Министерство образования и науки российской федерации

Федеральное агентство по образованию РФ

Государственное образовательное учреждение   
высшего профессионального образования

"Ижевский государственный технический университет"

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к курсовой работе по дисциплине

«Теория языков программирования и методы трансляции»

Ижевск

Издательство ИжГТУ

2008

УДК 62-50 (076.5)

Составитель: д-р техн. наук проф. М. А. Сенилов

Рецензент: д-р техн. наук, проф. А. И. Мурынов

Методические указания к курсовой работе по дисциплине «Теория языков программирования и методы трансляции» / Сост. М. А. Сенилов. – Ижевск:

Изд-во ИжГТУ, 2008. – 28 с.

Методические указания освещают основные вопросы выполнения курсовой работы по дисциплине «Теория языков программирования и методы трансляции», включающей тематику теории автоматов, теории формальных грамматик и языков. В них дается индивидуальное задание, а также приводятся рекомендации и сведения, необходимые для выполнения курсовой работы.

Методические указания предназначены для студентов направления 230100 – «Информатика и вычислительная техника» и специальности 230105 – «Программное обеспечение вычислительной техники и автоматизированных систем».

© Сенилов М. А. составление, 2008

© Издательство Ижевского государственного

технического университета

ОГЛАВЛЕНИЕ

[1. ТЕМА, ЦЕЛЬ И СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ 4](#_Toc214701638)

[2. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 4](#_Toc214701639)

[3. ПЕРЕХОД ОТ ПРАВОЛИНЕЙНОЙ ГРАММАТИКИ К АВТОМАТНОЙ 6](#_Toc214701640)

[4. ПОСТРОЕНИЕ НЕДЕТЕРМИНИРОВАННОГО КОНЕЧНОГО АВТОМАТА 6](#_Toc214701641)

[5. СВЕДЕНИЕ НЕДЕТЕРМИНИРОВАННОГО КОНЕЧНОГО АВТОМАТА К ДЕТЕРМИНИРОВАННОМУ 10](#_Toc214701642)

[6. ПОСТРОЕНИЕ МИНИМАЛЬНОГО АВТОМАТА 14](#_Toc214701643)

[7. ИСПОЛЬЗОВАНИЕ СЕТЕЙ ПЕТРИ ПРИ ПЕРЕХОДЕ ОТ ГРАММАТИКИ К МИНИМАЛЬНОМУ АВТОМАТУ 17](#_Toc214701644)

[8. ПОРЯДОК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ 22](#_Toc214701645)

[9. РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ 23](#_Toc214701646)

[СПИСОК ЛИТЕРАТУРЫ 25](#_Toc214701647)

# 1. ТЕМА, ЦЕЛЬ И СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ

Тема курсовой работы: Синтез распознающего автомата.

Цель курсовой работы состоит в изучении способов задания языков грамматиками, распознающими автоматами и сетями Петри, построении модели конечного автомата, распознающего заданный язык, и его программной реализации.

Содержание и основные этапы работы:

1. построение праволинейной грамматики;
2. построение автоматной грамматики по праволинейной;
3. построение недетерминированного конечного автомата;
4. сведение недетерминированного конечного автомата к детерминированному;
5. построение минимального автомата;
6. выполнение этапов 2-5 с использованием аппарата сетей Петри;
7. программная реализация автомата.

# 2. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Исходными данными для курсовой работы являются две таблицы – табл. 1 и табл. 2 – и правила вывода R, приведенные ниже.

В табл. 1 первоначально записана лишь первая строка, содержащая перечисление 18 символов сi. Во вторую строку si записываются первые 18 символов фамилии, имени и отчества студента с обязательными пробелами между фамилией и именем, именем и отчеством. Затем в третью строку студент заносит для каждого из 18 символов строки символ из алфавита {x0, x1, x2, x3, x4, x5, x6 ,x7} в соответствии с табл. 2.

*Таблица 1*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ci | c1 | c2 | c3 | c4 | c5 | c6 | c7 | c8 | c9 | c10 | c11 | c12 | c13 | c14 | c15 | c16 | c17 | c18 |
| si | Б | Е | Р | Е | С | Н | Е | В | \_ | В | И | К | Т | О | Р | \_ | В | Я |
| xi | x5 | x6 | x0 | x6 | x4 | x7 | x6 | x2 | x5 | x2 | x3 | x7 | x5 | x4 | x0 | x5 | x2 | x7 |

Таблица 2 построена на основе подсчета появлений каждой буквы русского алфавита в фамилиях, именах и отчествах контингента студентов.

Затем буквы были сформированы в восемь групп с таким расчетом, чтобы появление каждого из символов x0 – x7 было равновероятным.

