

コンパイラ
 第12回 上昇型構文解析(2)

http://www.info.kindai.ac.jp/compiler
 E館3階E-331 内線5459
 takasi-i@info.kindai.ac.jp

1

移動還元構文解析 (shift reduce parsing)

- 上昇型構文解析
- 構文解析表とスタックで解析

スタック 入力記号列

\$	(4	+	x)	*	...	\$
----	---	---	---	---	---	---	-----	----

トップに "\$" 末尾に "\$"

2

移動還元構文解析 (shift reduce parsing)

- 構文解析表とスタックで解析
 - 初期状態
 - 入力記号列末尾に "\$"
 - スタックトップに "\$"
 - if (スタックトップが生成規則の右辺に一致)
 - 生成規則の右辺をポップ, 左辺をプッシュ (還元)
 - else 入力記号をプッシュ (移動)
 - if (スタックトップが "\$" && 入力記号が "\$")
 - 解析終了

3

還元(reduce)

- 還元
 - 生成規則の右辺から左辺に戻す
- ハンドル
 - 右辺に一致する部分

<namelist> ::= <name> | <namelist> “,” <name>
 <name> ::= “a” | “b” | “c”

“a” “,” “b”
 → <name> “,” “b” <name> → “a” の還元
 → <namelist> “,” “b” <namelist> → <name> の還元

ハンドル

4

還元

生成規則
 スタック P = {E→T+T, T→F*F, F→i, F→(E)}

\$	\$
((
T	E
+	
T	

右辺 T+T をポップ
 左辺 E をプッシュ

E→T+T の
 右辺と一致
 = ハンドル

出力
 ADD

対応する
 コードを出力

5

移動(shift)

生成規則
 スタック P = {E→T+T, T→F*F, F→i, F→(E)}

\$	\$
((
E)
	...
	...
	...
	\$

右辺と
 不一致

入力記号を
 プッシュ

6

移動と還元

- 移動：右辺を読み込み途中
- 還元：右辺を読み込み完了

例： $E \rightarrow T + T \Rightarrow$ 読み込み位置を移動
 \uparrow 読み込み位置
 $E \rightarrow T_+ T \Rightarrow$ 読み込み位置を移動
 \uparrow 読み込み位置
 $E \rightarrow T + T_ \Rightarrow$ 読み込み位置を移動
 \uparrow 読み込み位置
 $E \rightarrow T + T \uparrow \Rightarrow$ 読み込み完了, 還元する
 \uparrow 読み込み位置

7

移動還元構文解析の問題点

- 生成規則が複数ある場合
 - 例： $\$ \rightarrow E\$$, $E \rightarrow T+T$, $E \rightarrow T$, $T \rightarrow i$, $T \rightarrow n$ $E \rightarrow T [+T]$ を解析
 \uparrow 解析位置
 - { $\Rightarrow E \rightarrow T+T$ の一部と見做して移動
 - { $\Rightarrow E \rightarrow T$ と見做して還元

どちらの規則を使用する？

8

例： $\$ x + 1 \$ \{ \$ \rightarrow E\$$, $E \rightarrow T+T$, $E \rightarrow T$, $T \rightarrow i$, $T \rightarrow n \}$

スタック	入力記号	操作
\$	x + 1 \$	移動
\$ x	+ 1 \$	還元 $T \rightarrow n$
\$ T	+ 1 \$	還元 $E \rightarrow T$
\$ E	+ 1 \$	移動
\$ E +	1 \$	移動
\$ E + 1	\$	還元 $T \rightarrow i$
\$ E + T	\$	還元 $E \rightarrow T$
\$ E + E	\$	移動
\$ E + E \$	ϵ	還元 $\$ \rightarrow E\$$
\$ E + \$	ϵ	解析失敗

移動に変更

バックトラック

9

例： $\$ x + 1 \$ \{ \$ \rightarrow E\$$, $E \rightarrow T+T$, $E \rightarrow T$, $T \rightarrow i$, $T \rightarrow n \}$

スタック	入力記号	操作
\$	x + 1 \$	移動
\$ x	+ 1 \$	還元 $T \rightarrow n$
\$ T	+ 1 \$	移動
\$ T +	1 \$	移動
\$ T + 1	\$	還元 $T \rightarrow i$
\$ T + T	\$	還元 $E \rightarrow T$
\$ T + E	\$	移動
\$ T + E \$	ϵ	還元 $\$ \rightarrow E\$$
\$ T + \$	ϵ	解析失敗

$E \rightarrow T+T$ に変更

バックトラック

10

例： $\$ x + 1 \$ \{ \$ \rightarrow E\$$, $E \rightarrow T+T$, $E \rightarrow T$, $T \rightarrow i$, $T \rightarrow n \}$

スタック	入力記号	操作
\$	x + 1 \$	移動
\$ x	+ 1 \$	還元 $T \rightarrow n$
\$ T	+ 1 \$	移動
\$ T +	1 \$	移動
\$ T + 1	\$	還元 $T \rightarrow i$
\$ T + T	\$	還元 $E \rightarrow T+T$
\$ E	\$	移動
\$ E \$	\$	還元 $\$ \rightarrow E\$$
\$ \$	ϵ	解析完了

