының *k дәрежесін* анықтауға болады: егер берілген екі кескін C_1 және C_k $(k \ge 1)$ арасында $C_1, C_2, \ldots, C_k$

кескіндер тізбесі табылса және олардың арасында $C_1 \models C_2 \models ... \models C_k$ қатынастары орнатылса, онда осы ақиқатты былай $C_1 \models^k C_k$ жазуға болады.

Ары қарай, егер кейбір $\boldsymbol{\tau} \in \boldsymbol{T}^*$, $\alpha \in \boldsymbol{U}^*$ және кез келген $i \geq 1$ немесе $i \geq 0$ бүтін саны үшін $(q_s, \tau, 1) \models^i (q, \alpha, i)$ қатынасы орындалса, онда оны $(q_s, \tau, 1) \models^+ (q, \alpha, i)$ немесе $(q_s, \tau, 1) \models^* (q, \alpha, i)$ түрде жазуға болады. Мұнда $\models$ қатынасының транзитивті тұйықталуы $\models^+$ деп, ал оның рефлексивті және транзитивті тұйықталуы $\models^*$ деп белгіленген.

ТМ $\boldsymbol{M}$ танитын $\boldsymbol{L}(\boldsymbol{M})$ тілі былай анықталады:

$\boldsymbol{L}(\boldsymbol{M}) \rightleftharpoons \{\tau: \tau \in \boldsymbol{T}^* \ \& \ (q_s, \tau, 1) \models^* (q_f, \alpha, i) \ \& \ \alpha \in \boldsymbol{U}^* \& \ i \in \boldsymbol{N}\}$

$\boldsymbol{L}(\boldsymbol{M})$ тілін танитын $\boldsymbol{M}$ ТМ әрбір қолжетпек кірісте тоқтайды (яғни тізбені танып, жұмысын тоқтатады) деп есептеледі. Екінші жағынан, $\boldsymbol{M}$ ТМ тоқтамайды, егер кіріс тізбесі $\boldsymbol{L}(\boldsymbol{M})$ тіліне кірмесе.

III.4.1.1-мысал. $\{t^n b^n : n \geq 1\}$ тілін танитын ТМ:

$$\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{T}, \boldsymbol{U}, q_0, \boldsymbol{F}, |\!\!\leftarrow, \|, \rightarrow\!\!|, \delta>, \text{ мұнда}$$

$\boldsymbol{Q} = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, $\boldsymbol{T} = \{t, b\}$, $\boldsymbol{U} = \{t, b, \phi, x, y\}$, $\boldsymbol{F} = \{q_5\}$ және δ:

$$\delta(q_0, t) = (q_1, x, \rightarrow\!\!|),\ \delta(q_2, x) = (q_3, x, \rightarrow\!\!|),$$
$$\delta(q_1, t) = (q_1, t, \rightarrow\!\!|),\ \delta(q_3, \phi) = (q_5, y, \rightarrow\!\!|),$$
$$\delta(q_2, y) = (q_2, y, |\!\!\leftarrow),\ \delta(q_4, x) = (q_0, x, \rightarrow\!\!|),$$
$$\delta(q_3, y) = (q_3, y, \rightarrow\!\!|),\ \delta(q_1, b) = (q_2, y, |\!\!\leftarrow),$$
$$\delta(q_4, t) = (q_4, t, |\!\!\leftarrow),\ \delta(q_2, t) = (q_4, t, |\!\!\leftarrow),$$
$$\delta(q_1, y) = (q_1, y, \rightarrow\!\!|),$$

Кіріс тізбесі $\tau = ttbb$ тану үшін мына әрекеттерді орындау керек:

$$(q_0, ttbb, 1) \models (q_1, xtbb, 2)$$
$$\models (q_1, xtbb, 3) \models (q_2, xtyb, 2)$$
$$\models (q_4, xtyb, 1) \models (q_0, xtyb, 2)$$
$$\models (q_1, xxyb, 3) \models (q_1, xxyb, 4)$$
$$\models (q_2, xxyy, 3) \models (q_2, xxyy, 2)$$
$$\models (q_3, xxyy, 3) \models (q_3, xxyy, 4) \models$$
$$(q_3, xxyy, 5) \models (q_5, xxyyy, 6).$$

III.4.1.2-мысал. Мына $U(n)=n+1$ функциясын есептейтін ТМ былай беріледі $\boldsymbol{Q}=\{q_s, q_t, q_f\}$, $\boldsymbol{T} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, q_0 –

алғашқы күй, q_f – соңғы күй, ал ауысулар функциясы 1-кестемен беріледі.

1-кесте. $U(n)=n+1$ функциясын жүзеге асыратын пәрмендер тізбегі

Пәрмендер №	***a***	***b***
1	$\delta(q_s, 0) = (q_t, 1, \Vert)$	$\delta(q_t, 0) = (q_t, 0, \vert\!\leftarrow)$
2	$\delta(q_s, 1) = (q_t, 2, \Vert)$	$\delta(q_t, 1) = (q_t, 1, \vert\!\leftarrow)$
3	$\delta(q_s, 2) = (q_t, 3, \Vert)$	$\delta(q_t, 2) = (q_t, 2, \vert\!\leftarrow)$
4	$\delta(q_s, 3) = (q_t, 4, \Vert)$	$\delta(q_t, 3) = (q_t, 3, \vert\!\leftarrow)$
5	$\delta(q_s, 4) = (q_t, 5, \Vert)$	$\delta(q_t, 4) = (q_t, 4, \vert\!\leftarrow)$
6	$\delta(q_s, 5) = (q_t, 6, \Vert)$	$\delta(q_t, 5) = (q_t, 5, \vert\!\leftarrow)$
7	$\delta(q_s, 6) = (q_t, 7, \Vert)$	$\delta(q_t, 6) = (q_t, 6, \vert\!\leftarrow)$
8	$\delta(q_s, 7) = (q_t, 8, \Vert)$	$\delta(q_t, 7) = (q_t, 7, \vert\!\leftarrow)$
9	$\delta(q_s, 8) = (q_t, 9, \Vert)$	$\delta(q_t, 8) = (q_t, 8, \vert\!\leftarrow)$
10	$\delta(q_s, 9) = (q_s, 0, \vert\!\leftarrow)$	$\delta(q_t, 9) = (q_t, 9, \vert\!\leftarrow)$
11	$\delta(q_s, \phi) = (q_f, 1, \vert\!\leftarrow)$	$\delta(q_t, \phi) = (q_f, \phi, \Vert)$

