SECTION 1. Theoretical research in mathematics.

Baydarmanova Balausa Nurmakhanovna
master 2 courses of the specialty «Mathematics»,
Taraz State University named after M.Kh. Dulati,
Kazakhstan

SOME METHODS OF FINDING OF EQUIVALENT TRANSFORMATIONS IN THE CONTEXT FREE GRAMMARS.

The article considers the methods and algorithms factorization grammars when developing compilers of formal languages.

Keywords: Formal language, machines, algorithms, grammar, factorization.

НЕКОТОРЫЕ СПОСОБЫ НАХОЖДЕНИЯ ЭКВИВАЛЕНТНЫХ ПРЕОБРАЗОВАНИЙ В КОНТЕКСТЕ СВОБОДНЫХ ГРАММАТИК.

В статье рассматриваются методы и алгоритмы факторизации грамматик при разработке компиляторов формальных языков и автоматов.

Ключевые слова: формальные языки, автоматы, алгоритмы, грамматика, факторизация

МӘНМӘТІННЕН ЕРКІН ГРАММАТИКАЛАРДЫ ЭКВИВАЛЕНТТІ ТҮРЛЕНДІРУДІҢ КЕЙБІР ТӘСІЛДЕРІ.

Формальды тілдер және автоматтар теориясы информатиканың математикалық негізі болып есептелінеді. Жоғарғы деңгейдегі алгоритмдік тілдер үшін (мысалы, Паскаль, С++) компиляторларды құру формальды тілдер және автоматтар теориясының нәтижелеріне сүйене отырып жүзеге асырылады. Теорияда зерттелетін жалпы заңдылықтарды білу негізгі технологияларды қолдану арқылы практикада түрлі тілдер үшін түрлі компиляторларды іске асыру мүмкіндігіне жеткізеді [2].

Формальды тілдер теориясының дамуына Хомскидің (Chomsky N.) формальды грамматикалар бойынша жұмысы (1959ж.) және Бэкус (Backus J.W.) пен Наур (Naur P.) ұсынған белгілеулер жүйесі арқылы Алгол-60 тілінің синтаксисінің сипатталуы өз ықпалын тигізді [1-2].

Формальды грамматиканың Хомский анықтаған түрі (типі) тек ол грамматиканың ережелерінің құрылым үлгісімен ғана байланыста болады. Хомский грамматикаларды олардың түрлеріне сәйкес төрт класқа топтастыруды ұсынған, яғни Хомский классификациясын (иерархиясын) 0,1,2- және 3-түрдегі грамматикалар топтары құрады да, бірдей түрдегі (типтегі) грамматикалар тек бір класта жатады [2].

Кейінгі зерттеулерде Хомский иерархиясындағы әрбір грамматика үшін оның класына сәйкес талдаушы (автомат) табылатыны дәлелденді. Сонымен, алгоритмдерді сипаттаудың жаңа тәсілімен оған дейін белгілі болған Тьюринг (Turing A.M.) машиналарының (1936ж.) байланысы айқын болды. Талдаушылардың қалған түрлері Тьюринг машиналарының дербес жағдайларында алынады, мысалы, ақырлы автомат (1943ж.), магазиндік жады бар автомат (1961ж.).

Мақалада [1], [2], [3] жұмыстарының нәтижелеріне сүйене отырып, грамматикалар үшін жоғарыға талдайтын синтаксистік талдаушы магазиндік жады бар автомат түрінде іске асырылған.

Анықтама 1. Қайсыбір α сөзі үшін $A \overset{+}{\Rightarrow} A\alpha$ қорытып шығаруы орындалатындай A бейтерминалы табылса, грамматика *сол жақты рекурсиялы* деп аталады.

Айтылған қорытып шығару бір-ақ қадамды қажет етсе, ол *тікелей сол жақты рекурсия* деп аталады (яғни грамматикада $A \to A\alpha$ ережесі бар). Бірден артық қадам қажет болған жағдайды *тікелей емес рекурсия* дейді[2].

Ескерту 1. Төменге синтаксистік талдау әдістері сол жақты рекурсиялы грамматикалармен жұмыс істей алмайды, сондықтан мұндай грамматикаларда сол жақты рекурсияны жою мәселесі маңызды болып табылады.

Мысал 1. Сол жақты рекурсиялы $A \to A\alpha | \beta$ ережелерін сол жақты рекурсиясыз

$A \to \beta A'$,

$A' \to \alpha A' | \varepsilon$

ережелеріне алмастыруға болады[1].