11

移動還元構文解析の問題点

- 生成規則が複数ある場合
 - 例： $\$ \rightarrow E\$$, $E \rightarrow T+T$, $E \rightarrow T$, $T \rightarrow i$, $T \rightarrow n$

E の後には必ず $\$$

還元 $E \rightarrow T$ よりも $E \rightarrow T+T$ を優先 (最長一致)

E への還元はその後に “\$” が来る場合のみ
 \Rightarrow 後に来る記号で判断

Follow 集合を用いる

12

Follow集合 (後続終端記号)

- Follow (A) = { "b" | S ⇒ αAbβ }
非終端記号 <A> の直後に来る終端記号の集合

例 : <S> ::= "a" | <C>
 ::= "b" | ε { a, bc, c }
 <C> ::= "c"

Follow (S) = { \$ } (\$: 入力の末尾)
 Follow (C) = { \$ }
 Follow (B) = First (C) = { "c" }

13

Follow 集合の求め方

- Follow (A) の求め方 (A ∈ N)
 - 初期状態では Follow (A) = φ (空集合)

- A = S のとき
Follow (S) += { "\$" };
- 生成規則 X → αAβ があるとき
Follow (A) += (First (β) - { ε });
- 生成規則 X → αA があるとき
Follow (A) += Follow (X)

14

Follow集合を用いた 移動還元構文解析

- 例 : S → E\$, E → T+\$, E → T, T → i, T → n

Follow (E) = { "\$" }
 Follow (T) = { "\$", "+" }

E の還元は次に "\$" が来るときのみ
 T の還元は次に "\$" が "+" が来るときのみ

15

例 : \$ x + 1 \$ { S → E\$, E → T+\$, E → T, T → i, T → n }

スタック	入力記号	Follow	操作
\$	x + 1 \$		移動
\$ x	+ 1 \$	+ ∈ Follow (T)	還元 T → n
		先読み記号 : +	
		+ ∈ Follow (T) なので	
		T → n で還元していい	

Follow (E) = { "\$" }
 Follow (T) = { "\$", "+" }

16

例 : \$ x + 1 \$ { S → E\$, E → T+\$, E → T, T → i, T → n }

スタック	入力記号	Follow	操作
\$	x + 1 \$		移動
\$ x	+ 1 \$	+ ∈ Follow (T)	還元 T → n
\$ T	+ 1 \$	+ ∉ Follow (E)	移動
		先読み記号 : +	
		+ ∉ Follow (E) なので	
		E → T の還元はできない	

Follow (E) = { "\$" }
 Follow (T) = { "\$", "+" }

17

例 : \$ x + 1 \$ { S → E\$, E → T+\$, E → T, T → i, T → n }

スタック	入力記号	Follow	操作
\$	x + 1 \$		移動
\$ x	+ 1 \$	+ ∈ Follow (T)	還元 T → n
\$ T	+ 1 \$	+ ∉ Follow (E)	移動
\$ T +	1 \$		移動
\$ T + 1	\$	\$ ∈ Follow (T)	還元 T → i
		先読み記号 : \$	
		\$ ∈ Follow (T) なので	
		T → i で還元していい	

Follow (E) = { "\$" }
 Follow (T) = { "\$", "+" }

18

例: $\$ x + 1 \$ \{ \$ \rightarrow E \$, E \rightarrow T+T, E \rightarrow T, T \rightarrow i, T \rightarrow n \}$

スタック	入力記号	Follow	操作
\$	x + 1 \$		移動
\$ x	+ 1 \$	$+ \in \text{Follow}(T)$	還元 $T \rightarrow n$
\$ T	+ 1 \$	$+ \notin \text{Follow}(E)$	移動
\$ T +	1 \$		移動
\$ T + 1	\$	$\$ \in \text{Follow}(T)$	還元 $T \rightarrow i$
\$ T + T	\$	$\$ \in \text{Follow}(E)$	還元 $E \rightarrow T+T$
先読み記号: \$			
$\$ \in \text{Follow}(E)$ なので			
Follow (E) = { "\$E → T, E → T+T で還元していい"			
Follow (T) = { "\$E → T+T を優先"			

19

例: $\$ x + 1 \$ \{ \$ \rightarrow E \$, E \rightarrow T+T, E \rightarrow T, T \rightarrow i, T \rightarrow n \}$

スタック	入力記号	Follow	操作
\$	x + 1 \$		移動
\$ x	+ 1 \$	$+ \in \text{Follow}(T)$	還元 $T \rightarrow n$
\$ T	+ 1 \$	$+ \notin \text{Follow}(E)$	移動
\$ T +	1 \$		移動
\$ T + 1	\$	$\$ \in \text{Follow}(T)$	還元 $T \rightarrow i$
\$ T + T	\$	$\$ \in \text{Follow}(E)$	還元 $E \rightarrow T+T$
\$ E	\$		移動
\$ E \$	ϵ		還元 $\$ \rightarrow E \$$
\$ \$	ϵ		解析完了

20

LR構文解析

- Left to right scan & Right most derivation 解析
 - 上昇型解析
 - スタックと解析表を用いて解析
 - 演算順位構文解析よりも広い文法を受理

演算子順位構文解析: 式を解析
LR構文解析: 全て解析

21

演算子順位構文解析の問題点

- 演算子順位構文解析の解析手順
 - スタックトップと入力記号で次の操作を決定

式の解析ならこれでOK
しかしプログラム全体だと?

