**Задача**

Для грамматики G = ({S, A, B}, {**if**, **do**, **otherwise**, **true**, **false**, ';', **id**, **funcall**, '(', ')'}, P, S) с правилами

P = {S  **if** B **do** S | **if** B **do** S **otherwise** S | A;, A  **funcall id** (A) | **funcall id**, B  **true** | **false**} построить SLR(1)-анализатор и выполнить разбор цепочки w = "**if false do funcall id(funcall id); otherwise funcall id;**". Конфликт типа «перенос/свертка» разрешать в пользу переноса входного символа.

**Решение**

1) Построим пополненную грамматику G'.

G' = ({S', S, A, B}, {**if**, **do**, **otherwise**, **true**, **false**, ';', **id**, **funcall**, '(', ')'}, P', S'), где

P' = {

S'  S (0)

S  **if** B **do** S (1)

S  **if** B **do** S **otherwise** S (2)

S  A; (3)

A  **funcall id** (A) (4)

A  **funcall id** (5)

B  **true** (6)

B  **false (7)**

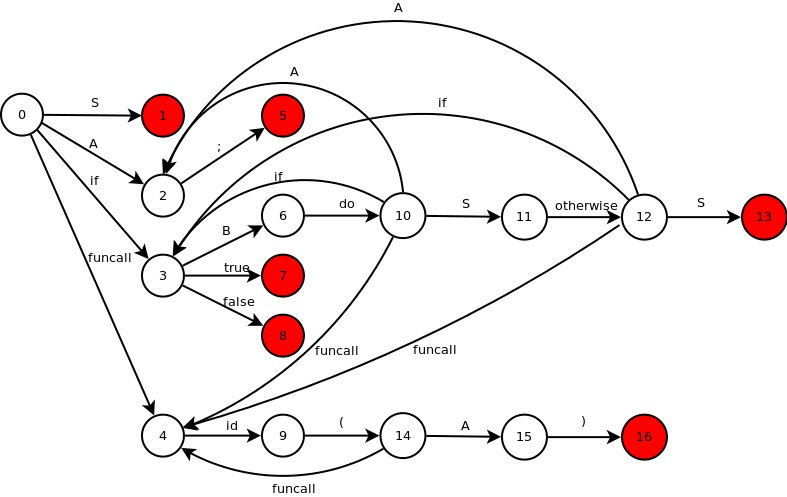
}

2) Построим таблицу для SLR-анализа

* + 1. Построим С = {I0, I1, …, In}- систему множеств LR(0)-ситуаций для грамматики G'.

|  |  |
| --- | --- |
| I0:  S'  · S S  · **if** B **do** S S  · **if** B **do** S **otherwise** S S  · A; A  · **funcall id** (A) A  · **funcall id** | I9:  A  **funcall id** · (A) A  **funcall id** · |
| I1:  S'  S · | I10:  S  **if** B **do** · S S  **if** B **do** · S **otherwise** S S  · **if** B **do** S S  · **if** B **do** S **otherwise** S S  · A; A  · **funcall id** (A) A  · **funcall id** |
| I2:  S  A · ; | I11:  S  **if** B **do** S · S  **if** B **do** S · **otherwise** S |
| I3:  S  **if** · B **do** S S  **if** · B **do** S **otherwise** S B  · **true** B  · **false** | I12:  S  **if** B **do** S **otherwise** · SS  · **if** B **do** S S  · **if** B **do** S **otherwise** S S  · A; A  · **funcall id** (A) A  · **funcall id** |
| I4:  A  **funcall** · **id** (A) A  **funcall** · **id** | I13:  S  **if** B **do** S **otherwise** S · |
| I5:  S  A ; · | I14:  A  **funcall id** ( · A) A  · **funcall id** (A) A  · **funcall id** |
| I6:  S  **if** B · **do** S S  **if** B · **do** S **otherwise** S | I15:  A  **funcall id** ( A · ) |
| I7:  B  **true** · | I16:  A  **funcall id** ( A ) · |
| I8:  B  **false** · |  |

Функция переходов goto в виде диаграммы переходов представлена на рис. 1.