*Таблица* 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П |
| x1 | x5 | x2 | x4 | x6 | x6 | x4 | x3 | x3 | x0 | x7 | x0 | x3 | x7 | x4 | x5 |
| P | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ь | Ы | Э | Ю | Я | \_ |
| x0 | x4 | x5 | x7 | x2 | x5 | x1 | x2 | x2 | x0 | x6 | x1 | x1 | x3 | x7 | x5 |

Наконец, задана формальная грамматика: G=<Vt, Vn, S, R>, где Vt={c1, c2, c3, … , c18} – терминальный словарь; Vn={S, A, B, C, D, E, F} – нетерминальный словарь; S∈Vn – начальный символ грамматики; R – множество правил вывода, которые имеют следующий вид:

S→c1 c2 c3 A; S→c1 c4 c5 B; S→c6 C; S→c7 F;

A→c8 D; A→c9; B→c8 E; B→c9; C→c8E; C→c9;

D→c10 S; D→c11; E→c10 S; E→c11;

F→c12 c13 c14 c15; F→c16 c13 c14 c15; F→c17 c18 c15.

Эта грамматика, являющаяся праволинейной, приводится к виду

G'=<V't, V'n, S, R'>, где V't={x0, …, x7} – новый терминальный словарь;

R' – множество правил вывода, получаемых из заданных заменой символов из алфавита Vt символами из алфавита V't в соответствии с таблицей 1. В данном примере они имеют вид:

S→x5 x6 x0 A | x5 x6 x4 B | x7 C | x6 F;

A→x2 D | x5;

B→x2 E | x5;

C→x2 E | x5;

D→x2 S | x3;

E→x2 S | x3;

F→x7 x5 x4 x0 | x5 x5 x4 x0 | x2 x7 x0.

Здесь “|” – металингвистический символ (связка), читаемый как "ИЛИ".

Примеры цепочек, принадлежащих языку L(G'), порождаемому грамматикой G': x5 x6 x0 x5, x7 x2 x3, x6 x7 x5 x4 x0. Напротив, цепочки x2 x7 x0 x2 x7, x6 x5 x2 x5 x6 не принадлежат L(G'), так как они не выводимы в грамматике G'.

Грамматика G', порождаемая из заданной грамматики G, является индивидуальным заданием студента.

Примечание. Мощность |V't| словаря V't (число символов в нем) в рассмотренном случае равна 7: словарь не содержит символа x1.

# 3. ПЕРЕХОД ОТ ПРАВОЛИНЕЙНОЙ ГРАММАТИКИ К АВТОМАТНОЙ

Этот этап выполняется путем расширения нетерминального словаря способом, вытекающим из возможности преобразования праволинейной грамматики в модифицированную автоматную G''=< V't, V''n, S, R''>. Для рассматриваемого примера, который будет сквозным в данных указаниях, получим множество R'' правил вывода:

S→x5 S1; S1→x6 S2; S2→x0 A;

S→x5 S3; S3→x6 S4; S4→x4 B;

S→x7 C;

S→x6 F;

A→x2 D; A→x5 A1; A1→ε;

B→x2 E; B→x5 B1; B1→ε;

C→x2 E; C→x5 C1; C1→ε;

D→x2 S; D→x3D1; D1→ε;

E→x2 S; E→x3 E1; E1→ε;

F→x7 F1; F1→x5 F2; F2→x4 F3; F3→x0 F4; F4→ε;

F→x5 F5; F5→x5 F6; F5→x4 F7; F7→x0 F8; F8→ε;

F→x2 F9; F9→x7 F10; F10→x0 F11; F11→ε.

Таким образом, нетерминальный словарь теперь имеет вид V''n = {S, S1, S2, S3, S4, A, A1, B, B1, C, C1, D, D1, E, E1, F, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11} и его мощность |V''n| равна 27.

# 4. ПОСТРОЕНИЕ НЕДЕТЕРМИНИРОВАННОГО КОНЕЧНОГО АВТОМАТА

Построим на основе грамматики G'' конечный автомат. При этом нетерминальным символам грамматики V''n поставим в соответствие состояния автомата (оставим для них те же обозначения). Начальному нетерминалу S поставим в соответствие начальное состояние автомата S. Правилам вывода поставим в соответствие переходы автомата.

В результате получим таблицу 3 – таблицу переходов недетерминированного конечного автомата, соответствующего рассматриваемому примеру.

*Таблица* 3

Таблица переходов недетерминированного частичного автомата

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | x0 | x2 | x3 | x4 | x5 | x6 | x7 |  |
| S |  |  |  |  | S1,S3 | F | C | 0 |
| S1 |  |  |  |  |  | S2 |  | 0 |
| S2 | A |  |  |  |  |  |  | 0 |
| S3 |  |  |  |  |  | S4 |  | 0 |
| S4 |  |  |  | B |  |  |  | 0 |
| A |  | D |  |  | A1 |  |  | 0 |
| A1 |  |  |  |  |  |  |  | 1 |
| B |  | E |  |  | B1 |  |  | 0 |
| B1 |  |  |  |  |  |  |  | 1 |
| C |  | E |  |  | C1 |  |  | 0 |
| C1 |  |  |  |  |  |  |  | 1 |
| D |  | S | D1 |  |  |  |  | 0 |
| D1 |  |  |  |  |  |  |  | 1 |
| E |  | S | E1 |  |  |  |  | 0 |
| E1 |  |  |  |  |  |  |  | 1 |
| F |  | F9 |  |  | F5 |  | F1 | 0 |
| F1 |  |  |  |  | F2 |  |  | 0 |
| F2 |  |  |  | F3 |  |  |  | 0 |
| F3 | F4 |  |  |  |  |  |  | 0 |
| F4 |  |  |  |  |  |  |  | 1 |
| F5 |  |  |  |  | F6 |  |  | 0 |
| F6 |  |  |  | F7 |  |  |  | 0 |
| F7 | F8 |  |  |  |  |  |  | 0 |
| F8 |  |  |  |  |  |  |  | 1 |
| F9 |  |  |  |  |  |  | F10 | 0 |
| F10 | F11 |  |  |  |  |  |  | 0 |
| F11 |  |  |  |  |  |  |  | 1 |

