

Общие ненаправленные алгоритмы анализа СУК. НФХ.

В. С. Полозов

Кафедра системного программирования СПбГУ



Теория автоматов и формальных языков

Метод СΥΚ

Назван по именам создателей: J. Cocke, D.H. Younger, и T. Kasami.

Две фазы

- 1 Строится таблица соответствия выводам подстрок входного предложения нетерминалам
- 2 Используя построенную таблицу строим все возможные выводы

Пример

$Numbers_S \rightarrow Integer \mid Real$
 $Integer \rightarrow Digit \mid Integer\ Digit$
 $Real \rightarrow Integer\ Fraction\ Scale$
 $Fraction \rightarrow .\ Integer$
 $Scale \rightarrow e\ Sign\ Integer \mid Empty$
 $Digit \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$
 $Empty \rightarrow \varepsilon$
 $Sign \rightarrow + \mid -$

и вход – 23.4e+5

- Что делать с ε ?

Рассмотрим вход – 43.1

Пример

$Numbers_S \rightarrow Integer \mid Real$
 $Integer \rightarrow Digit \mid Integer\ Digit$
 $Real \rightarrow Integer\ Fraction\ Scale$
 $Fraction \rightarrow .\ Integer$
 $Scale \rightarrow e\ Sign\ Integer \mid Empty$
 $Digit \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$
 $Empty \rightarrow \varepsilon$
 $Sign \rightarrow + \mid -$

и вход – 23.4e+5

- Что делать с ε ?

Рассмотрим вход – 43.1

Обозначения

Введем обозначения:

- $R_\varepsilon = \{Empty, Scale\}$ - множество нетерминалов A таких, что $A \rightarrow \varepsilon$
- $s_{i,l}$ - подстрока $z_i z_{i+1} \dots z_{i+l-1}$ строки $z = z_1 z_2 \dots z_n$ начиная с позиции i и длиной l .
- $R_{s_{i,l}}$ - множество нетерминалов из которых выводима строка $s_{i,l}$.
- Для строк длины 0 положим: $s_{i,0} = \varepsilon$ и $R_{s_{i,0}} = R_\varepsilon$

Замечание

Основная переборная сложность CYK связана с особенностями грамматики:

- Цепные правила: $A \rightarrow B \rightarrow \dots \rightarrow C \xrightarrow{*} s_{i,l}$
- ε -правила

Обозначения

Введем обозначения:

- $R_\varepsilon = \{Empty, Scale\}$ - множество нетерминалов A таких, что $A \rightarrow \varepsilon$
- $s_{i,l}$ - подстрока $z_i z_{i+1} \dots z_{i+l-1}$ строки $z = z_1 z_2 \dots z_n$ начиная с позиции i и длиной l .
- $R_{s_{i,l}}$ - множество нетерминалов из которых выводима строка $s_{i,l}$.
- Для строк длины 0 положим: $s_{i,0} = \varepsilon$ и $R_{s_{i,0}} = R_\varepsilon$

Замечание

Основная переборная сложность СЮК связана с особенностями грамматики:

- Цепные правила: $A \rightarrow B \rightarrow \dots \rightarrow C \xrightarrow{*} s_{i,l}$
- ε -правила

СΥК: НФХ

Определение

Нормальная форма Хомского (CNF): все правила имеют вид $A \rightarrow a$ или $A \rightarrow BC$.

Получаем:

- $R_\varepsilon = \emptyset$
- $R_{s_{i,1}}$ вычисляется непосредственно из правил $A \rightarrow a$.
- Для правила $A \rightarrow BC$
из B должно выводиться $s_{i,k}$, т.е. $B \in R_{s_{i,k}}$,
из C должно выводиться $s_{i+k,l-k}$, т.е. $C \in R_{s_{i+k,l-k}}$

СΥΚ: НФХ

Т.е. можно организовать *таблицу распознавания* (рис.).

$R_{s_1,n}$					
$R_{s_1,n-1}$	$R_{s_2,n-1}$				
...			
$R_{s_1,2}$	$R_{s_{n-1},2}$		
$R_{s_1,1}$	$R_{s_2,1}$...	$R_{s_{n-1},1}$	$R_{s_n,1}$	
s_1	s_2	...	s_{n-1}	s_n	

$R_{i,j} = ?$

Соответственно, можно изменить порядок обхода:

- off-line order
- on-line order

Сложность?

СΥΚ: НФХ

Т.е. можно организовать *таблицу распознавания* (рис.).

$R_{s_1,n}$					
$R_{s_1,n-1}$	$R_{s_2,n-1}$				
...			
$R_{s_1,2}$	$R_{s_{n-1},2}$		
$R_{s_1,1}$	$R_{s_2,1}$...	$R_{s_{n-1},1}$	$R_{s_n,1}$	
s_1	s_2	...	s_{n-1}	s_n	

$R_{i,j} = ?$

Соответственно, можно изменить порядок обхода:

- off-line order
- on-line order

Сложность?