Мұнда *a* тік жолы ағымдағы ұяшықтағы цифрдың бірге үлкейгенін жүзеге асырад ы. $\delta(q_s, 9) = (q_s, 0, \vert\!\leftarrow)$ пәрмені үлкен разрядқа бірліктің көшкенін есепке алады және мұнда q_s сақталады. Осы сақталған күйде кезекті ұяшықтағы цифр бірге үлкейтіледі. $\delta(q_s, \phi) = (q_f, 1, \vert\!\leftarrow)$ пәрмені көшіру нәтижесінде разрядтың бірге өскендігін есепке алады. Ал *b* тік жолындағы пәрмендер тізбесі нәтижені орналастыру ережесінің сақталалуын қамтамасыз етеді. Нәтижені орналастыру ережесі бойынша санды бірге үлкейткеннен кейін, барлық жазба бастиектің оң жағында болатындығын қадағалау керек.

Бейдетерминді ТМ-мен танылған ***L*** тілі қандай да болса детерминді ТМ-мен де танылады.

Егер ТМ-нда бастиекті таспаның кіріс тізбесі орналасқан аралығынан тыс жылжытуға тиім салынса, онда сызықты шенеуленген автомат шығады. Оған, кіріс таспасы символдар жиынына таспаның басын және таспаның соңын белгілеу үшін екі арнаулы символ ⊢ және ⊣ қосып қол жеткізуге болады. Егер ТМ-ның таспасы сол жақтан шектеулі болса, яғни ⊢ символы бар болса, онда тек ⊣ символын қоссақ жеткілікті. Таспада кіріс тізбесі осы екі символдың арасында орналасады және $\vdash\tau\dashv$ түрде болады. Сызықтық шенеуленген автоматтың ауысулар функциясы, егер ⊢ немесе ⊣ символы оқылса, онда бастиекті бір ұяшыққа оңға немесе солға жылжытады, ал қалған жағдайларда ТМ-ның функциясына ұқсас болады.

1. Әрине, бейдетерминді сызықтық шенеуленген автоматпен танылған кез келген тіл қандай да болса детерминді ТМ-мен де танылатындығы ақиқат.

III.4.1-тапсырмалар. Кіріс әліпбиі $\{a\}$ болатын мына функцияны есептейтін бейдетерминді ТМ тұрғызыңыз:

2. $f(a^n) = \varepsilon$.
3. $f(a^n) = a^{2n}$.
4. $\{a, aaa, aaaaaaa\}$.
5. $\{a\}^*$.
6. $\{a^{2n+1}: n\geq 0\}$.

III.4.1-сұрақтар. Төмендегі тіл ТМ-мен таныла ма:

1. $\{\tau c\tau : \tau\in\{a, b, c\}^*\}$?
2. $\{\tau\in\{a, b, c, d\}^* : |\tau|_a\neq|\tau|_b, |\tau|_c \neq|\tau|_d\}$?
3. $\{\tau\in\{a, b, c\}^* : |\tau|_a<|\tau|_b< |\tau|_c\}$?
4. $\{a^m b^n a^m b^n : m\geq 1 \ \&\ n\geq 1\}$;
5. $\{a^k b^m c^n : k\geq 1 \ \&\ m\geq 1 \ \&\ n\geq 1\}$;
6. $\{a^k b^k a^n b^n : k\geq 1 \ \&\ n\geq 1\}$;
7. $\{a^{k+n} b^{k-n}: k\geq 1 \ \&\ n\geq 1\}$;
8. $\{(a^k b^m)c^n : k\geq 1 \ \&\ m\geq 1 \ \&\ n\geq 1\}$;
9. $\{a^{k-m} b^m c^{n-m} : k\geq 1 \ \&\ m\geq 1 \ \&\ n\geq 1\}$;

V.3. Шенеуленбеген тілдердің қасиеттері

V.3.1. Шенеуленбеген грамматикалар мен Тьюринг машиналарының эквиваленттілігі

Бұл бөлімде шенеуленбеген тілдер қарастырылады, олар 0-типтегі грамматикаларға жататын шенеуленбеген грамматикалармен (ШГ) туындайды және Тьюринг машиналарымен (ТМ) танылады. ШГ мен ТМ-ның эквиваленттігі көрсетіледі. Шенеуленбеген тілдердің алгоритмдік проблемалары талқыланады. Мысаладар ұсынылады, тапсырмалар және сұрақтар қойылады [1-4,8,9,11-13,15-21,24-32].

Төменде берілген тіл ТМ-мен танылады тек қана сонда және тек қана сонда, қашан егер ол ШГ-мен туындаса екендігі дәлелденеді.

Жеткілікті дәлелдеу үшін грамматиканың шығарымын бейтерминді таңдап, осы шығарымның нәтижесі кіріс тізбесіне сай келетіндігін тексеретін бейтерминді ТМ-н тұрғызамыз. Егер сай келсе, онда кіріс тізбесі танылады.

Қажеттілікті дәлелдеу үшін алдымен терминалды тізбені тудыратын, сонан кейін осы тізбеде ТМ-н моделдейтін грамматиканы тұрғызамыз. Егер тізбе ТМ-мен танылса, онда тізбе өзі бейнелейтін терминал символдарға түрленеді.

V.3.1-лемма. Егер $\boldsymbol{L}$ тілін жалпы түрдегі грамматика $\boldsymbol{G}$ тудыратын болса, яғни $\boldsymbol{L}=\boldsymbol{L}(\boldsymbol{G})$, онда осы тілді танитын ТМ $\boldsymbol{M}$ бар болады, яғни $\boldsymbol{L} = \boldsymbol{L}(\boldsymbol{M})$.