Полученный автомат является неполностью определенным (частичным), поскольку в таблице переходов (табл. 3) есть незаполненные клетки, и, кроме того, недетерминированным, поскольку в той же таблице есть клетки, содержащие пары состояний.

Чтобы получить полностью определенный автомат, следует ввести дополнительное состояние Er (ошибка). Для этого нужно дополнить автомат соответствующей строкой с состоянием Er и во все пустые клетки таблицы переходов вписать это состояние Er (см. табл. 4).

*Таблица* *4*

Таблица переходов недетерминированного полностью определенного автомата

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | x0 | x2 | x3 | x4 | x5 | x6 | x7 |  |
| S | Er | Er | Er | Er | S1,S3 | F | C | 0 |
| S1 | Er | Er | Er | Er | Er | S2 | Er | 0 |
| S2 | A | Er | Er | Er | Er | Er | Er | 0 |
| S3 | Er | Er | Er | Er | Er | S4 | Er | 0 |
| S4 | Er | Er | Er | B | Er | Er | Er | 0 |
| A | Er | D | Er | Er | A1 | Er | Er | 0 |
| A1 | Er | Er | Er | Er | Er | Er | Er | 1 |
| B | Er | E | Er | Er | B1 | Er | Er | 0 |
| B1 | Er | Er | Er | Er | Er | Er | Er | 1 |
| C | Er | E | Er | Er | C1 | Er | Er | 0 |
| C1 | Er | Er | Er | Er | Er | Er | Er | 1 |
| D | Er | S | D1 | Er | Er | Er | Er | 0 |
| D1 | Er | Er | Er | Er | Er | Er | Er | 1 |
| E | Er | S | E1 | Er | Er | Er | Er | 0 |
| E1 | Er | Er | Er | Er | Er | Er | Er | 1 |
| F | Er | F9 | Er | Er | F5 | Er | F1 | 0 |
| F1 | Er | Er | Er | Er | F2 | Er | Er | 0 |
| F2 | Er | Er | Er | F3 | Er | Er | Er | 0 |
| F3 | F4 | Er | Er | Er | Er | Er | Er | 0 |
| F4 | Er | Er | Er | Er | Er | Er | Er | 1 |
| F5 | Er | Er | Er | Er | F6 | Er | Er | 0 |
| F6 | Er | Er | Er | F7 | Er | Er | Er | 0 |
| F7 | F8 | Er | Er | Er | Er | Er | Er | 0 |
| F8 | Er | Er | Er | Er | Er | Er | Er | 1 |
| F9 | Er | Er | Er | Er | Er | Er | F10 | 0 |
| F10 | F11 | Er | Er | Er | Er | Er | Er | 0 |
| F11 | Er | Er | Er | Er | Er | Er | Er | 1 |
| Er | Er | Er | Er | Er | Er | Er | Er | 0 |

Граф переходов автомата, построенный по таблице 4, показан на рис. 1.

A1

S2

F5

D

F

S

B

C

S1

F2

F9

F4

E

A

F1

F10

F3

F11

F6

F8

F7

x0

x7

x5

x5

x6

x7

x6

x6

x0

x4

x5

x2

x2

x5

x5

x2

x2

x3

x3

x2

x5

x7

x5

x4

x0

x5

x4

x0

x2

Er

D1

S3

S4

B1

C1

E1

x0∨x2∨x3∨x4∨x5∨x6∨x7

*Рис. 1. Граф переходов исходного (недетерминированного) автомата*

Примечание. Для того чтобы не затемнять картину, на рисунке опущены дуги, исходящие из недопускающих состояний и ведущие в состояние Er (ошибка). Так, например, опущена дуга, ведущая из состояния S в состояние Er, которая должна быть помечена дизъюнкцией входных символов x0∨x2∨x3∨x4.

Все дуги, ведущие из допускающих состояний в состояние Er, должны быть помечены дизъюнкцией x0∨x2∨x3∨x4∨x5∨x6∨x7.

Нетрудно убедиться, что построенный автомат допускает все те и только те цепочки символов, которые принадлежат языку L(G'), порождаемому грамматикой G'.