22

演算子順位構文解析の問題点

例: "while" "(" "i" ")" "--" "i" ";"

スタック
 \$
 while
 (
 |
 |

↑読み込み位置
 入力記号列
 [i |) | -- | i | ; | \$]

"(" と "i" で次の操作を決定

"if" "(" "i" ")" "--" "i" ";"

スタック
 \$
 while
 (
 |
 |

↑読み込み位置
 入力記号列
 [i |) | -- | i | ; | \$]

スタックトップと入力と同じだが異なる操作が必要

23

演算順位構文解析の問題点

- スタックトップと入力だけでは解析不能

↓

状態を記憶する

状態: 過去の入力の履歴

状態記号: それより下のスタックの内容を表す記号

例: $E \rightarrow T(+T | -T)$

$\uparrow \uparrow \uparrow$
 $s_0 \ s_1 \ s_2$

s_0 : Tまで読んだ
 s_1 : T+まで読んだ
 s_2 : T-まで読んだ

24

状態の記憶

■ スタックに状態を記憶する

X_i : 文法記号 ($\in N \cup \Sigma$)
 s_j : 状態記号

文法記号と状態記号の対を
スタックに記憶



状態と入力で操作を決定

スタック

文法記号	状態記号
\$	s_0
X_1	s_1
X_2	s_2
X_3	s_3
:	:
X_m	s_m

25

構文解析表

■ action 表

- 構文解析の動作を決定 (移動, 還元, 受理)
- 状態 \times 終端記号 \rightarrow 動作 (移動, 還元, 受理)

■ goto 表

- 還元時の状態遷移を決定
- 状態 \times 非終端記号 \rightarrow 移動

26

action 表による動作

■ s: 移動 (shift)

- 入力記号と表で示された状態をプッシュ

■ r: 還元 (reduce)

- 表で示された規則によりスタックトップを還元
- goto 表で示された状態をプッシュ

■ a: 受理 (accept)

- 解析完了

27

構文解析表

■ 例: $\$ \rightarrow E\$$, $E \rightarrow E*F$, $E \rightarrow F$, $F \rightarrow i$

F を還元後
状態 0 ならば
状態 2 へ

状態	action			goto	
	i	*	\$	E	F
0	移動 3			移動 1	移動 2
状態 0 で i を読めば 状態 3 へ		移動 4	受理		
		還元 $E \rightarrow F$	還元 $E \rightarrow F$		
		還元 $F \rightarrow i$	還元 $F \rightarrow i$		
4	移動 3				移動 5
5		還元 $E \rightarrow E*F$	還元 $E \rightarrow E*F$		

空欄は構文解析エラー

状態 5 で
\$ を読めば
 $E \rightarrow E*F$ で還元

28

LR構文解析アルゴリズム

スタック: $(\$s_0)(X_1s_1)(X_2s_2)\dots(X_ms_m)$, 入力: a_i のとき

- action $[s_m, a_i]$ = 「移動 s 」の場合
 - (a_i, s) をプッシュ, 次の入力を読み込む
- action $[s_m, a_i]$ = 「還元 $A \rightarrow \beta$ 」の場合
 - $|\beta|$ 組の記号 (X_i, s_i) ($m-|\beta|+1 \leq i \leq m$) をポップ
 - $(A, \text{goto}[s_{m-|\beta|}, A])$ をプッシュ
 - $A \rightarrow \beta$ に対応するコードを生成
- action $[s_m, a_i]$ = 「受理」の場合
 - 解析完了

29

処理の手順 (初期状態)

入力末尾に \$

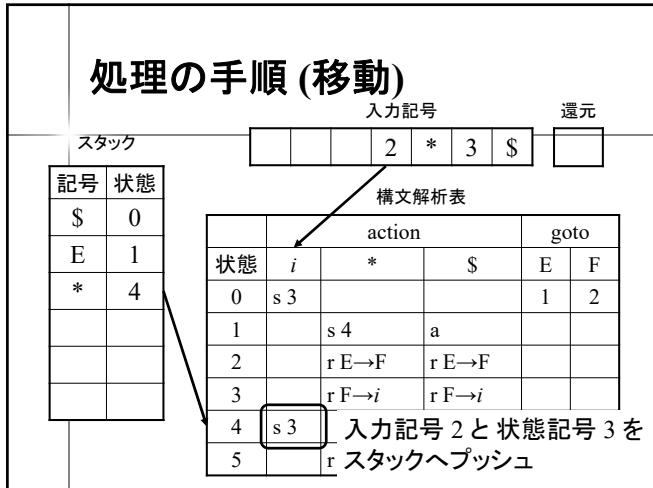
スタック		入力記号					還元
記号	状態	1	*	2	*	3	\$
\$	0						