Дәлелдеу. Айталық, жалпы түрдегі $\boldsymbol{G} = <\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, S>$ грамматикасы $\boldsymbol{L}$ тілін тудырсын, яғни $\boldsymbol{L}(\boldsymbol{G})= \boldsymbol{L}$. Осы $\boldsymbol{L}$ тілін танитындай бейдетерминді ТМ $\boldsymbol{M}$ тұрғызылады, яғни $\boldsymbol{L}(\boldsymbol{M}) = \boldsymbol{L}$.

Мейлі ТМ $\boldsymbol{M} = < \boldsymbol{Q}, \boldsymbol{U}, \boldsymbol{T}, q_0, \boldsymbol{F}, \vdash, \leftarrow\!\!|, \|, \rightarrow\!|, \delta>$ берілсін, мұнда $\boldsymbol{U} = \boldsymbol{N} \cup \boldsymbol{T} \cup \{\#, \phi\}$, мұнда # – тізбелерді ажырату үшін арналған символ, ϕ – бос ұяшықтарды толтыру үшін арналған символ, және $\#, \phi \notin \boldsymbol{N} \cup \boldsymbol{T}$.

Алдымен $\boldsymbol{M}$ ТМ-ның таспасы $\tau \in \boldsymbol{T}^*$ тізбесін қамтиды. Сонан кейін ол τ тізбесінің барлық символдарын бір ұяшыққа оңға жылжытып, τ-дың алдындағы босаған орынға # символын қояды, ал τ тізбесінен кейін таспаның ішіндегісі $\#\tau\#S\#$ болатындай етіп $\#S\#$ тізбесі қойылады. Осы мезгілден бастап ТМ $\boldsymbol{M}$ бейтерминал S символынан бастап $\boldsymbol{G}$ грамматикасындағы шығарымды моделдей бастайды. Грамматиканың шығарымындағы әрбір сентенциялық пішін екі ажыратқыш # символының арасында кезекпен пайда болады. Егер қайсы бір қозғалыс терминал тізбесіне алып келсе, онда ол τ тізбесімен салыстырылады. Егер екі тізбе тең болса, онда $\boldsymbol{M}$ ТМ τ тізбесін таниды.

Дәлірек айтсақ, мейлі қайсы бір мезгілде ТМ $\boldsymbol{M}$ өзінің таспасында $\#\tau\#A_1A_2...A_k\#$ түрдегі тізбені қамтысын. Алдымен машина өзінің бастиегін $A_1A_2...A_k$ тізбесінде i позициясын және $\boldsymbol{P}$ жиынындағы ережелердің сол жағының ең үлкен ұзындығына тең r константасын бейдетерминді таңдап, жылжытады. Кейін, машина $A_iA_{i+1}...A_{i+r-1}$ іштізбесін зерттейді: егер бұл іштізбе $\boldsymbol{P}$ жиынындағы қайсы бір ереженің сол жағы болса, онда оны осы ереженің оң жағымен ауыстыруға болады; егер ереженің оң жағының ұзындығы r-ге тең болмаса, онда машина орынды босату немесе толтыру үшін $A_{i+r}A_{i+r+1}...A_k\#$ іштізбесін иә солға, иә оңға жылжытуға мәжбүр болады. Оңға жылжыған кезде босаған орынды уақытша толтыру үшін ϕ символы пайдаланылады.

Осы $\boldsymbol{G}$ грамматикасындағы шығарымды қарапайым моделдеуден ТМ $\boldsymbol{M}$ өзінің таспасына $\#\tau\#\lambda\#$ түрдегі тізбені жазатындығы көрінеді, мұнда $\lambda\in \boldsymbol{T}^*$ дәл сонда, қашан $S\Rightarrow^*_{\boldsymbol{G}}\lambda$. Сонымен қатар, егер $\lambda=\tau$, онда $\boldsymbol{M}$ ТМ $\boldsymbol{L}$ тілін таниды, яғни $\boldsymbol{L}=\boldsymbol{L}(\boldsymbol{M})$. Керек дәлелдеу осы.

V.3.2-лемма. Егер $\boldsymbol{L}$ тілін ТМ $\boldsymbol{M}$ танитын болса, яғни $\boldsymbol{L}=\boldsymbol{L}(\boldsymbol{M})$, онда осы тілді тудыратын жалпы түрдегі грамматика $\boldsymbol{G}$ бар болады, яғни $\boldsymbol{L}=\boldsymbol{L}(\boldsymbol{G})$.

Дәлелдеу. Айталық, $\boldsymbol{M}=<\boldsymbol{Q}, \boldsymbol{U}, \boldsymbol{T}, q_0, \boldsymbol{F}, \vdash, \leftarrow\!\!\!\shortmid, \|, \rightarrow\!\!\!\shortmid, \delta>$ ТМ $\boldsymbol{L}$ тілін танысын, яғни $\boldsymbol{L}(\boldsymbol{M})=\boldsymbol{L}$. Енді алдымен $\boldsymbol{T}^*$ жиынындағы қайсы бір тізбенің екі көшірмесін бейдетерминді тудыратын, кейін осы көшірмелердің біреуінде $\boldsymbol{M}$ ТМ-ның әрекеттерін моделдейтін жалпы түрдегі грамматика $\boldsymbol{G}$ тұрғызамыз. Егер $\boldsymbol{M}$ ТМ тізбені танитын болса, онда $\boldsymbol{G}$ грамматикасы екінші көшірмені терминалды тізбеге айналдырады. Егер $\boldsymbol{M}$ ТМ тізбені танымайтын болса, онда $\boldsymbol{G}$ грамматикасындағы шығарым ешқашан терминалды тізбеге әкелмейді.

Талқылау жалпылылығын жоғалтпай, әрбір $q\in\boldsymbol{F}$ және $t\in\boldsymbol{T}$ үшін $\delta(q, t)$ мәні анықталмағандығын жорамалдаймыз.