# 5. СВЕДЕНИЕ НЕДЕТЕРМИНИРОВАННОГО КОНЕЧНОГО АВТОМАТА К ДЕТЕРМИНИРОВАННОМУ

Процедура построения детерминированного автомата Ад, эквивалентного недетерминированному автомату Ан, задается следующими шагами:

* + - 1. Пометить первую строку таблицы переходов для Ад множеством начальных состояний автомата Ан. Применить к этому множеству шаг 2.
      2. По данному множеству состояний В, помечающему строку таблицы переходов автомата Ад, для которой переходы еще не вычислены, вычислить те состояния автомата Ан, которые могут быть достигнуты из В с помощью каждого входного символа х, и поместить полученные множества состояний в соответствующие ячейки таблицы для автомата Ад. Символически это выражается так: если δ – функция недетерминированных переходов, то функция детерминированных перехода δ' задается формулой δ'(B, x) = {S | S ∈ δ (T, x), ∀ Т ∈ В}
      3. Для каждого нового множества, порожденного переходами на шаге 2, посмотреть, имеется ли уже в Ад строка, помеченная этим множеством. Если нет, то создать новую строку и пометить ее этим множеством. Если множество уже использовалось как метка, никаких действий не требуется.
      4. Если в таблице автомата Ад есть строка, для которой еще не вычислены переходы, вернуться назад и применить к этой строке шаг 2. Если все переходы вычислены, перейти к шагу 5.
      5. Пометить строку как допускающее состояние автомата Ад тогда и только тогда, когда она содержит допускающее состояние недетерминированного автомата. В противном случае пометить как отвергающее состояние.

Описанная процедура гарантирует, что детерминированный автомат не содержит недостижимых состояний.

В результате применения этого алгоритма от недетерминированного автомата (табл. 4, рис. 1) можно перейти к эквивалентному детерминированному автомату, таблица переходов которого приведена в таблице 5.

*Таблица* *5*

Таблица переходов детерминированного автомата.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | x0 | x2 | x3 | x4 | x5 | x6 | x7 |  |
| {S} | {Er} | {Er} | {Er} | {Er} | {S1,S3} | {F} | {C} | 0 |
| {S1,S3} | {Er} | {Er} | {Er} | {Er} | {Er} | {S2,S4} | {Er} | 0 |
| {S2,S4} | {A} | {Er} | {Er} | {B} | {Er} | {Er} | {Er} | 0 |
| {A} | {Er} | {D} | {Er} | {Er} | {A1} | {Er} | {Er} | 0 |
| {A1} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | 1 |
| {B} | {Er} | {E} | {Er} | {Er} | {B1} | {Er} | {Er} | 0 |
| {B1} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | 1 |
| {C} | {Er} | {E} | {Er} | {Er} | {C1} | {Er} | {Er} | 0 |
| {C1} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | 1 |
| {D} | {Er} | {S} | {D1} | {Er} | {Er} | {Er} | {Er} | 0 |
| {D1} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | 1 |
| {E} | {Er} | {S} | {E1} | {Er} | {Er} | {Er} | {Er} | 0 |
| {E1} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | 1 |
| {F} | {Er} | {F9} | {Er} | {Er} | {F5} | {Er} | {F1} | 0 |
| {F1} | {Er} | {Er} | {Er} | {Er} | {F2} | {Er} | {Er} | 0 |
| {F2} | {Er} | {Er} | {Er} | {F3} | {Er} | {Er} | {Er} | 0 |
| {F3} | {F4} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | 0 |
| {F4} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | 1 |
| {F5} | {Er} | {Er} | {Er} | {Er} | {F6} | {Er} | {Er} | 0 |
| {F6} | {Er} | {Er} | {Er} | {F7} | {Er} | {Er} | {Er} | 0 |
| {F7} | {F8} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | 0 |
| {F8} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | 1 |
| {F9} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | {F10} | 0 |
| {F10} | {F11} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | 0 |
| {F11} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | 1 |
| {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | {Er} | 0 |

Перейдем к более простым обозначениям состояний автомата:

{S}→Y; {S1,S3}→Y1; {S2,S4}→Y2;

{A}→Y3; {A1}→Y4; {B}→Y5; {B1}→Y6;

{C}→Y7; {C1}→Y8; {D}→Y9; {D1}→Y10;

{E}→Y11; {E1}→Y12; {F}→Y13; {F1}→Y14;

{F2}→Y15; {F3}→Y16; {F4}→Y17; {F5}→Y18;

{F6}→Y19; {F7}→Y20; {F8}→Y21; {F9}→Y22;

{F10}→Y23; {F11}→Y24; {Er}→Er.

В новых обозначениях таблица переходов автомата приведена в табл. 6.