СΥΚ: НФХ

Т.е. можно организовать *таблицу распознавания* (рис.).

$R_{s_1,n}$					
$R_{s_1,n-1}$	$R_{s_2,n-1}$				
...			
$R_{s_1,2}$	$R_{s_{n-1},2}$		
$R_{s_1,1}$	$R_{s_2,1}$...	$R_{s_{n-1},1}$	$R_{s_n,1}$	
s_1	s_2	...	s_{n-1}	s_n	

$R_{i,j} = ?$

Соответственно, можно изменить порядок обхода:

- off-line order
- on-line order

Сложность?

СΥΚ: НФХ

Т.е. можно организовать *таблицу распознавания* (рис.).

$R_{s_1,n}$					
$R_{s_1,n-1}$	$R_{s_2,n-1}$				
...			
$R_{s_1,2}$	$R_{s_{n-1},2}$		
$R_{s_1,1}$	$R_{s_2,1}$...	$R_{s_{n-1},1}$	$R_{s_n,1}$	
s_1	s_2	...	s_{n-1}	s_n	

$R_{i,j} = ?$

Соответственно, можно изменить порядок обхода:

- off-line order
- on-line order

Сложность?

Преобразование к НФХ

Преобразование к НФХ:

- удаление ε -правил
- удаление цепных правил ($A \rightarrow B$)
- удаление непродуктивных нетерминалов
- удаление недостижимых нетерминалов
- оставшаяся грамматика преобразуется добавлением правил, пока все правила не будут вида $A \rightarrow a$ или $A \rightarrow BC$.

Удаление ε -правил

Грамматика $G = (V_T, V_N, P, S)$

- 1 Для правила $A \rightarrow \varepsilon$
 - Правила вида $B \rightarrow \alpha A \beta$ заменим на $B \rightarrow \alpha A' \beta$ и $B \rightarrow \alpha \beta$
 - Повторить, в т.ч. и для новых правил
- 2 Повторить для всех ε -правил
- 3 Добавим нетерминал A' для всех нетерминалов A :
 $A' \rightarrow \alpha$, если $A \rightarrow \alpha \in P$ и $\alpha \neq \varepsilon$

Удаление цепных правил

Грамматика $G = (V_T, V_N, P, S)$

- 1 Для правила $A \rightarrow B$, где
$$B \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n$$
$$A \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n$$
- 2 Повторить, кроме правил $A \rightarrow A$

Удаление непродуктивных нетерминалов

Идея: непродуктивными являются нетерминалы, которые не являются продуктивными.

Удаление недостижимых нетерминалов

Идея: недостижимыми являются нетерминалы, которые не являются достижимыми.

Замечание

Удаление недостижимых нетерминалов не добавляет непродуктивных нетерминалов. (Обратное неверно)

Преобразование к CNF

Грамматика $G = (V_T, V_N, P, S)$, без ε -правил, цепных правил, все нетерминалы достижимы и продуктивны.

Осталось два типа правил:

- 1 $A \rightarrow a$ – уже в хорошей форме
- 2 $A \rightarrow X_1 X_2 \dots X_m, m \geq 2$
 - Для каждого терминала вводим новый нетерминал $T_b \rightarrow b$
 - Заменить вхождения
 - Повторить
- 3 $A \rightarrow X_1 X_2 \dots X_m, m \geq 3, X_i \in V_N$
 - Заменить правило на
 $A \rightarrow A_1 X_3 \dots X_m$
 $A_1 \rightarrow X_1 X_2$
 - Повторить

Пример

Numbers → 0|1|2|3|4|5|6|7|8|9
Numbers → *Integer Digit*
Numbers → *N1 Scale' | Integer Fraction*
 N1 → *Integer Fraction*
 Integer → 0|1|2|3|4|5|6|7|8|9
 Integer → *Integer Digit*
Fraction → *T1 Integer*
 T1 → .
 Scale' → *N2 Integer*
 N2 → *T2 Sign*
 T2 → *e*
 Digit → 0|1|2|3|4|5|6|7|8|9
 Sign → +|-

Откат преобразования к CNF

При преобразовании к CNF могут потеряться нетерминалы:

- при удалении ϵ -правил (В примере: Scale, Real, Empty)
- удаление unit-правил (В примере: Real)

Дополним грамматику правилом:

Real \rightarrow *N1 Scale'* | *Integer Fraction*

Откат преобразования к CNF

При преобразовании к CNF могут потеряться нетерминалы:

- при удалении ϵ -правил (В примере: Scale, Real, Empty)
- удаление unit-правил (В примере: Real)

Дополним грамматику правилом:

$$Real \rightarrow N1 \text{ Scale}' \mid Integer \text{ Fraction}$$

Вопросы