Дәлірек, мейлі ШГ $\boldsymbol{G} = \langle\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{P}, A_1\rangle$ берілсін,

Мұнда $\boldsymbol{N} =\{[t, u] : t\in\boldsymbol{T}\cup\{\varepsilon\} \ \& \ u\in\boldsymbol{U}\}\cup\boldsymbol{Q}\cup\{A_1, A_2, A_3\}$, ал $\boldsymbol{P}$ мынаны қамтиды:

(1) $A_1\rightarrow q_0A_2$;

(2) $A_2 \rightarrow[t, t]A_2$ әрбір $t\in\boldsymbol{T}$ үшін;

(3) $A_2 \rightarrow A_3$;

(4) $A_3 \rightarrow[\varepsilon, B]A_3$;

(5) $A_3 \rightarrow\varepsilon$;

(6) $q[t, c] \rightarrow[t, d]\ p$ тек $\delta(q, c) = (p, d, \rightarrow\!\mid)$ болатындай әрбір $q\in\boldsymbol{Q}$, $t\in\boldsymbol{T}\cup\{\varepsilon\}$, $c\in\boldsymbol{U}$, $d\in\boldsymbol{U}$, $p\in\boldsymbol{Q}$ үшін;

(7) $[u, e]\ q[t, c] \rightarrow p[u, e]\ [t, d]$ тек $\delta(q, c) = (\ p, d, \mid\!\leftarrow)$ болатындай әрбір $u\in\boldsymbol{T}\cup\{\varepsilon\}$, $e\in\boldsymbol{U}$, $q\in\boldsymbol{Q}$, $t\in\boldsymbol{T}\cup\{\varepsilon\}$, $c\in\boldsymbol{U}$, $p\in\boldsymbol{Q}$, $d\in\boldsymbol{U}$ үшін;

(8) $[t, c]q \rightarrow qtq$, $q[t, c] \rightarrow qtq$, $q \rightarrow\varepsilon$ әрбір $t\in\boldsymbol{T}\cup\{\varepsilon\}$, $c \in\boldsymbol{U}$, $q\in\boldsymbol{F}$ үшін.

Ары қарай жоғарыдағы 1-ші және 2-ші ережені қолданып, кейбір $t_i\in\boldsymbol{T}$, $1\leq i\leq k$ үшін мына түрдегі шығарымды аламыз

$$A_1\Rightarrow^* q_0[t_1, t_1]\ [t_2, t_2] \ldots [t_k, t_k]A_2$$

Айталық $\boldsymbol{M}$ ТМ $t_1t_2\ldots t_k$ тізбегін танысын, сонда кейбір m үшін ол өзінің кірісінің оң жағынан саны m-нен аспайтын ұяшықтарды пайдаланады. Алдымен 3-ші ережені, кейін m рет 4-ші ережені, соңынан 5-ші ережені пайдаланып, мына шығарымды аламыз

$$A_1\Rightarrow^* q_0[t_1, t_1]\ [t_2, t_2] \ldots [t_k, t_k]\ [\varepsilon, B]^m$$

Ары қарай танылатын күй тумайынша тек 6-ші және 7-ші ережелер ғана қолданыла алатындығын ескертеміз. Осында $[(\boldsymbol{T}\cup\{\varepsilon\})\times\boldsymbol{U}]$ бейтерминалдарды белгілейтін таспалық символдардың бірінші компоненттері ешқашан өзгермейді, ал екіншілері $\boldsymbol{M}$ ТМ-ның таспасында өндірілетін жазбаларды моделдейді.

$\boldsymbol{M}$ ТМ-ның қадам саны k бойынша индукция арқылы мынаны көрсете аламыз: егер $(q_0, t_1t_2\ldots t_k, 1) \models_{\boldsymbol{M}}^* (q, X_1X_2\ldots X_s, r)$, онда

$$q_0[t_1, t_1]\ [t_2, t_2] \ldots [t_k, t_k]\ [\varepsilon, B]^m \Rightarrow_{\boldsymbol{G}}^*$$

$$[t_1, X_1][t_2, X_2]...[t_{r-1}, X_{r-1}]\ q[t_r, X_r]...[t_{k+m}, X_{k+m}],$$

мұнда $t_1, t_2, ..., t_k \in \boldsymbol{T}$; $t_{k+1}=t_{k+2}=...=t_{k+m}=\varepsilon$;

$X_1, X_2, ..., X_{k+m} \in \boldsymbol{U}$; $X_{s+1}=X_{s+2}=...=X_{k+m}=\phi$.

Индукция базасы. Мейлі қадам саны $k = 0$, онда тұжырым ақиқат.

Индукциялық гипотеза. Айталық тұжырым барлық *k-1* қадамдары үшін орындалады.

Индукциялық өту. Мейлі ТМ $\boldsymbol{M}$ келесі k қадамдарын орындайды:

$$(q_0, t_1t_2...t_k, 1) \vdash_{\boldsymbol{M}}^*$$
$$(q_1, X_1, X_2, ..., X_r, j_1) \vdash_{\boldsymbol{M}}^*$$
$$(q_2, Y_1Y_2...Y_s, j_2).$$

Сонда алғашқы k қадамға қолданылған индукциялық гипотеза бойынша мына түрдегі шығарым бар болады

$$q_0[t_1, t_1][t_2, t_2]...[t_k, t_k][\ \varepsilon, B]^m \Rightarrow_{\boldsymbol{G}}^*$$
$$[t_1, X_1][t_2, X_2]...\ q_1[t_{j1}, X_{j1}]\ ...\ [t_{k+m}, X_{k+m}]$$

Соңғы қадам бойынша қарасақ $\delta(q_1, X_{j1}) = (\ q_2, Y_{j1}, d)$ міндетті болу керек және осында $d = \leftarrow$, егер $j_2 = j_1$-1 немесе $d = \rightarrow$, егер $j_2 = j_1 + 1$. Сәйкесінше, $d = \rightarrow$ болғанда, мына түрдегі 6-ші ереже бар:

$$q_1[t_{j1}, X_{j1}] \rightarrow [t_{j1}, Y_{j1}]q_2,$$

ал $d = \leftarrow$ болғанда, мына түрдегі 7-ші ереже бар:

$$[t_{j1-1}, X_{j1-1}]q_1[t_{j1}, X_{j1}] \rightarrow q_2\ [t_{j1-1}, X_{j1-1}][t_{j1}, Y_{j1}].$$