*Таблица 6*

Таблица переходов детерминированного автомата

(новые обозначения состояний)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | x0 | x2 | x3 | x4 | x5 | x6 | x7 |  |
| Y | Er | Er | Er | Er | Y1 | Y13 | Y7 | 0 |
| Y1 | Er | Er | Er | Er | Er | Y2 | Er | 0 |
| Y2 | Y3 | Er | Er | Y5 | Er | Er | Er | 0 |
| Y3 | Er | Y9 | Er | Er | Y4 | Er | Er | 0 |
| Y4 | Er | Er | Er | Er | Er | Er | Er | 1 |
| Y5 | Er | Y13 | Er | Er | Y6 | Er | Er | 0 |
| Y6 | Er | Er | Er | Er | Er | Er | Er | 1 |
| Y7 | Er | Y11 | Er | Er | Y8 | Er | Er | 0 |
| Y8 | Er | Er | Er | Er | Er | Er | Er | 1 |
| Y9 | Er | Y | Y10 | Er | Er | Er | Er | 0 |
| Y10 | Er | Er | Er | Er | Er | Er | Er | 1 |
| Y11 | Er | Y | Y12 | Er | Er | Er | Er | 0 |
| Y12 | Er | Er | Er | Er | Er | Er | Er | 1 |
| Y13 | Er | Y22 | Er | Er | Y18 | Er | Y14 | 0 |
| Y14 | Er | Er | Er | Er | Y15 | Er | Er | 0 |
| Y15 | Er | Er | Er | Y16 | Er | Er | Er | 0 |
| Y16 | Y17 | Er | Er | Er | Er | Er | Er | 0 |
| Y17 | Er | Er | Er | Er | Er | Er | Er | 1 |
| Y18 | Er | Er | Er | Er | Y19 | Er | Er | 0 |
| Y19 | Er | Er | Er | Y20 | Er | Er | Er | 0 |
| Y20 | Y21 | Er | Er | Er | Er | Er | Er | 0 |
| Y21 | Er | Er | Er | Er | Er | Er | Er | 1 |
| Y22 | Er | Er | Er | Er | Er | Er | Y23 | 0 |
| Y23 | Y24 | Er | Er | Er | Er | Er | Er | 0 |
| Y24 | Er | Er | Er | Er | Er | Er | Er | 1 |
| Er | Er | Er | Er | Er | Er | Er | Er | 0 |

Граф переходов автомата, построенный по таблице 6, показан на рис. 2.

Y1

Y2

Y10

Y7

Y3

Y9

Y5

Y6

Y11

Y12

Y8

Y22

Y23

Y14

Y15

Y16

Y17

Y18

Y19

Y20

Y21

x0

x7

x5

x6

x7

x6

x0

x4

x5

x2

x2

x5

x5

x2

x2

x3

x3

x2

x5

x7

x5

x4

x0

x5

x4

x0

Y

Y13

Y24

x25

Y4

*Рис. 2 Граф переходов детерминированного автомата эквивалентного исходному*

Примечание. Чтобы не загромождать рисунок, не изображено состояние Er а также дуги, ведущие в него из других состояний.

Построенный автомат не является минимальным, поэтому необходимо выполнить этап минимизации автомата, на котором строится минимальный (иначе – приведенный) автомат.

# 6. ПОСТРОЕНИЕ МИНИМАЛЬНОГО АВТОМАТА

Произвольный конечный автомат можно превратить в эквивалентный ему минимальный, выбрасывая недостижимые и объединяя эквивалентные состояния. Разбиение состояний на классы эквивалентности можно осуществить с помощью метода разбиения.

Метод разбиения заключается в разбиении множества состояний на непересекающиеся подмножества или блоки, такие, что неэквивалентные состояния попадают в разные блоки.

Начальное разбиение P0 заключается в разбиении всего множества состояний на подмножества допустимых и недопустимых состояний:

P0=({Y, Y1, Y2, Y3, Y5, Y7, Y9, Y11, Y13, Y14, Y15, Y16, Y18, Y19, Y20, Y22, Y23, Er}, {Y4, Y6, Y8, Y10, Y12, Y17, Y21, Y24})

Далее произведем разбиение блока 1 по входу X5 и получим разбиение

P1=({Y, Y1, Y2, Y9, Y11, Y13, Y14, Y15, Y16, Y18, Y19, Y20, Y22, Y23, Er}, {Y3, Y5, Y7}, {Y4, Y6, Y8, Y10, Y12, Y17, Y21, Y24})

Произведем разбиение блока 1 по входу X0, получим разбиение

P2=({Y, Y1, Y9, Y11, Y13, Y14, Y15, Y18, Y19, Y22, Er}, {Y16, Y20, Y23}, {Y2}, {Y3, Y5, Y7}, {Y4, Y6, Y8, Y10, Y12, Y17, Y21, Y24})

Произведем разбиение блока 1 по входу X7, получим разбиение

P3=({Y1, Y9, Y11, Y13, Y14, Y15, Y18, Y19, Er}, {Y}, {Y22}, {Y16, Y20, Y23}, {Y2}, {Y3, Y5, Y7}, {Y4, Y6, Y8, Y10, Y12, Y17, Y21, Y24})

Произведем разбиение блока 1 по входу X6, получим разбиение

P4=({Y9, Y11, Y13, Y14, Y15, Y18, Y19, Er}, {Y1}, {Y}, {Y22}, {Y16, Y20, Y23}, {Y2}, {Y3, Y5, Y7}, {Y4, Y6, Y8, Y10, Y12, Y17, Y21, Y24})