状態	action			goto	
	i	*	\$	E	F
s 3				1	2
s 4					
2		r $E \rightarrow F$	r $E \rightarrow F$		
3		r $F \rightarrow i$	r $F \rightarrow i$		
4	s 3				5
5		r $E \rightarrow E*F$	r $E \rightarrow E*F$		

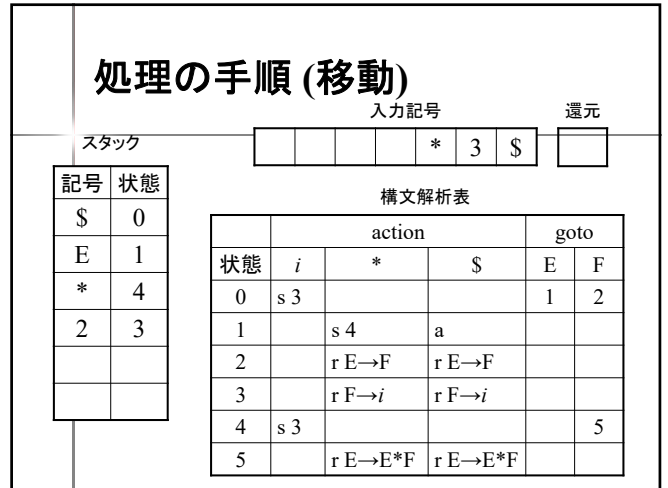
スタックトップには (開始記号, 開始状態)

