*Алимбекова Салтанат Нурлановна «Автоматтандыру және басқару» мамандығының студенті Омаров Бауыржан Султанович аға оқытушысы Әл-Фараби атындағы Қазақ Ұлттық университеті, Алматы қ.*

БІР КЛАССТЫҢ КОНТЕКСТТІ-ЕРКІН ТІЛДЕРІН ТАЛДАУ ҮШІН ДЕТЕРМИНДЕЛГЕН АВТОМАТ ҚҰРУ

The article outlines the basic concepts, methods and algorithms of the theory of formal languages - a science that studies mathematical models of languages and has a practical orientation on designing software support for the intelligent languages of interfaces for humancomputer communication, as well as store automata that allow solving for context-free languages the task of recognition, which consists in the fact that, according to a given chain, it is necessary to determine whether it belongs to a given language.

Қазіргі уақытта компьютерлік техника мен ақпараттық технологиялардың қарыштап дамуы ақпарат көлемінің өсуіне, ақпаратты сақтауға, өңдеу технологияларын жетілдіруге әкелуде. Техникалық жүйелерді алгоритм арқылы басқарудың жаңа мүмкіндіктері елімізде және шет елдерде жиырма жылдан астам уақыт бойы қолданылып келеді.Электронды есептеуіш машиналары пайда болғаннан бастап,осы құрылғылармен қарым-қатынас жасау қажеттігі өрби түсті. Осыған байланысты адамдардың өзара қатынас орнататын табиғи тілдерінен бөлек жасанды деп аталатын тілдер пайда бола бастады. Жасанды тілдер бір жағынан адам үшін түсінікті және ыңғайлы болуы керек әрі оны электронды есептеуіш құралғылары қабылдауы қажет. Бұндай талаптарды бір тіл ішінде іске асыру біраз қиындықтарға алып келді, сондықтан адамға түсінікті тілден құрылғы қабылдайтын тілге түрлендіретін трансляторлар пайда болды.Электронды есептеуіш машина машиналық кодтарында жазылған бағдарламаларды орындай алады. Транслятор бағдарламаны жадыға көшіру кезінде және оның орындалуы кезінде бағдарламаны берілген тілден машиналық тілге аударады.Трансляторды құру жұмысы айтарлықтай жауапты әрі қиын іс. Оның шешім қабылдау сапасына өзгертуді орындауға ресурстар байланысты болады. Бұнымен қатар кіріс тілінде жазылған мәтін мазмұнын дәлме-дәл жеткізу трансляторға қойылатын негізгі талап болып табылады. Ондай өзгертулерді іске қосу үшін тіл синтаксисі деп аталатын кіріс мәтіндерінің құрылу ережелерінің дәлме-дәл суреттелуі мен тіл семантикасы деп аталатын мәтін мазмұнын баяндайтын талдау ережелері кажет. Синтаксис пен семантиканы мазмұндау үшін түрлі құралдар қолданылады. Осылайша тіл синтаксисін формальді грамматика, ал семантиканы атрибуттық грамматика көмегімен мазмұндауға болады[1]. Формальді грамматика формальді ережелердің шектелген жиынынан тұрады. Ал осы формальді ережелерге сәйкес құрылған барлық сөздер дұрыс сөздер болады. Қазіргі таңда информатикада формальді тілдерді тудыратын бірнеше граматиканы беру әдістері жасалынған. Қарапайым әдістердің бірі Бэкустің нормаль формалары деп аталатын әдіс. Формальды тілдер граматикасын қарастыруды алгоритмнің қатаң сипаттамасын құру қажеттілігінен келетін болсақ, шын мәнісінде олардың қолданылу облысы әлдеқайда кеңірек. Формальды грамматика негізінде бағдарламалау тілдері және оларға трансляторлар құрылады. 147 Мәтінді компьютер көмегімен автоматты түрде өңдегенде, символдар ретімен ұсынылады. Мәтінді алдын-ала өңдеу – сөздер, сандар, тыныс белгілері және т.б базалық элементтерді бөлуден тұрады. Бұл үрдісі токендеу (tokenization) немесе токенмен бөлу деп аталады. Токен (token) – екі жағынан символ-бөлгіштермен шектелген, келесі анализге мәтіннің лингвистикалық мағынасын және бағасын көрсететін символдардың реттілігі. Қарапайым токендеу мәтіндегі екі жағы бос орынмен немесе пунктуациялық белгілермен шектелген жолдарды бөлуден тұрады. Токендеу процессі кезінде ТАӘ (ФИО), мекен-жай, телефон және т.б күрделі токендерді бөлу жүзеге асырылады. Токендер құрамды және бөлгіштерден тұрады (мысалы New York, әттең-ай). Бірақ New York жолын бір токен ретінде санау немесе санамау керектігі алдымен берілген тапсырмаға тәуелді. Токендеу кезінде токен немесе токен емес екендігін анықтау берілген тапсырмаға байланысты. Токендеуде қысқартулар, аббревиатуралар, құраушы көмекші сөздер және бірнеше мәтіннің басқа да элементтері бөлінеді. Токендеу нәтижесінде белгіленген токен ретінде: сөз, сан, дата, мекен-жай және т.б тиісті типтер бекітіледі. Токендеудің кең тараған ыңғайлы тәсілі регулярлы өрнектерді қолдануға негізделген. Регулярлы өрнек деп формальді тілде жазылған көптеген символдар тізбегін іздеуге арналған үлгілер. Регулярлы өрнек - мәтінді табиғи тілде өңдеу үшін кез келген қиындықтағы токендерді бөлуге мүмкіндік беретін таптырмас құрал. Регулярлы өрнек тәсілінің тамаша айырмашылығы болып анализ орындайтын бағдарламаның, анализ тәртібін сипаттайтын бағдарламадан бөлектігі болып табылады. Бұл тәсіл өндеу алгоритмін өзгертпей-ақ, тіркестерді жеңіл басқаруға және жаңасын қосуға мүмкіндік береді. Регулярлы өрнек ақырлы автоматтарда құрастырылады, сол себепті мәтінді өндеу процессі өте тез орындалады. Тұрақты тіркестермен жұмыс жасау үшін көптеген бағдарлама тілдерінде және өңдеу ортасында көптеген бағдарламалық кітапханалар бар. Сонымен қатар, жаңа тұрақты тіркестерді жасап шығаруды жеңілдететін редакторлар және тұрақты тіркестердің үлгі кітапханалары бар. Токендеуді жүргізетін компонентті токендеуші немесе токенайзер (tokenizer) деп атайды. Көптеген бағдарламалау тілдері үшін, токендерді бағдарламалық жүзеге асыруға, жеткілікті көлемде стандартты және ыңғайластырылған кітапханалар бар. Мұндай токендердегі тапсырмаларды өңдеу үшін қосымша баптаулар талап етіледі. Мәтінді алдынала өңдеуді айтарлықтай жеңілдететін көптеген лексикалық анализдің генераторлары бар. Сол құралдардың ішінен lex [Браун], flex [FLEX] құралдарын көрсетуге болады. Бұл құралдар – тұрақты тіркес түрінде үлгілерді қоюға және мәтінді өңдеу кезінде үлгілер сәйкес келгенде оларға С++ тілінде тікелей кодпен салыстыруға мүмкіндік береді[2]. Тілді формализациялаудың ең танымал тәсілдеріне: грамматикалар, регулярлы өрнектер, ақырлы автоматтар және Тьюринг машинасы жатады.Ақырлы автомат технологиясы басқа алгоритмдердің құрастырылуын жеңілдете түседі. Автоматтардың формальсыз сипатталынуы келесі түрде болады: автомат кіру лентасынан, күйдің нөмірін сақтайтын ақырлы жадысы бар басқару құрылғысынан тұрады, сонымен қатар қосымша және шығыстық лентасы болуы мүмкін (1-сурет). Бұл қабылданатын кіріс мәліметіне қарай бір күйден басқа күйге ауысып отыратын жүйе. Автомат – бұл A = (K, X,Y,δ ,γ ) түрдегі бестік, мұндағы K – күйлердің жиыны (күйлердің алфавиті), X – кіру алфавиті, Y – шығу алфавиті, δ − K : X → K бейнелеуді беретін өту функциясы, γ − K : X → Y бейнелеуді беретін шығу функциясы. Автоматтың функцияларын келесі түрдегі командалардың жиыны ретінде беруге болады: qx → py , мұндағы q және p ∈ K, x∈ X, y ∈Y қайсыбір тактісінде басқару құрылғысы q жағдайда болсын, ал кіру лентасынан x символы оқылсын делік. Егер командалардың жиынында qx → py командасы бар болса, онда i t тактісінде шығу летасына y символы жазылады, ал келесі t i +1 тактісінде басқару құрылғысы p жағдайына өтеді, яғни: y(t) = γ (q(t), x(t)),q(t −1) = δ (q(t), x(t))