Произведем разбиение блока 1 по входу X3, получим разбиение

P5=({Y13, Y14, Y15, Y18, Y19, Er}, {Y9, Y11}, {Y1}, {Y}, {Y22}, {Y16, Y20, Y23}, {Y2}, {Y3, Y5, Y7}, {Y4, Y6, Y8, Y10, Y12, Y17, Y21, Y24})

Произведем разбиение блока 1 по входу X4, получим разбиение

P6=({Y13, Y14, Y18, Er}, {Y15, Y19}, {Y9, Y11}, {Y1}, {Y}, {Y22}, {Y16, Y20, Y23}, {Y2}, {Y3, Y5, Y7}, {Y4, Y6, Y8, Y10, Y12, Y17, Y21, Y24})

Произведем разбиение блока 1 по входу X2, получим разбиение

P7=({Y14, Y18, Er}, {Y13}, {Y15, Y19}, {Y9, Y11}, {Y1}, {Y}, {Y22}, {Y16, Y20, Y23}, {Y2}, {Y3, Y5, Y7}, {Y4, Y6, Y8, Y10, Y12, Y17, Y21, Y24})

Произведем разбиение блока 1 по входу X5, получим разбиение

P8=({Er}, {Y14, Y18}, {Y13}, {Y15, Y19}, {Y9, Y11}, {Y1}, {Y}, {Y22}, {Y16, Y20, Y23}, {Y2}, {Y3, Y5, Y7}, {Y4, Y6, Y8, Y10, Y12, Y17, Y21, Y24})

Множество Р8 не допускает дальнейшего разбиения ни по одному входу, оно содержит подмножества (блоки) эквивалентных состояний, которые и являются состояниями минимального автомата.

Введем обозначения для этих подмножеств – состояний минимального автомата (табл. 7).

*Таблица 7*

Состояния минимального автомата

|  |  |
| --- | --- |
| Блок эквивалентных состояний | Состояние минимального автомата |
| {Y} | 1 |
| {Y1} | 2 |
| {Y2} | 3 |
| {Y3, Y5, Y7} | 4 |
| {Y9, Y11} | 5 |
| {Y13} | 6 |
| {Y14, Y18} | 7 |
| {Y15, Y19} | 8 |
| {Y16, Y20, Y23} | 9 |
| {Y22} | 10 |
| {Y4, Y6, Y8, Y10, Y12, Y17, Y21, Y24} | 11 |
| {Er} | Er |

Таблица переходов минимального автомата показана в табл. 8.

*Таблица8*

Таблица переходов минимального автомата

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | x0 | x2 | x3 | x4 | x5 | x6 | x7 |  |
| 1 | Er | Er | Er | Er | 2 | 6 | 4 | 0 |
| 2 | Er | Er | Er | Er | Er | 3 | Er | 0 |
| 3 | 4 | Er | Er | 4 | Er | Er | Er | 0 |
| 4 | Er | 5 | Er | Er | 11 | Er | Er | 0 |
| 5 | Er | 1 | 11 | Er | Er | Er | Er | 0 |
| 6 | Er | 10 | Er | Er | 7 | Er | 7 | 0 |
| 7 | Er | Er | Er | Er | 8 | Er | Er | 0 |
| 8 | Er | Er | Er | 9 | Er | Er | Er | 0 |
| 9 | 11 | Er | Er | Er | Er | Er | Er | 0 |
| 10 | Er | Er | Er | Er | Er | Er | 9 | 0 |
| 11 | Er | Er | Er | Er | Er | Er | Er | 1 |
| Er | Er | Er | Er | Er | Er | Er | Er | 0 |

Граф переходов минимального автомата, построенный по табл. 8, показан на рис. 3.

8

1

2

9

5

3

7

4

6

10

11

x5

x6

x0

x7

x2

x2

x5

x7

x2

x3

x0

x51

x5

x4

x6

x4

x7

Рис*. 3. Граф переходов минимального распознающего автомата*

Примечание. Чтобы не загромождать рисунок, не изображено состояние Er, а также дуги, ведущие в него из других состояний.

# 7. ИСПОЛЬЗОВАНИЕ СЕТЕЙ ПЕТРИ ПРИ ПЕРЕХОДЕ ОТ ГРАММАТИКИ К МИНИМАЛЬНОМУ АВТОМАТУ

Построим для грамматики G' сеть Петри. Это можно сделать, поставив в соответствие нетерминальным символам позиции (кружки) сети, а терминалам – переходы (планки) сети. Будем помечать позиции и переходы соответствующими нетерминалами и терминалами. Поскольку сеть Петри является двудольным графом, соединение дугами позиций разрешается только через переходы, а переходов - через позиции. Если в правой части некоторого правила вывода из R' имеет место конкатенация терминалов, то в сети Петри между переходами, помеченными терминалами, должны появляться дополнительные позиции, которые можно помечать символами левой части правила вывода с верхними индексами 1, 2, ... . Так, фрагмент сети Петри, соответствующий первому правилу вывода (S → x5 x6 x0 A) , будет иметь вид, показанный на рис. 4. При построении остальных фрагментов, соответствующих последующим правилам вывода, следует использовать ранее обозначенные позиции (но не переходы!). Таким образом, позиции могут иметь несколько входящих и исходящих дуг, но переходы – в точности по одной входящей и не более чем одной исходящей дуге (исходящая дуга может отсутствовать, если в правой части правила вывода отсутствует замыкающий нетерминал).