動作: s: 移動, a: 受理, r: 還元

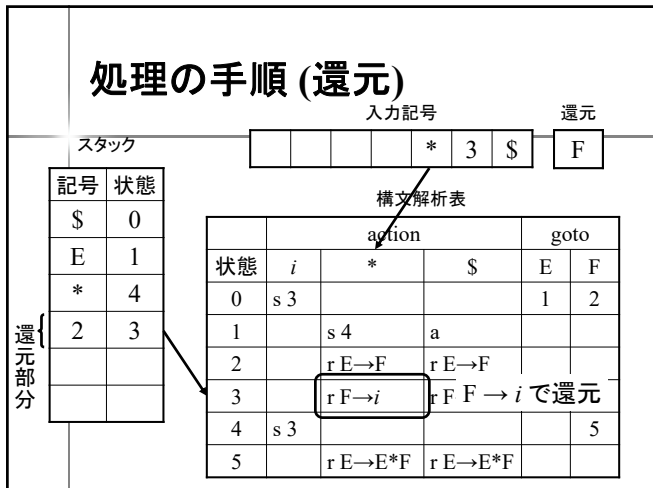
30



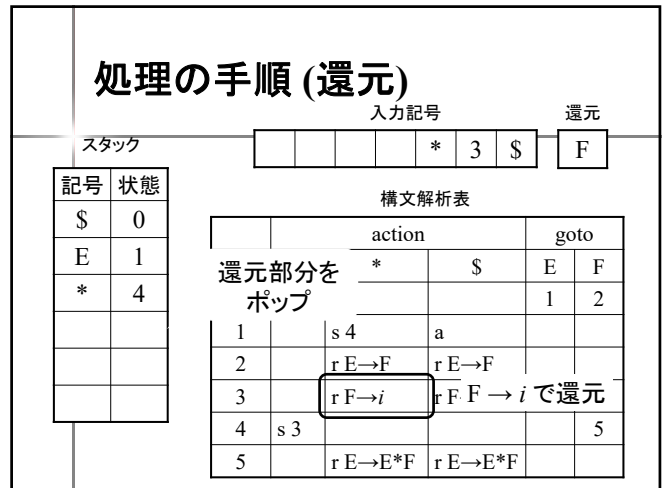
31



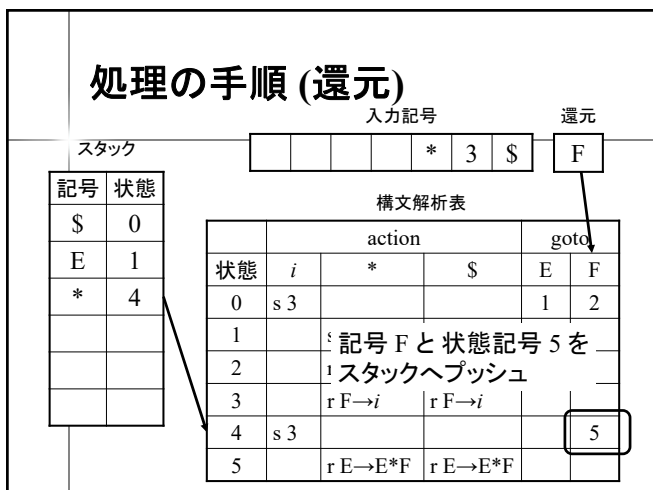
32



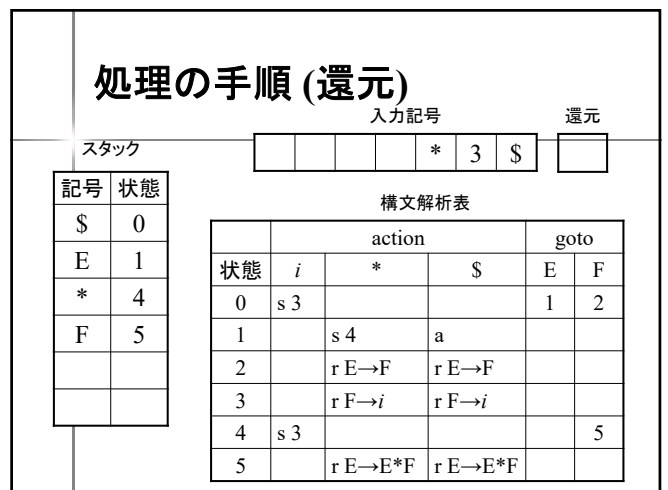
33



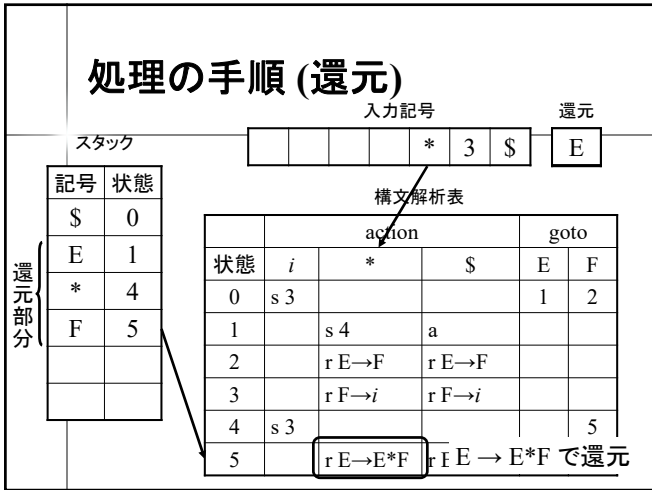
34



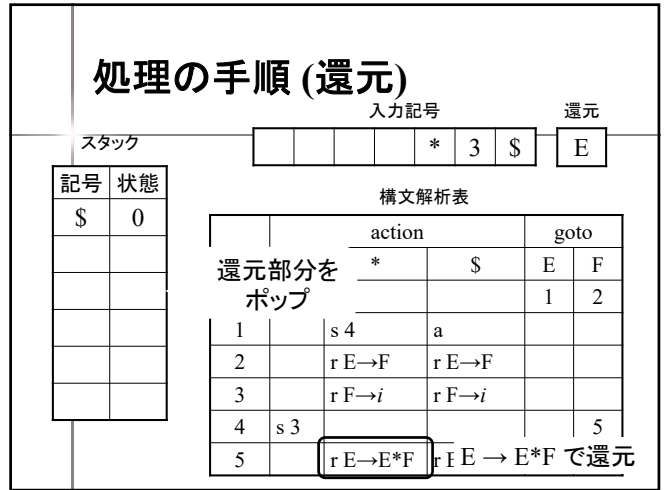
35



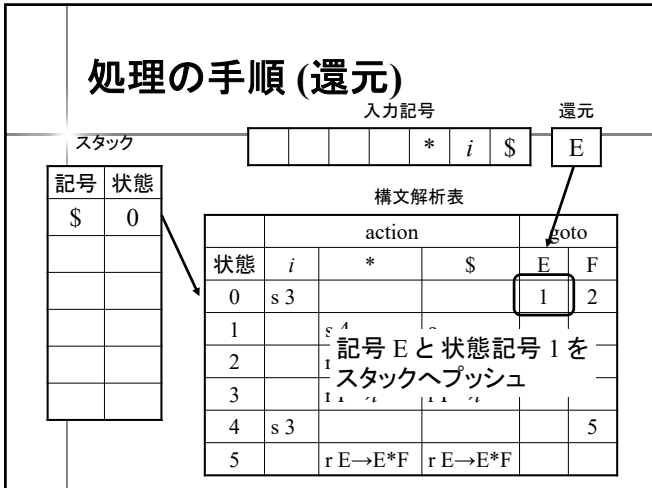
36



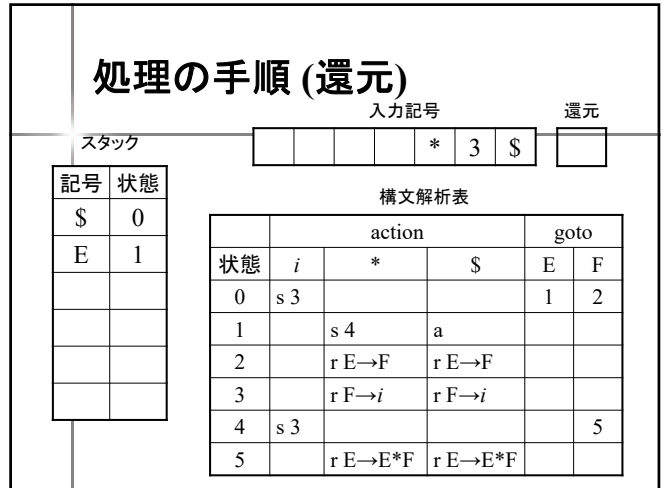
37



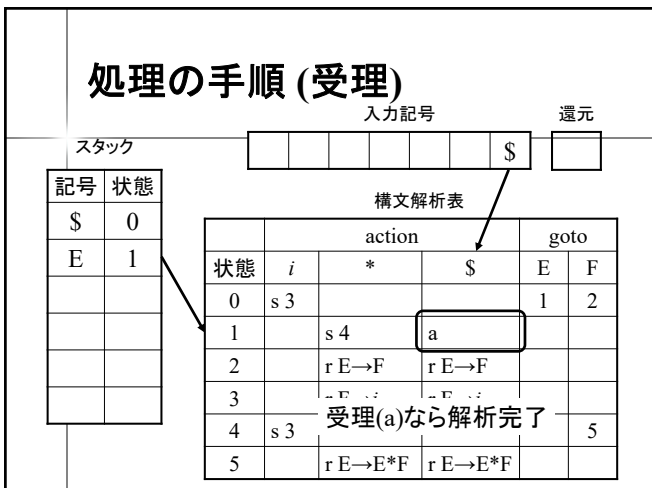
38



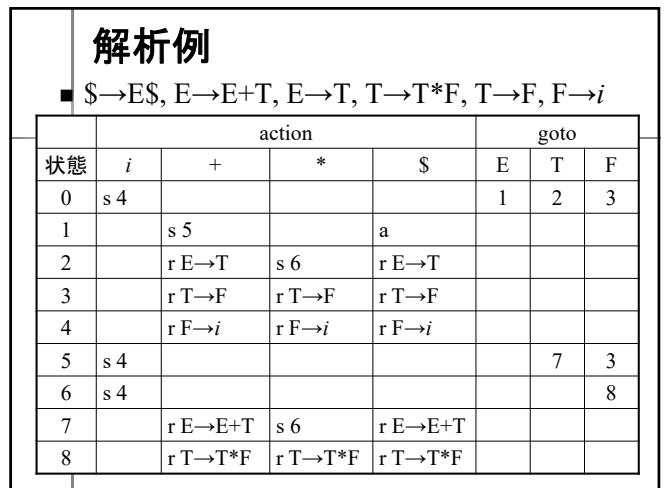
39



40



41



42

解析例

状態	action					goto		
	<i>i</i>	+	*	\$		E	T	F
0	s 4					1	2	3
1		s 5		a				
2		r E→T	s 6	r E→T				
3		r T→F	r T→F	r T→F				
4		r F→i	r F→i	r F→i				
5	s 4					7	3	
6	s 4							8
7		r E→E+T	s 6	r E→E+T				
8		r T→T*F	r T→T*F	r T→T*F				

スタック

記号	状態
\$	0

入力記号

1	+	2	*	3	\$
---	---	---	---	---	----

還元

--

初期状態ではスタックトップに (開始記号, 初期状態)

43

解析例

状態	action					goto		
	<i>i</i>	+	*	\$		E	T	F
0	s 4					1	2	3
1		s 5		a				
2		r E→T	s 6	r E→T				
3		r T→F	r T→F	r T→F				
4		r F→i	r F→i	r F→i				
5	s 4					7	3	
6	s 4							8
7		r E→E+T	s 6	r E→E+T				
8		r T→T*F	r T→T*F	r T→T*F				

スタック