Егер qx → py командасы болмаса, онда автомат бұғатталады және i t моментінде қабылданған символға ешқандай әсер білдірмейді, сонымен қатар уақыттың келесі моментіндегі символдарды да қабылдамайды. Инициалданбаған автоматтың анықтамасына сәйкес, жағдайдың алғашқы сәтінде автомат ерікті болуы мүмкін. Егер қандай да бір алғашқы жағдай алдын – ала бекітілген болса, онда мұндай автоматты инициалданған автомат деп атайды, яғни 0 q(0) = q . Контекстті еркін тілдер көпшілік заманауи программалау тілдірінің синтаксистік құрылымдарының негізінде жатыр. Біржақты детерминделген емес магазиндік (стектік) сыртқы жадысы бар автоматтар танушы болып табылады. Бұндай автоматтың жұмыс істеу алгоритімін іске асыру экспоненциалды күрделі болып табылады, алайда жетілдірулердің кейбір жолдарымен тізбектің ұзындығына байланысты керекті кіріс тізбегін талдау арқылы уақыттың полиноминальді байланысына қол жеткізуге болады. Сондай-ақ контекстті еркін тілдер арасында осы тәуелделік квадратты немесе сызықты болып табылатын көптеген тілдер класстары бар. Автоматтың жұмыс істеу жылдамдығы өте маңызды болғандықтан әртүрлі программалау тілдері үшін тексттерді анализдеу барысында контексті еркін түрлендіргіштер қолданылады. Лента саналымды ұяшықтарға бөлінген. Әрбір ұяшықта Σалфавитінің бір символы бар. Стек басы әрқашан стектің ең жоғары жағын cимволды оқиды.Онда негізгі екі амал бар[3]: Push:Стекке жаңа символ қосу. Pop: Бас символды оқужәне алып тастау. Лентаны әр жолы тек бір ғана ұяшықты оқиды әрі тек оңға қарай қозғалады. Ақырлы басқару саналымды күйлерден тұрады, күйлер жиынын Qдеп белгілейміз.Әрбір жылжыу барысында күйлер қозғалыс функциясына сәйкес төмендегі аралықта болады[4]: ( { }) : ( { }) ( { }) 2 ε δ ε ε Q x ГU Q x ∑U x ГU → Танушы деп біз белгілі бір жиынды анықтайтын кеңейтілген мағынадағы алгоритмді қарастырамыз (біздің жағдайда – тіл). Ол өзінің жұмысында келесі компоненттерді қолданады: кіріс лентасы, ақырлы жадысы бар басқару құрылғысы және қосымша жады. Әдетте басқарушы құрылғы тек кіріс лентасындағы ақпаратты оқи алады және лента бойымен алға жылжиды деп саналады. Танушы ақырлы сызықты тізім түріндегі ұяшықтарды немесе стек түріндегі жадының күйін де өзгерте алады. Егер де кіріс лентасына берілген тізбек бойымен танушы анықталған рет бойынша қадам жасап, ақырлы күйде аяқталса демек бұл тізбек тілге тиесілі деп қараймыз.Көптеген синтаксистік анализдің әдістері ағаш тәрізді болады. Қарапайым түрлердің бірі – контекстті- еркін грамматикалар болып табылады, ол мына ережелерге сүйене отырып S = NP +VP немесе VP =V + NP , мұнда сол жағы оң жаққа 149 контесттіксіз ауыса алады.Контекстті –еркін грамматикалар көптеген машиналық тілдерде көп қолданылады, және ол арқылы жоғары эффектілік анализдің әдістері құрылған. бұл әдістің жетіспеуі– граматикалықбұрыс фразаларға тыйым салуы жоқ болады, мысалы, бастауыш баяндауышпен қатысы жоқ.Бұл проблеманың шешімі бөлек екі грамматикалардың параллель жұмыс істеуі: бір жаңынан- жалғыз, басқа жағынан – көптік жалғауы. Бұдан басқа сөйлемдерге өзінің грамматикасы болуы керек және т.с.с.Семантикалық бұрыс сөйлем семантикалық жазуға айналуы мүмкін.Мұның бәрі үлкен ережелер жиынтығын береді, және контекстті– еркінграмматикалар NLP -ге жарамай қалады[5]. Жылжымалы нүктелі сан стринг жиым ретінде беріледі. Алгоритмнің әр қадамында автоматтың келесі конфигурациясы құрастырылатын команданы анықтау қажет. Сондықтан ағымдағы конфигурация бойынша δ функциясның аргументтерін құрастыру қажет. «Сан» конструкциясын талдауға арналған дүкен жадысы бар автоматтың грамматикасы төмендегі 1 кестеде бейнеленген.