S

x5

S1

x6

S2

x0

A

*Рис. 4. Фрагмент интерпретированной сети Петри, соответствующий правилу вывода S → x5 x6 x0 A*

Маркером отмечается позиция, соответствующая начальному символу грамматики G' (в данном случае это позиция S).

В остальном правила построения сети Петри ясны из рассматриваемого сквозного примера - сравнить правила R' с рис. 5.

x4

x7

S

S3

S1

S2

A

D

S4

B

E

C

F1

F2

F3

F6

F5

F4

F

F7

F8

x2

x3

x2

x2

x5

x5

x6

x6

x0

x4

x5

x3

x5

x5

x2

x2

x7

x6

x5

x7

x2

x5

x5

x4

x0

x0

x0

*Рис. 5. Сеть Петри, соответствующая исходной грамматике*

Понятно, что сеть на рис. 5 представляет собой ни что иное, как автоматную сеть Петри (подкласс сетей Петри) и что позиции можно трактовать как состояния конечного автомата, а переходы - как входные символы.

Для полноты соответствия построенной сети Петри распознающему автомату Мура введем не показанную на рис. 5 заключительную позицию Z, в которую направим дуги из всех переходов, ранее не имевших исходящих дуг (см. рис. 6).

S

x4

Z

x7

S3

S1

S2

A

D

S4

B

E

C

F1

F2

F3

F6

F5

F4

F

F7

F8

x2

x3

x2

x2

x5

x5

x6

x6

x0

x4

x5

x3

x5

x5

x2

x2

x7

x6

x5

x7

x2

x5

x5

x4

x0

x0

x0

*Рис 6. Сеть Петри, дополненная заключительной позицией Z.*

Теперь следует заметить, что в сети на рис. 6 имеются три идентичных фрагмента, содержащие позиции A, B и C, каждой из которых инцидентны по два выходных перехода x2 и x5, и еще два идентичных фрагмента с позициями D и E, каждой из которых инцидентен выходной переход x3.

Функционирование сети не изменится, если склеить идентичные фрагменты. Позиции S1 и S3; F1 и F4; F2 и F5; F3, F6 и F8 также можно склеить (см. рис. 7).

Z

x7

S

S1,S3

S2

A,B,C

D,E

S4

F1,F4

F2,F5

F3,F6,F8

F

F7

x2

x3

x2

x5

x6

x6

x0

x5

x5

x7

x6

x5

x7

x2

x5

x4

x0

*Рис. 7. Преобразованная недетерминированная сеть Петри*

Этот этап соответствует минимизации числа состояний автомата, но он выполнен для автомата, сохраняющего недетерминированность. Источником недетерминированности, очевидно, могут являться лишь позиции свободного выбора, исходящие дуги которых являются входящими дугами переходов, помеченных одинаковыми терминалами. В данном случае к таким позициям следует отнести лишь позицию (S1, S3). Способ устранения недетерминированности виден из рисунка 9.

Pi

Pi

tj

tj

tj

tq

ts

tq

tr

ts

*Рис. 8. Типичный пример устранения недетерминированности*

Недетерминированность устраняется склеиванием двух позиций Pl и Pk в одну (Pl, Pk). При этом позиции (Pl, Pk) инцидентны все исходящие дуги, ведущие в переходы tr, tq, ts, являющиеся исходящими дугами позиций Pl и Pk.

Применим этот способ к сети на рис. 7, получим сеть, показанную на рисунке 9.

Z

x7

S

S1,S3

S2,S4

A,B,C

D,E

F1,F4

F2,F5

F3,F6,F8

F

F7

x2

x3

x2

x5

x6

x0

x4

x5

x7

x6

x5

x7

x2

x5

x4

x0

(1)

(11)

(5)

(4)

(3)

(2)

(6)

(7)

(8)

(9)

(10)

*Рис. 9. Минимизированная детерминированная сеть Петри*

В данном случае можно установить взаимно однозначное соответствие между сетью рис. 9 и графом переходов минимального автомата – рис. 3. (В скобках на рис. 9 указаны соответствующие состояния минимального автомата).

Указание. Выполнение обоих способов построения минимальнного автомата обязательно, так как это обеспечивает контроль правильности выполнения этапов курсовой работы.

От рис. 9 можно перейти к графу переходов автомата, заменив переходы сети Петри дугами автомата, переименовав позиции сети соответствующими им состояниями автомата и взвесив дуги наименованиями этих переходов.

# 8. ПОРЯДОК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Этап 1. Получение задания. Построение праволинейной грамматики. Переход от праволинейной грамматики к автоматной.

Результаты: праволинейная грамматика, ее граф, автоматная грамматика.