記号	状態
\$	0

入力記号

1	+	2	*	3	\$
---	---	---	---	---	----

還元

--

action [0, i] = "s 4"
⇒ (1, 4) をプッシュ

44

解析例

状態	action					goto		
	<i>i</i>	+	*	\$		E	T	F
0	s 4					1	2	3
1		s 5		a				
2		r E→T	s 6	r E→T				
3		r T→F	r T→F	r T→F				
4		r F→i	r F→i	r F→i				
5	s 4					7	3	
6	s 4							8
7		r E→E+T	s 6	r E→E+T				
8		r T→T*F	r T→T*F	r T→T*F				

スタック

記号	状態
\$	0
1	4

入力記号

	+	2	*	3	\$
--	---	---	---	---	----

還元

F

action [4, +] = "r F→i"
⇒ (1, 4) をポップ

45

解析例

状態	action					goto		
	<i>i</i>	+	*	\$		E	T	F
0	s 4					1	2	3
1		s 5		a				
2		r E→T	s 6	r E→T				
3		r T→F	r T→F	r T→F				
4		r F→i	r F→i	r F→i				
5	s 4					7	3	
6	s 4							8
7		r E→E+T	s 6	r E→E+T				
8		r T→T*F	r T→T*F	r T→T*F				

スタック

記号	状態
\$	0

入力記号

	+	2	*	3	\$
--	---	---	---	---	----

還元

F

action [4, +] = "r F→i"
⇒ (1, 4) をポップ
goto [0, F] = "3"
⇒ (F, 3) をプッシュ
コード生成 PUSH 1

46

解析例

状態	action					goto		
	<i>i</i>	+	*	\$		E	T	F
0	s 4					1	2	3
1		s 5		a				
2		r E→T	s 6	r E→T				
3		r T→F	r T→F	r T→F				
4		r F→i	r F→i	r F→i				
5	s 4					7	3	
6	s 4							8
7		r E→E+T	s 6	r E→E+T				
8		r T→T*F	r T→T*F	r T→T*F				