Енді $X_i = Y_i$ барлық $i \neq j_1$ үшін. Сонымен, тағы бір қадам шығарым береді.

$$q_0[t_1, t_1][t_2, t_2]...[t_k, t_k][\ \varepsilon, B]^m \Rightarrow_{\boldsymbol{G}}^*$$
$$[t_1, Y_1][t_2, Y_2]...q_2[t_{j2}, Y_{j2}]\ ...[t_{k+m}, Y_{k+m}],$$

бұл индукцияның болжамын дәлелдейді. Ары қарай, егер $q \in \boldsymbol{F}$, онда грамматика ережесінің 8-ші түрі бойынша мына шығарымды алуға болады

$$X_1][t_2, X_2]\ ...\ q[t_j, X_j]\ ...\ [t_{k+m}, X_{k+m}] \Rightarrow_{\boldsymbol{G}}^*$$
$$q\ t_1\ q\ t_2\ q...\ q\ t_k\ q \Rightarrow_{\boldsymbol{G}}^*$$
$$t_1t_2\ ...\ t_k$$

Сонымен, $\boldsymbol{G}$ грамматикасы $t_1t_2 \ldots t_k$ тізбесін тудыра алады, егер $t_1t_2 \ldots t_k$ тізбесі $\boldsymbol{M}$ ТМ-мен танылса, яғни егер $t_1t_2 \ldots t_k$ тізбесі $\boldsymbol{M}$ ТМ-мен танылса, онда $t_1t_2 \ldots t_k$ тізбесі $\boldsymbol{L}(\boldsymbol{G})$ тіліне кіреді.

Дәлелдеуді аяқтау үшін, $\boldsymbol{L}(\boldsymbol{G})$ тілінің барлық тізбелері $\boldsymbol{M}$ ТМ-мен танылатындығын көрсету керек.

Егер $A_1 \Rightarrow_{\boldsymbol{G}}^* \tau$ болса, онда τ ТМ-мен танылатындығы индукциямен дәлелденеді.

Ескере кетсек, кез келген шығарым, соның ішінде $A_1 \Rightarrow_{\boldsymbol{G}}^* \tau$ шығарымы да, тек 1-ші – 5-ші ережелермен басталып, мынадай түрде нәтиже береді

$$A1 \Rightarrow_{\boldsymbol{G}}^* q_0[t_1, t_1][t_2, t_2] \ldots [t_k, t_k][\varepsilon, B]^m.$$

Ары қарай 6-ші және 7-ші ережелерді қолданылып, мынадай нәтиже аламыз

$$[t_1, X_1][t_2, X_2] \ldots q_1[t_{j1}, X_{j1}] \ldots [t_{k+m}, X_{k+m}],$$

мұнда $q_1 \in \boldsymbol{F}$, сонан кейін 8-ші ережелерден $\tau = t_1t_2 \ldots t_k$ шығады.

ТМ $\boldsymbol{M}$ моделденетін шығарымның кесіндісі мынау

$$q_0[t_1, t_1][t_2, t_2] \ldots [t_k, t_k]\ [\varepsilon, B]^m \Rightarrow_{\boldsymbol{G}}^* [t_1, Y_1][t_2, Y_2] \ldots q[t_j, Y_j] \ldots [t_{k+m}, Y_{k+m}],$$

мұнда $q \in \boldsymbol{F}$к. Егер осындай шығарым бар болса, онда $\boldsymbol{M}$ ТМ-ның мына түрдегі қадамы бар болады

$$(q_0, t_1, t_2, \ldots, t_k, 1) \vdash_{\boldsymbol{M}}^* (q, Y_1Y_2 \ldots Y_{k+m}, j).$$

Тұжырымды шығарым ұзындығы $\boldsymbol{L}$ бойынша дәлелдейік.

Индукция базасы. Егер $\boldsymbol{L}= 0$, онда тұжырым анық орындалады.

Индукциялық гипотеза. Айталық, тұжырым $l=n$ $(n \geq 0)$ үшін орындалсын.

Индукциялық өту. Мейлі, ұзындығы $\boldsymbol{L}= n+1$ шығарым болсын. Жалпы жағдайда

$$q_0[t_1, t_1][t_2, t_2] \ldots [t_k, t_k]\ [\varepsilon, B]^m \Rightarrow_{\boldsymbol{G}}^*$$
$$[t_1, X_1][t_2, X_2] \ldots q_1[t_{j1}, X_{j1}] \ldots [t_{k+m}, X_{k+m}] \Rightarrow_{\boldsymbol{G}}^*$$
$$[t_1, Y_1][t_2, Y_2] \ldots q[t_j, Y_j] \ldots [t_{k+m}, Y_{k+m}],$$

мұнда $q \in \boldsymbol{F}$. Индукциялық гипотезаға сай мына ауысу бар

$$(q_0, t_1, t_2, \ldots, t_k, 1) \vdash_{\boldsymbol{M}}^* (q_1, X_1X_2 \ldots X_{k+m}, j_1).$$

Шығарымның соңғы қадамы ереженің тек 6-ші немесе 7-ші түрі арқылы орындала алады. Егер ереженің 6-ші түрі қолданылса, онда $j = j_1+1$; егер ереженің 7-ші түрі қолданылса, онда $j = j_1 - 1$.

Бұл ережелер $\delta(q_1, X_{j1}) = (q, Y_{j1}, d)$ болғандықтан бар, мұнда $d = \rightarrowtail$, егер $j = j_1+1$ немесе $d = \leftarrowtail$, егер $j = j_1 - 1$.