Рис. 1. Функция goto в виде диаграммы переходов

Найдем значения функции FIRST(X) для всех нетерминалов грамматики G', кроме S'. Для любого терминала грамматики a выполняется равенство FIRST(a) = {a}.

|  |  |  |
| --- | --- | --- |
| FIRST(S) | FIRST(A) | FIRST(B) |
| if funcall | funcall | true false |

Найдем значения функции FOLLOW(X) для всех нетерминалов грамматики G', кроме S'.

|  |  |  |
| --- | --- | --- |
| FOLLOW(S) | FOLLOW(A) | FOLLOW(B) |
| $ otherwise | ) ; | do |

* + 1. Заполним таблицу

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | if | do | otherwise | true | false | ; | id | funcall | ( | ) | $ | S | A | B |
| 0 | S3 |  |  |  |  |  |  | S4 |  |  |  | 1 | 2 |  |
| 1 |  |  | R0 |  |  |  |  |  |  |  | доп |  |  |  |
| 2 |  |  |  |  |  | S5 |  |  |  |  |  |  |  |  |
| 3 |  |  |  | S7 | S8 |  |  |  |  |  |  |  |  | 6 |
| 4 |  |  |  |  |  |  | S9 |  |  |  |  |  |  |  |
| 5 |  |  | R3 |  |  |  |  |  |  |  | R3 |  |  |  |
| 6 |  | S10 |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  | R6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  | R7 |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  | R5 |  |  | S14 | R5 |  |  |  |  |
| 10 | S3 |  |  |  |  |  |  | S4 |  |  |  | 11 | 2 |  |
| 11 |  |  | S12 |  |  |  |  |  |  |  | R1 |  |  |  |
| 12 | S3 |  |  |  |  |  |  | S4 |  |  |  | 13 | 2 |  |
| 13 |  |  | R2 |  |  |  |  |  |  |  | R2 |  |  |  |
| 14 |  |  |  |  |  |  |  | S4 |  |  |  |  | 15 |  |
| 15 |  |  |  |  |  |  |  |  |  | S16 |  |  |  |  |
| 16 |  |  |  |  |  | R4 |  |  |  | R4 |  |  |  |  |

2) Проведем разбор цепочки w = "**if false do funcall id(funcall id); otherwise funcall id;**"с использованием полученной таблицы

|  |  |  |
| --- | --- | --- |
| Стек | Входной поток | Действие |
| 0 | **if false do funcall id(funcall id); otherwise funcall id;** | S3 |
| 0if3 | **false do funcall id(funcall id); otherwise funcall id;** | S8 |
| 0if3false8 | **do funcall id(funcall id); otherwise funcall id;** | R7 |
| 0if3B6 | **do funcall id(funcall id); otherwise funcall id;** | S10 |
| 0if3B6do10 | **funcall id(funcall id); otherwise funcall id;** | S4 |
| 0if3B6do10funcall4 | **id(funcall id); otherwise funcall id;** | S9 |
| 0if3B6do10funcall4id9 | **(funcall id); otherwise funcall id;** | S14 |
| 0if3B6do10funcall4id9(14 | **funcall id); otherwise funcall id;** | S4 |
| 0if3B6do10funcall4id9(14funcall4 | **id); otherwise funcall id;** | S9 |
| 0if3B6do10funcall4id9(14funcall4id9 | **); otherwise funcall id;** | R5 |
| 0if3B6do10funcall4id9(14A15 | **); otherwise funcall id;** | S16 |
| 0if3B6do10funcall4id9(14A15)16 | **; otherwise funcall id;** | R4 |
| 0if3B6do10A2 | **; otherwise funcall id;** | S5 |
| 0if3B6do10A2;5 | **otherwise funcall id;** | R3 |
| 0if3B6do10S11 | **otherwise funcall id;** | S12 |
| 0if3B6do10S11otherwise12 | **funcall id;** | S4 |
| 0if3B6do10S11otherwise12funcall4 | **id;** | S9 |
| 0if3B6do10S11otherwise12funcall4id9 | **;** | R5 |
| 0if3B6do10S11otherwise12A2 | **;** | S5 |
| 0if3B6do10S11otherwise12A2;5 |  | R3 |
| 0if3B6do10S11otherwise12S13 |  | R2 |
| 0S1 |  | допуск |