スタック

記号	状態
\$	0
F	3

入力記号

	+	2	*	3	\$
--	---	---	---	---	----

還元

T

action [3, +] = "r T→F"
⇒ (F, 3) をポップ

47

解析例

状態	action					goto		
	<i>i</i>	+	*	\$		E	T	F
0	s 4					1	2	3
1		s 5		a				
2		r E→T	s 6	r E→T				
3		r T→F	r T→F	r T→F				
4		r F→i	r F→i	r F→i				
5	s 4					7	3	
6	s 4							8
7		r E→E+T	s 6	r E→E+T				
8		r T→T*F	r T→T*F	r T→T*F				

スタック

記号	状態
\$	0

入力記号

	+	2	*	3	\$
--	---	---	---	---	----

還元

T

action [3, +] = "r T→F"
⇒ (F, 3) をポップ
goto [0, T] = "2"
⇒ (T, 2) をプッシュ

48

解析例

状態	i	action			goto		
		+	*	\$	E	T	F
0	s4				1	2	3
1	s5			a			
2	rE→T	s6		rE→T			
3	rT→F	rT→F		rT→F			
4	rF→i	rF→i		rF→i			
5	s4				7	3	
6	s4						8
7	rE→E+T	s6		rE→E+T			
8	rT→T*F	rT→T*F		rT→T*F			

スタック

記号	状態
\$	0
T	2

入力記号

	+	2	*	3	\$
--	---	---	---	---	----

還元

E

action [2, +] = "r E→T"
⇒ (T, 2) をポップ

49

解析例

状態	i	action			goto		
		+	*	\$	E	T	F
0	s4				1	2	3
1	s5			a			
2	rE→T	s6		rE→T			
3	rT→F	rT→F		rT→F			
4	rF→i	rF→i		rF→i			
5	s4				7	3	
6	s4						8
7	rE→E+T	s6		rE→E+T			
8	rT→T*F	rT→T*F		rT→T*F			

スタック

記号	状態
\$	0

入力記号

	+	2	*	3	\$
--	---	---	---	---	----

還元

E

action [2, +] = "r E→T"
⇒ (T, 2) をポップ
goto [0, E] = "1"
⇒ (E, 1) をプッシュ

50

解析例

状態	i	action			goto		
		+	*	\$	E	T	F
0	s4				1	2	3
1	s5			a			
2	rE→T	s6		rE→T			
3	rT→F	rT→F		rT→F			
4	rF→i	rF→i		rF→i			
5	s4				7	3	
6	s4						8
7	rE→E+T	s6		rE→E+T			
8	rT→T*F	rT→T*F		rT→T*F			

スタック

記号	状態
\$	0
E	1

入力記号

	+	2	*	3	\$
--	---	---	---	---	----

還元

--

action [1, +] = "s 5"
⇒ (+, 5) をプッシュ

51

解析例

状態	i	action			goto		
		+	*	\$	E	T	F
0	s4				1	2	3
1	s5			a			
2	rE→T	s6		rE→T			
3	rT→F	rT→F		rT→F			
4	rF→i	rF→i		rF→i			
5	s4				7	3	
6	s4						8
7	rE→E+T	s6		rE→E+T			
8	rT→T*F	rT→T*F		rT→T*F			

スタック

記号	状態
\$	0
E	1
+	5

入力記号

		2	*	3	\$
--	--	---	---	---	----

還元

--

action [5, i] = "s 4"
⇒ (2, 4) をプッシュ

52

解析例

状態	i	action			goto		
		+	*	\$	E	T	F
0	s4				1	2	3
1	s5			a			
2	rE→T	s6		rE→T			
3	rT→F	rT→F		rT→F			
4	rF→i	rF→i		rF→i			
5	s4				7	3	
6	s4						8
7	rE→E+T	s6		rE→E+T			
8	rT→T*F	rT→T*F		rT→T*F			

スタック

記号	状態
\$	0
E	1
+	5
2	4

入力記号

			*	3	\$
--	--	--	---	---	----

還元

F

action [4, *] = "r F→i"
⇒ (2, 4) をポップ

53

解析例

状態	i	action			goto		
		+	*	\$	E	T	F
0	s4				1	2	3
1	s5			a			
2	rE→T	s6		rE→T			
3	rT→F	rT→F		rT→F			
4	rF→i	rF→i		rF→i			
5	s4				7	3	
6	s4						8
7	rE→E+T	s6		rE→E+T			
8	rT→T*F	rT→T*F		rT→T*F			

スタック

記号	状態
\$	0
E	1
+	5

入力記号

			*	3	\$
--	--	--	---	---	----

還元

F

action [4, *] = "r F→i"
⇒ (2, 4) をポップ
goto [5, F] = "3"
⇒ (F, 3) をプッシュ
コード生成 PUSHI 2

54

解析例

状態	action					goto		
	<i>i</i>	+	*	\$		E	T	F
0	s 4					1	2	3
1		s 5		a				
2		r E→T	s 6	r E→T				
3		r T→F	r T→F	r T→F				
4		r F→i	r F→i	r F→i				
5	s 4					7	3	
6	s 4							8
7		r E→E+T	s 6	r E→E+T				
8		r T→T*F	r T→T*F	r T→T*F				

