

第2章 LR 構文解析 (教科書 p.72)

2.1 LR 構文解析の特徴

- パーサーの _____ (一方、人手での生成には向かない)
→ Yacc, Bison などの構文解析器生成系
- 取り扱える文法の範囲が広い
- 演算子順位法と同様に、 _____ ・ _____ である
(“LR” は Left-to-Right Rightmost derivation に由来する)
 - スタックを用いるのは同じ
 - シフト/還元の判断法がちがう (Bison のプログラム ~.y をデバッグするとき判断法の知識が必要になる)
_____ を用いる (ただしスタックの方に!!)
この DFA は直観的には BNF の右辺のどこまで処理しているか? を表す

LR 構文解析の例

例 1

$$\begin{aligned} S' &\rightarrow S \$ \\ S &\rightarrow (L) \\ &\quad | x \\ L &\rightarrow S \\ &\quad | L , S \end{aligned}$$

注: LL 法 (後述) は左再帰は扱えないが、LR 法は左再帰の文法のほうが効率が良い。

問 2.1.1 (復習) この BNF の S から導出される終端記号列の例を導出列とともに3つ挙げよ

この BNF に対応する DFA は以下ようになる (ただし、DFA の作成法は講義の範囲外である)。

※ ... スタックをもう一度 DFA にかける

実際にはスタック全体に繰り返し DFA を適用する必要はない。下の表のようにスタックに状態も積んでおくと良い

スタック	入力
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
⋮	

問 2.1.2 例 1 の BNF に対して、以下の入力例に対する LR 構文解析の過程を書け。

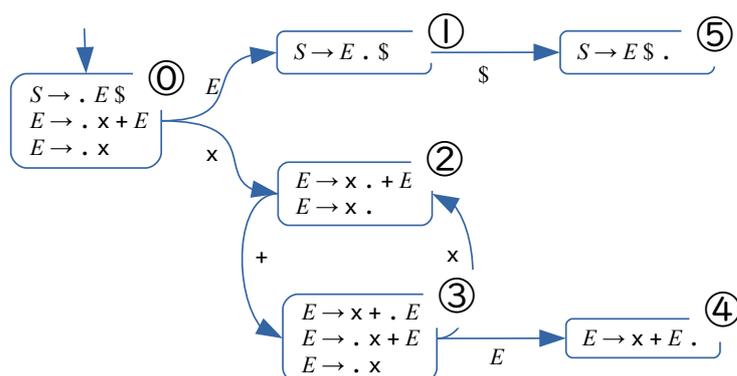
1. ((x), x)\$
2. (x, (x))\$

長くなるようならば、5 回還元 (reduce) した時点で止めて良い。

例 2

一般にシフト/還元の判定には"先読み" (入力の先頭) も用いる

- $$S \rightarrow E \$ \quad (1)$$
- $$E \rightarrow x + E \quad (2) \text{ 右再帰になっている}$$
- $$| x \quad (3)$$



状態②では 先読みが + → shift
先読みが + 以外 → reduce

LR 構文解析の階層

先読みの利用の仕方によって、以下のような階層がある

問 2.1.3 次の BNF に対して LR 構文解析表を作成すると、下のようになる。 S' , S, T は非終端記号であり、 $;$, w , $\{, \}$ は終端記号である。

$$\begin{aligned}
 S' &\rightarrow S \$ & (0) \\
 S &\rightarrow ; & (1) \\
 &| \{ T \} & (2) \\
 &| w S & (3) \\
 T &\rightarrow S T & (4) \\
 &| \varepsilon & (5)
 \end{aligned}$$

状態 \ 先読み	$;$	$\{$	w	$\}$	$\$$		S	T
①	shift①	shift②	shift③				goto④	
①	reduce(1)							
②	shift①	shift②	shift③	reduce(5)			goto⑤	goto⑥
③	shift①	shift②	shift③				goto⑦	
④					shift⑧			
⑤	shift①	shift②	shift③	reduce(5)			goto⑤	goto⑨
⑥				shift⑩				
⑦	reduce(3)							
⑧	accept							
⑨	reduce(4)							
⑩	reduce(2)							

以下の入力例に対する LR 構文解析の過程を書け。

- $\{ ; w ; \} \$$
- $\{ w \{ ; \} ; \} \$$

長くなるようならば、5 回還元 (reduce) した時点で止めてよい。

LR 構文解析と曖昧な文法

曖昧な文法に対して無理に LR 構文解析表を作ると _____ (conflict) が起こる (つまり、表の 1 つの場所に複数の動作が入る)

例 3

$$E \rightarrow x \mid E * E \mid E + E$$

_____ が起こる。

例 $x + x * x$

Yacc (Bison) では演算子の _____ ・ _____ を指定して conflict を解消できる

例 4 (_____ , dangling else)

$$\begin{aligned}
 S &\rightarrow \text{if} (E) S \\
 &| \text{if} (E) S \text{ else } S \\
 &| \dots
 \end{aligned}$$

例 `if (E1) if (E2) S1 else S2` ← これは
`if (E1) {if (E2) S1} else S2` と解釈するの
か?
`if (E1) {if (E2) S1 else S2}` と解釈するの
か?

_____ が起こる。Yacc (Bison) では _____ を採用する

例 5 (特殊例の優先)

$$E \rightarrow E \wedge E _ E \mid E \wedge E \mid E _ E \mid \text{id}$$

(優先順位・結合性を与えても) _____ が起こる

例 `x ^ y _ z $`

Yacc (Bison) では先に書かれている生成規則を優先する

これらの例 3 ~ 5 はよく知られている形だが、このような形以外の conflict は
文法を見直す