Этап 2. Построение недетерминированного распознающего автомата.

Результаты: таблица переходов и граф переходов недетерминированного автомата.

Этап 3. Переход от недетерминированного автомата к полностью определенному детерминированному автомату.

Результаты: таблица переходов и граф переходов детерминированного автомата.

Этап 4. Минимизация автомата.

Результаты: таблица переходов и граф переходов минимального автомата. Этап 5. Выполнение предыдущих этапов с использованием аппарата сетей Петри.

Результаты: исходная сеть Петри, сеть свободного выбора, детерминированная сеть Петри, сравнение полученной автоматной сети с графом минимального автомата.

Этап 6. Написание программы, реализующей распознающий автомат.

Результаты: текст программы и документация к ней.

Этап 7. Тестирование программы.

Результаты:

1. Примеры тестов (правильно построенные цепочки, порождаемые исходной грамматикой, и неправильные цепочки);
2. Результат проверки допустимости тестовых цепочек автоматом (построение соответствующих последовательностей переходов автомата);
3. Сравнение результатов работы программы с результатами работы автомата на тестовом множестве цепочек.

# 9. РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ

Ниже приведено типовое содержание пояснительной записки.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1. Постановка задачи
2. Индивидуальное задание. Построение праволинейной грамматики
3. Построение автоматной грамматики по праволинейной
4. Построение недетерминированного конечного автомата
5. Сведение недетерминированного конечного автомата к детерминированному
6. Построение минимального автомата
7. Построение сети Петри, моделирующей работу распознающего автомата
8. Описание программы, реализующей распознающий автомат
   1. Вводная часть
   2. Функциональное назначение
   3. Описание информации
   4. Описание логики
9. Описание контрольного примера
   1. Назначение
   2. Исходные данные
   3. Результаты расчета
   4. Результаты испытания программы

Заключение

Список литературы

Приложения

Приложение 1. Текст программы

Приложение 2. Результаты расчета на ЭВМ контрольного примера

Во «Введении» показывается актуальность темы и формулируется цель курсовой работы.

Здесь можно показать значение формальных грамматик, конечных автоматов(распознавателей), сетей Петри для решения задач построения лингвистических процессоров.

Целью курсовой работы является изучение способов задания языков грамматиками, распознающими автоматами и сетями Петри, синтез и программная реализация конечного автомата, распознающего заданный язык.

В разделе 1 необходимо дать четкую формулировку задачи, описать подход к ее решению, отразить основные этапы разработки(синтеза) распознающего автомата, описать входную и выходную информацию.

В разделе 2 описывается формирование индивидуального задания, построение праволинейной грамматики на основе индивидуального задания, приводится граф этой грамматики.

В разделе 3 приводится процедура перевода праволинейной грамматики в автоматную, описывается построение автоматной грамматики.

В разделе 4 приводится процедура построения недетерминированного автомата по автоматной грамматике, описывается

построение таблицы переходов и графа переходов недетерминированного автомата.

В разделе 5 приводится процедура перевода недетерминированного автомата в детерминированный, описывается построение таблицы переходов и графа переходов детерминированного автомата.

В разделе 6 описывается метод минимизации автомата, приводится процедура минимизации автомата, описывается построение таблицы переходов и графа переходов минимального автомата.

В разделе 7 описывается процедура построения сети Петри по праволинейной грамматике. Описывается построение для заданной праволинейной грамматики исходной сети Петри, описывается минимизация сети Петри – построение сети с позициями свободного выбора (недетерминированной сети), описывается устранение недетерминированности – построение детерминированной сети Петри. Описывается построение по полученной детерминированной сети Петри графа переходов минимального автомата. Производится сравнение графа автомата, построенного на основе сети Петри, с графом минимального автомата, полученным ранее.

# СПИСОК ЛИТЕРАТУРЫ

1. Льюис Ф., Розенкранц Д., Стирнз Р. «Теоретические основы проектирования компиляторов». – М.: Мир, 1979.
2. Компаниец Р. И., Маньяков Е. В., Филатов Н. Е., «Системное программирование. Основы построения трансляторов». – СПб.: КОРОНА принт, 2000.
3. Мозговой М. В. «Классика программирования: алгоритмы, языки, автоматы, компиляторы. Практический подход». – СПб.: Наука и Техника, 2006.
4. Молчанов А. Ю. «Системное программное обеспечение: Учебник для вузов». – СПб.: Питер, 2003.
5. Синтез распознающего автомата. Методические указания к курсовой работе. – Новочеркасск: издание НПИ, 1987.

*Учебное издание*

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

**к курсовой работе по дисциплине**

**«Теория языков программирования и методы трансляции»**

Сенилов Михаил Андреевич

(составление)

*В редакции составителя*

Подписано в печать . Формат 60х84/16. Бумага офсетная.

Гарнитура Таймс. Усл. печ. л. 0,93. Уч.-изд. л. 1,12. Тираж 50 экз. Заказ №

Отпечатано и изготовлено в Издательстве ИжГТУ.

426069, Ижевск, Студенческая ул., 7