Тікелей сол жақты рекурсияны жоюдың алгоритмі

Кіріс: $G = (N, \Sigma, P, S)$ – МЕ – грамматика;

Шығыс: $G' = (N', \Sigma, P', S)-$ тікелей сол жақты рекурсиясы болмағанМЕ– грамматика.

Қадам 1. Сол жақты рекурсиялы $A \in N$ бейтерминалы үшін барлық ережелерді топтау:

$A \to A\alpha_1 | A\alpha_2 | \ldots | A\alpha_m$,

$A \to \beta_1 | \beta_2 | \ldots | \beta_n$,

мұнда β_i сөздері A бейтерминалынан басталмайды,

ал α_i сөздерінің ешқайсысы ε бос сөзіне тең емес.

Қадам 2. Жаңа A' бейтерминалын енгізумен жоғарыдағы ережелерді жаңа ережелерге ауыстыру:

$A \to \beta_1 A' | \beta_2 A' | \ldots | \beta_n A'$,

$A' \to \alpha_1 A' | \alpha_2 A' | \ldots | \alpha_m A' | \varepsilon$

Қадам 3. Бейтерминалдар жиынын жаңа A' бейтерминалымен толықтыру.

Қадам 4. Грамматиканың кезектегі сол жақты рекурсиялы ережелері үшін 1,2,3 қадамдарын қайталау.

Қадам 5. Табылған бейтерминалдар жиыны мен ережелер жиынын сәйкес түрде N' және P' деп қабылдау[2].

Ескерту 2. Тікелей сол жақты рекурсиялары жойылған грамматикада ε–ережелер кездесіп қалуы мүмкін. Бізді тікелей сол жақты рекурсияларымен қатар ε–ережелері де болмаған грамматика қызықтыратын болса, алгоритмнің екінші қадамын былайша өзгертуге болады:

$A \to \beta_1 A' | \beta_2 A' | \ldots | \beta_n A'$,

$A' \to \alpha_1 A' | \alpha_2 A' | \ldots | \alpha_m A' | \alpha_1 | \alpha_2 | \ldots | \alpha_m$,

$A \to \beta_1 | \beta_2 | \ldots | \beta_n$.

Ескерту 3. Сипатталған алгоритм тікелей емес сол жақты рекурсияларды жоя алмайды.

Мысал 2. $S \Rightarrow Aa \Rightarrow Sda$ болған себепті,

$S \to Aa|b$,

$A \to Ac|Sd|\varepsilon$

грамматикасында S бейтерминалы сол жақты рекурсиялы, бірақ бұл рекурсия тікелей емес.

Ескерту 4. Тікелей емес сол жақты рекурсияны жоюдың алгоритмі [2,184-бет] жұмысында кеңінен көрсетілген.

Мысал3. 1-кестеде тікелей сол жақты рекурсияны жоятын алгоритмнің

$S \to Aa$,

$A \to Bb$

$B \to Cc|d$,

$C \to Ccbz|dbz$

МЕ-грамматикасына қолданылуы көрсетілген.

1-кесте

3. мысалындағы грамматиканы түрлендіру

Қадам	**Әрекет және нәтиже**
1	$C \to Ccbz$, $C \to dbz$
2	$C \to dbzC'$, $C \to dbz$, $C' \to cbzC'$, $C' \to cbz$
3	$N = \{S, A, B, C\} \cup \{C'\}$
4	Басқа тікелей сол рекурсия жоқ
5	$N' = \{S, A, B, C, C'\}$, $P' = \{ S \to Aa$, $A \to Bb$, $B \to Cc\|d$, $C \to dbzC'\|dbz$, $C' \to cbzC'\|cbz \}$

Мысал4. Арифметикалық өрнектер үшін

$B \to B + T|T$,

$T \to T * M|M$,

$M \to (B)|id$

грамматикасын қарастырамыз, мұнда $N = \{B$-өрнек, T-терм, M-көбейткіш$\}$, $\Sigma = \{+, (, *,), id$-идентификатор$\}$[3].

Тікелей сол жақты рекурсияларды жоя отырып алатынымыз:

$B \to TB'$,

$B' \to +TB'|\varepsilon$,

$T \to MT'$,

$T' \to *MT'|\varepsilon$,

$M \to (B)|id$.