Кесте 1

«Сан» конструкциясын талдауға арналған дүкен жадысы бар автоматтың грамматикасы



Зерттеу жұмысы барысында контекстті еркін грамматикасын қолдана отырып жылжымылы нүктелі сандардың грамматикасы сипатталып, күйді стекте сақтайтын дүкен жадысы бар автомат құрылды. Бағдарлама кірісі ретінде жылжымалы нүктелі сан қабылданады, ал шығыс ретінде сол санның кофигурациялар тізімі болады бастапқы конфигурациядан басталып соңғысына дейін.Егерде санның құрылысы дұрыс емес болған жағдайда шығыс ретінде сол санның конфигурациясы болады, бірақ бастапқыдан бастап қатеге дейінгі соңғы алынған санға дейін қарастырылады. Және де қай жерде қате кеткен конструкцияны көрсетеді.

 **Пайдаланылған әдебиеттер тізімі:**

1. Белоногов Г.Г., Кузнецов Б.А. «Языковые средства автоматизированных информационных систем».

2. Гладкий A.B. «Агоритмическая нераспознаваемость существенной неопределенности КС-языков».

3. Сафонов К.В. «О распознавании контекстно-свободных грамматик» // Вестник Томского государственного университета. Серия " Математика. Информатика. Кибернетика". 2003. - Прил. No 6. -С. 16-18.

4. Агафонов В.Н. Синтаксический анализ языков программирования: Уч. пособие. - Новосибирск: НГУ, 1981. - 91 с.

5. Ю.Волкова И. А. Руденко Т.В. Формальные грамматики и языки. Элементы теории трансляции. Москва: МГУ, 1999.

6. Братчиков И.Л. Синтаксис языков программирования.Москва: Издательство «Наука», 1974.