Бұл жерде $X_i = Y_i$ барлық $i \neq j_1$ үшін. δ функциясының осы мәндері арқасында ТМ $\boldsymbol{M}$ танитын кескінге аударатын тағы да бір қадам жасайды:

$$(q_0, t_1t_2...t_k, 1) \vdash_{\boldsymbol{M}}^* (q_1, X_1X_2...X_{k+m}, j_1) \vdash_{\boldsymbol{M}}^* (q, Y_1Y_2 ...Y_{k+m}, j),$$

мұнда $q \in \boldsymbol{F}$.

Басқаша айтқанда, $\tau = t_1t_2...t_k$ тізбесі $\boldsymbol{M}$ ТМ-мен танылатындығы көрсетіледі.

Сонымен, егер $\tau \in \boldsymbol{L}(\boldsymbol{G})$, онда τ тізбесі $\boldsymbol{M}$ ТМ-мен танылады.

V.3.1-тапсырма. Мына $\boldsymbol{L}(\boldsymbol{G})$ тілін тудыратын грамматикаға эквивалентті ТМ $\boldsymbol{M} = <\boldsymbol{Q}, \boldsymbol{U}, \boldsymbol{T}, q_0, \boldsymbol{F}, \vdash, \leftarrowtail, \|, \rightarrowtail, \delta>$ тұрғызыңыз:

1. $\boldsymbol{L}(\boldsymbol{G}) = \{a^mb^na^mb^n : m\geq 1 \ \& \ n\geq 1\}$?
2. $\boldsymbol{L}(\boldsymbol{G}) = \{a^kb^mc^n : k\geq 1 \ \& \ m\geq 1 \ \& \ n\geq 1\}$?
3. $\boldsymbol{L}(\boldsymbol{G}) = \{a^{k+n}b^{k-n}: k\geq 1 \ \& \ n\geq 1\}$?
4. $\boldsymbol{L}(\boldsymbol{G}) = \{\varepsilon\}$;
5. $\boldsymbol{L}(\boldsymbol{G}) = \{a^{2n}: n\geq 1\}$;
6. $\boldsymbol{L}(\boldsymbol{G}) = \{a, aaa, aaaaaaa\}$;
7. $\boldsymbol{L}(\boldsymbol{G}) = \{a\}^*$;

V.3.1-сұрақтар: ТМ мына тілдерін тани ма:

1. $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{ab^nc : n\geq 0\}$, тудыратын грамматика $\boldsymbol{N} = \{A, B, C\}$, $\boldsymbol{T}=\{a,b,c\}$, $\boldsymbol{P}=\{S\to aB, B\to aB, B\to b, B\to bC, C\to c, C\to cC, S\to aC, S\to a\}$.
2. $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{(abc)^n: n\geq 1\}$, тудыратын грамматика $\boldsymbol{N}=\{A,B,C\}$, $\boldsymbol{T} = \{a,b,c\}$, $\boldsymbol{P} = \{S\to aSBC|abC, CB\to BC, bB\to bb, bC\to bc, cC\to cc\}$
3. $\boldsymbol{L}(\boldsymbol{G}) \rightleftharpoons \{c^nb^{n+1} : n\geq 0\}$, тудыратын грамматика $\boldsymbol{N} = \{A,S\}$, $\boldsymbol{T}=\{a,b\}$, $\boldsymbol{P} = \{S\to A, A\to b, A\to cA\}$.
4. ТМ-нда алғашқы деректер қалай беріледі?
5. ТМ-нда нәтижелі деректер қалай беріледі?
6. ТМ-нда нбасталу және аяқталу ережелері қалай беріледі?

V.3.2. Шенеуленбеген тілдердің алгоритмдік проблемалары

Бұл параграфта шенеуленбеген тілдердің алгоритмдік проблемалары талқыланады. Шенеуленбеген грамматикаларда және Тьюринг машиналарында бар алгоритмдік шешілмейтін проблемалар келтіріледі. Сонымен қатар тапсырмалар беріледі және сұрақтар қойылады [1-4,8,9,11-13,15-21,24-32].

Басқа топтағы грамматикалардағы сияқты ШГ үшін келесі алгоритмдік проблемаларды талқылаймыз:

1. *Бостық проблемасы* – берілген $\boldsymbol{G}$ ШГ үшін $\boldsymbol{L}(\boldsymbol{G})$ бос тіл бола ма, яғни , $\boldsymbol{L}(\boldsymbol{G}) = \varnothing$?
2. *Шексіздік проблемасы* – берілген $\boldsymbol{G}$ ШГ үшін $\boldsymbol{L}(\boldsymbol{G})$ шексіз тіл бола ма?
3. *Тиістілік проблемасы* – кез келген ξ тізбе берілген $\boldsymbol{G}$ ШГ-нан туындайтын $\boldsymbol{L}(\boldsymbol{G})$ тіліне тиісті бола ма, яғни $\xi\in\boldsymbol{L}(\boldsymbol{G})$?
4. *Эквиваленттілік проблемасы* – кез келген екі $\boldsymbol{G_i}$ және $\boldsymbol{G_j}$ ($\boldsymbol{i\neq j}$) грамматикалар эквивалентті бола ма, яғни $\boldsymbol{L}(\boldsymbol{G_i})=\boldsymbol{L}(\boldsymbol{G_j})$?
5. *Тұйықтылық проблемасы* – белгілі бір типтегі кез келген екі тілге жиындық амалдарды қолданғанда нәтижесі сол топқа жататындығын анықтау.

Басқа топтағы автоматтардағы сияқты ТМ үшін келесі алгоритмдік проблемаларды талқылаймыз:

1. *Бостық проблемасы* – берілген $\boldsymbol{M}$ ТМ үшін $\boldsymbol{L}(\boldsymbol{M})$ бос тіл бола ма, яғни , $\boldsymbol{L}(\boldsymbol{M}) = \varnothing$?
2. *Шексіздік проблемасы* – берілген $\boldsymbol{M}$ ТМ үшін $\boldsymbol{L}(\boldsymbol{M})$ шексіз тіл бола ма?
3. *Тиістілік проблемасы* – кез келген ξ тізбе берілген $\boldsymbol{M}$ ТМ танитын $\boldsymbol{L}(\boldsymbol{M})$ тіліне тиісті бола ма, яғни $\xi\in\boldsymbol{L}(\boldsymbol{M})$?
4. *Эквиваленттілік проблемасы* – кез келген екі $\boldsymbol{M_i}$ және $\boldsymbol{M_j}$ ($\boldsymbol{i\neq j}$) автоматтар эквивалентті бола ма, яғни $\boldsymbol{L}(\boldsymbol{M_i})=\boldsymbol{L}(\boldsymbol{M_j})$?