Грамматика ережелерін сол жақтан факторизациялау

Анықтама 2. *Тізбектің төменге талдануы* деп талдау ағашының оның тамырынан бастап төбелеріне қарай құрылуын айтады. *Төменге талдауды* тізбекті сол жақты тудырудың әрекеті деп те түсінсе болады[3].

Анықтама 3. *Рекурсиялы төмен түсумен талдау* деп бірқатар рекурсиялы шаралар орындалған төменге талдау тәсілін айтады (шара әрбір терминалмен байланысқан).

Анықтама 4. Егер рекурсиялы төмен түсу әдісі тізбекті қайталап оқуды қажет етпесе, онда ол *болжағыш талдау* әдісі деп аталады.

Анықтама 5. Грамматиканы болжағыш әдіспен талдау үшін ыңғайлы түрге келтіруді *сол жақтан факторизациялау* деп атайды.

Ереже туындысын (продукциясын) таңдау варианты белгісіз болған жағдайда ережелерді факторизациялау арқылы дұрыс шешім қабылдауды кіріс ағынынан жеткілікті сандағы символ оқылғанға дейін уақытша кешіктіруге болады[2-3]. Мысалы,

Инструкция→**if** Өрнек **then** Инструкция **else** Инструкция,

Инструкция→**if** Өрнек **then** Инструкция,

альтернативалы ережелерімен жұмыс істеген жағдайда кіріс ағынынан **if** сөзінкездестіре сала, бұл ережелердің кайсысы таңдалатынын айту қиын. Жалпы, егер $A \to \alpha\beta_1|\alpha\beta_2$ болса және α бос емес тізбек тудырса, онда осы тізбек толығымен қарастырылып болғанға дейін ереже таңдамай-ақ күте тұруға болады:

$A \to \alpha A'$,

$A' \to \beta_1 | \beta_2$.

Ережелерді сол жақтан факторизациялаудың алгоритмі

Кіріс: $G = (N, \Sigma, P, S)$ – МЕ – грамматика;

Шығыс: $G' = (N', \Sigma, P', S)$ – ережелерінің оң жақтарында бірдей префикстері болмаған МЕ– грамматика[1].

Қадам 1. $A \in N$ бейтерминалының екі немесе одан да артық туындысына (продукциясына) ортақ болатын ең ұзын α префиксін табу.

Қадам 2. $\alpha = \varepsilon$ болса, онда 4-қадамға көшу,

әйтпесе $A \to \alpha\beta_1 | \alpha\beta_2 | \ldots | \alpha\beta_m | \gamma$ ережелерін

$A \to \alpha A' | \gamma$,

$A' \to \beta_1 | \beta_2 | \ldots | \beta_m$

ережелеріне алмастыру. Мұндағы γ тізбектері- префиксінде α болмаған тізбектер, A' -жаңа бейтерминал.

Қадам 3. Бейтерминалдар жиынын A' жаңа бейтерминалымен толықтыру.

Қадам 4. Екі альтернативаның ешқайсысы үшін ортақ префикс табылмай қалғанға дейін грамматиканың кезекті бейтерминалдары үшін 1,2,3 қадамдарын қайталау.

Қадам 5. Бейтерминалдар мен ережелердің табылған жиындарын сәйкес түрде N' және P' деп қабылдау[1].

Мысал 5. 2-кестеде сол жақтан факторизациялау алгоритмінің

$S \to aSb$,

$S \to aSc$,

$S \to d$

МЕ-грамматикасына қолданылуы көрсетілген.

2-кесте

5. мысалындағы грамматиканы түрлендіру

Қадам	Әрекет және нәтиже
1	$\alpha = aS \neq \varepsilon$
2	$S \to aSS' \mid d$, $S' \to b \mid c$
3	$N = \{S\} \cup \{S'\}$
4	Бейтерминалдардың ешқайсысы үшін ережелердің оң жақтарында ортақ префикс қалған жоқ
5	$N = \{S, S'\}$, $P' = \{ S \to aSS' \mid d , S' \to b \mid c \}$

ӘДЕБИЕТТЕР

1. Хопкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений. –М.: Вильямс, 2002.-528 с.

2. Хомский Н. Три модели для описания языка // Кибернетический сборник. –М.: ИЛ, 1961.-Вып.2.-С.237-266.

3. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. –М.: Мир, 1979.-536 с.