5. *Тұйықтылық проблемасы* – белгілі бір типтегі кез келген екі тілге жиындық амалдарды қолданғанда нәтижесі сол топқа жататындығын анықтау.

6. *Өз-өзіне қолдану проблемасы* – өзінің формалды жазбасы болатын деректерде жұмысын сәтті аяқтайтын ***M*** ТМ бар ма жоқ па?

7. *Тоқтату проблемасы* – берілген ***M*** ТМ және берілген кіріс деректері үшін ТМ жұмысы қашан да бір тоқтай ма жоқ па?

Осы проблемалардың кейбіреулерінің алгоритмдік шешілмейтіндігін дәлеледеу келесі теоремадан туады:

V.3.1-теорема. ТМ ***M*** –ның өз-өзіне қолдану проблемасын шешетін ТМ ***M*** жоқ.

V.3.2-теорема. Кез келген ТМ ***M*** үшін тоқтату проблемасын шешетін ТМ $\boldsymbol{M}_0$ жоқ.

Осыдан, ТМ үшін өз-өзіне қолдану мен тоқтату проблемасы алгоритмді шешілмейді, яғни нәтижелікті анықталу проблемамы шешілмейді.

V.3.2-тапсырмалар:

1. ТМ үшін алгоритмдік проблемаларды атаңыздар.
2. ТМ үшін алгоритмдік шешілмейтін проблемаланы көрсетіңіз.
3. «Әрбір алгоритм Тьюринг машинасымен жүзеге асырылады» деген Тьюринг тезисінің мағынасын анықтаңыз.
4. ТМ ***M*** –ның өз-өзіне қолдану проблемасын түсіндіріңіз.
5. ТМ ***M*** –ның тоқтату проблемасын түсіндіріңіз

V.3.2-сұрақтар:

1. ТМ-нда алғашқы деректер қалай беріледі?
2. ТМ-нда мүмкін нәтижелер қалай беріледі?
3. ТМ-нда аралық нәтижелер қалай беріледі?
4. ТМ-нда жұмысты бастау ережесі қалай беріледі?
5. ТМ-нда жұмысты аяқтау ережесі қалай беріледі?
6. ТМ-нда нәтижелер қалай шығарылады?

ӘДЕБИЕТТЕР ЖӘНЕ ИНТЕРНЕТ РЕСУРСТАР

1. Ахо А., Ульман Дж. Синтаксистік талдау, аудару мен компиляциялау. Т. 1: Синтаксистік анализ. М.: Мир, 1978. - 612 б.
2. Бильгаева Н.Ц. Алгоритмдер, формалды тілдер, грамматикалар және автоматтар теориясы: Оқу құралы. Улан-Удэ: ШСМТУ, 2000.
3. Богаченко Н.Ф., Файзуллин Р.Т. Автоматтар, грамматикалар, алгоритмдер. Оқу құралы. Омбы. Диалог-Сибирь. 2006. -106 с.
4. Братчиков И. Л. Программалау тілдерінің синтаксисі. М.: Мир, 1975. - 232 с.
5. Брауэр В. Автоматтар теориясына кіріспе. -М.: Радио и связь, 1987.
6. Волкова И.А., Руденко Т.В. Формалды грамматикалар және тілдер. Трансляция теориясының элементтері. Екінші баспа (өңделген және толықтырылған) М: ЕМ ММУ, 1998. -62 б.
7. Гинзбург С. Контекстісіз тілдердің математикалық теориясы. М.: Мир, 1970. - 326 б.
8. Гладкий А. В. Формалды грамматикалар мен тілдер. М.: Ғылым, 1973. - 368 б.
9. Гладкий А. В., Мельчук И. А. Математикалық лингвистика элементтері. М.: Наука, 1969. - 192 б.
10. Глушков В.М. Цифрлық автоматтарды синтездеу. М.: Физматгиз, 1962. - 275 б.
11. Гросс М., Лантен А. Формалды грамматикалар теориясы. М.: Мир, 1971. - 294 б.
12. Касьянов В. Н. Формалды тілдер, автоматтар және есептеу күрделігі теориясы бойынша лекциялар. - Новосібір: НМУ, 1995. - 112 б.
13. Крючкова Е.Н. Формалды тілдер және автоматтар теориясы. - Барнаул; 1996.
14. Куратовский К., Мостовский А. Жиындар теориясы / А.Д. Тайманов редакциясымен ағылшыншадан аударған М. И. Кратко. — М.: Мир, 1970. — 416 б.

15. Мелихов А.Н., Кодачигов В.И. Алгоритмдер және формалды тілдер теориясы. - Таганрог; 1983.

16. Пентус А. Е., Пентус М. Р. Формалды тілдер теориясы: Оқу құралы. - М.: М.В.Ломоносов атындағы ММУ, 2004. - 80 б.

17. Пентус А. Е., Пентус М. Р. Формалды тілдердің математикалық теориясы: - М.: Ақпараттық технологиялар интернет университеті, Бином, Білімдер зертханасы, 2006. - 248 б.

18. Рейуорд-Смит В. Дж. Формалды тілдер теориясы. М.: Радио и связь, 1988. - 128 б.

19. Робин Хантер. Компиляторлардың негізгі тұжырымдамасы. The Essence of Comp*i*lers. — М.: «Вильямс», 2002. - б. 256

20. Саломаа А. Формалды грамматикалар теориясының меруеттері. М.: Мир, 1986. - 159 б.

21. Соколов В. А., Кушниренко О. Б., Бадин Н. М. Формалды тілдер мен грамматикалар. Есептер мен тапсырмалар. Ярославль: Ярославский мемлекеттік университеті, 1993. - 55 б.

22. Столл Р. Р. Жиындар. Логика. Аксиоматикалық теориялар. / Ю. А. Шихановича редакциясымен ағылшыншадан аударған Ю. А. Гастев және И. Х. Шмаин. — М.: Просвещение, 1968. — 232 б.

23. Трахтенброт Б. А., Барздинь Я. М. Ақырлы автоматтар (жұмысы және синтез). М.: Наука, 1970. - 400 б.

24. Хомский Н., Миллер Дж. Табиғи тілдерді формалды талдауға кіріспе // А.А.Ляпунов және О.Б.Лупанов редакциялаған. - М.: Мир, 1965.

25. Хопкрофт Дж. Э., Мотвани Р., Ульман Дж. Д. Автоматтар, тілдер мен есептеу теориясына кіріспе, 2-ші басылым. М.: Вильямс, 2002. - 528 б.

26. Rabin, M. O.; Scott, D. (April1959). "Finite Automata and Their Decision Problems". IBM Journalof Research and Development 3 (2): 114–125. doi: 10.1147/rd.32.0114. Retrieved 2007-03-15.

27. https: //ru.wikipedia.org/wiki/Формальная_грамматика

28. http: //www.intuit.ru/studies/courses/...

29. http: //window.edu.ru/library/pdf2txt/905/27905/11127.

30. http: //www.cs.odu.edu/~toida/nerzic/390teched/web_course.html/ Introduction to Theoretica***L*** Computer Science Web Course.

31. http: //courses.cs.vt.edu/~cs4114/lectures/Formal Languages and Automata Theory Cource Lecture notes.

32. http: //www.cis.ksu.edu/~stough/forlan/book-and-slides.html/An Introduction to Formal Language Theory.

33. https: //ru.wikipedia.org/wiki/Множество.

34. http: //en.wikipedia.org/wiki/Automata_theory.

А. Ә. ШӘРІПБАЙ

ТІЛДЕР МЕН АВТОМАТТАР ТЕОРИЯСЫ

Оқулық

(Екінші басылым. Өңделген және толықтырылған)

Пішімі 60х80 1/8 Ағартылған қағаз,
Тығыздығы 80 гр./ м2. 95%.
РИЗО басылымы.
Шартты баспа таб. 12,5. Көлемі 201 бет.

Оқулық мазмұнына баспа бөлімі жауап бермейді

"Эверо" баспада басылымға деген
баптаулы және басып шығарылды.
Қ Р, Алматы, Байтұрсынұлы көшесі, 22
Тел.: +7 /727/ 233 83 89, 233 83 43,
233 80 45, 233 80 42
e-mail: evero08mail.ru

ШӘРІПБАЙ Алтынбек Әмірұлы, техника ғылымдарының докторы, профессор, Халықаралық ақпараттандыру академиясының академигі, ҚР мемлекеттік сыйлығының лауреаты, Л.Н.Гумилев атындағы ЕҰУ «Жасанды интеллект» ҒЗИ-ның директоры (www.e-zerde.kz). Оның айналысатын ғылыми саласы информатика мен ақпараттық технологиялардың теориялық және қолданбалы проблемалары: программалаудың теориясы мен технологиялары, автоматтандырылған жүйелер, жасанды интеллект, компьютерлік лингвистика және электрондық оқыту. Ол бұл салада алмастырымдық және логикалық программалау тілдерінің семантикаларын формалдау, программалық және аппараттық құралдарды верификациялау әдістерін жасады. Осы зерттеулер нәтижелері негізінде ол «05.13.13 (05.13.11) – Есептеу машиналары, жүйелері және торлары (Математикалық, программалық және техникалық қамтымдар)» мамандығы бойынша «Есептеу машиналары мен жүйелерінің программалық және аппараттық құралдарын верификациялау» тақырыбында докторлық диссертация қорғады. Оның кейбір ғылыми нәтижелері ірі ғылыми орталықтарында ендірілді: 1976-1979 жылдары - «Ғарыштық жүйелерді имитациялау тілінің трансляторы» Ұшу-сынау институты, Жуковск қ; 1988-1989 жылдары - «Цифрлық сұлбаларды верификациялау жүйесі» Ғылыми-өндіріс орталығы, Зеленоград қ.; 1990-1991 жылдары - «Көппроцессорлы есептеу комплексі үшін параллелді программалау жүйесі» Электрондық есептеу техникасының ғылыми-зерттеу орталығы, Мәскеу қ.

Ол 400-ден астам ғылыми мақалалар, 4 оқулық, 12 оқу құралдарын және 4 монография жариялады; информатика мен есептеу техникасы бойынша 6 терминологиялық және түсіндірме сөздіктер шығарды; 30- нан астам зерделі меншіктің мемлекеттік тіркелу куәліктерін алды; көптеген мемлекеттік стандарттарды жасауға қатысты: 10 – ақпараттық және коммуникациялық технологиялар саласында, 9 – білім беру саласында.

А. Шәріпбай «Информатика, есептеу техникасы және басқару» мамандықтар тобы бойынша 4 ғылым докторын, 8 ғылым кандидатын және 6 PhD докторын даярлап шығарды. Оның ғылыми мектебі қазақ тілінің математикалық теориясын құрды, қазақ тілінің ауызша және жазбаша сөздері мен сөйлемдерін автоматтандырылған талдау мен синтездеу әдістерін жасады, электрондық оқыту басылымын даярлаудың технологиясын ұсынды. Бұл ғылыми нәтижелер 1994-2014 жылдары ҚР әртүрлі мемлекеттік және мемлекеттік емес құрылымдар мен ұйымдарда ендірілген көптеген электронды оқыту басылымын, қазақ тілін қашықтан оқыту жүйесін, қазақша сөйлеуді тану мен синтездеу жүйесін және басқа автоматтандырылған жүйелерді, солардың ішінде есепке алу және сараптамалық жүйелерді, жасау үшін қолданылды.