スタック

記号	状態
\$	0
E	1
+	5
F	3

入力記号

			*	3	\$
--	--	--	---	---	----

還元

					T
--	--	--	--	--	---

action [3, *] = "r T→F"
⇒ (F, 3) をポップ

55

解析例

状態	action					goto		
	<i>i</i>	+	*	\$		E	T	F
0	s 4					1	2	3
1		s 5		a				
2		r E→T	s 6	r E→T				
3		r T→F	r T→F	r T→F				
4		r F→i	r F→i	r F→i				
5	s 4					7	3	
6	s 4							8
7		r E→E+T	s 6	r E→E+T				
8		r T→T*F	r T→T*F	r T→T*F				

スタック

記号	状態
\$	0
E	1
+	5

入力記号

			*	3	\$
--	--	--	---	---	----

還元

					T
--	--	--	--	--	---

action [3, *] = "r T→F"
⇒ (F, 3) をポップ
goto [5, T] = "7"
⇒ (T, 7) をプッシュ

56

解析例

状態	action					goto		
	<i>i</i>	+	*	\$		E	T	F
0	s 4					1	2	3
1		s 5		a				
2		r E→T	s 6	r E→T				
3		r T→F	r T→F	r T→F				
4		r F→i	r F→i	r F→i				
5	s 4					7	3	
6	s 4							8
7		r E→E+T	s 6	r E→E+T				
8		r T→T*F	r T→T*F	r T→T*F				

スタック

記号	状態
\$	0
E	1
+	5
T	7

入力記号

			*	3	\$
--	--	--	---	---	----

還元

--	--	--	--	--	--

action [7, *] = "s 6"
⇒ (*, 6) をプッシュ

57

解析例

状態	action					goto		
	<i>i</i>	+	*	\$		E	T	F
0	s 4					1	2	3
1		s 5		a				
2		r E→T	s 6	r E→T				
3		r T→F	r T→F	r T→F				
4		r F→i	r F→i	r F→i				
5	s 4					7	3	
6	s 4							8
7		r E→E+T	s 6	r E→E+T				
8		r T→T*F	r T→T*F	r T→T*F				

スタック

記号	状態
\$	0
E	1
+	5
T	7
*	6

入力記号

				3	\$
--	--	--	--	---	----

還元

--	--	--	--	--	--

action [6, i] = "s 4"
⇒ (3, 4) をプッシュ

58

解析例

状態	action					goto		
	<i>i</i>	+	*	\$		E	T	F
0	s 4					1	2	3
1		s 5		a				
2		r E→T	s 6	r E→T				
3		r T→F	r T→F	r T→F				
4		r F→i	r F→i	r F→i				
5	s 4					7	3	
6	s 4							8
7		r E→E+T	s 6	r E→E+T				
8		r T→T*F	r T→T*F	r T→T*F				

スタック

記号	状態
\$	0
E	1
+	5
T	7
*	6
3	4

入力記号

					\$
--	--	--	--	--	----

還元

					F
--	--	--	--	--	---

action [4, \$] = "r F→i"
⇒ (3, 4) をポップ

59

解析例

状態	action					goto		
	<i>i</i>	+	*	\$		E	T	F
0	s 4					1	2	3
1		s 5		a				
2		r E→T	s 6	r E→T				
3		r T→F	r T→F	r T→F				
4		r F→i	r F→i	r F→i				
5	s 4					7	3	
6	s 4							8
7		r E→E+T	s 6	r E→E+T				
8		r T→T*F	r T→T*F	r T→T*F				

スタック

記号	状態
\$	0
E	1
+	5
T	7
*	6

入力記号

					\$
--	--	--	--	--	----

還元

					F
--	--	--	--	--	---

action [4, \$] = "r F→i"
⇒ (3, 4) をポップ
goto [6, F] = "8"
⇒ (F, 8) をプッシュ
コード生成 PUSHI 3

60

解析例

状態	action					goto		
	i	+	*	\$	E	T	F	
0	s4				1	2	3	
1		s5		a				
2		rE→T	s6	rE→T				
3		rT→F	rT→F	rT→F				
4		rF→i	rF→i	rF→i				
5	s4				7	3		
6	s4						8	
7		rE→E+T	s6	rE→E+T				
8		rT→T*F	rT→T*F	rT→T*F				

スタック

記号	状態
\$	0
E	1
+	5
T	7
*	6
F	8

入力記号

					\$
--	--	--	--	--	----

還元

					T
--	--	--	--	--	---

action [8, \$] = “r T→T*F”
⇒ (F, 8) (*, 6) (T, 7) をポップ

61

解析例

状態	action					goto		
	i	+	*	\$	E	T	F	
0	s4				1	2	3	
1		s5		a				
2		rE→T	s6	rE→T				
3		rT→F	rT→F	rT→F				
4		rF→i	rF→i	rF→i				
5	s4				7	3		
6	s4						8	
7		rE→E+T	s6	rE→E+T				
8		rT→T*F	rT→T*F	rT→T*F				

スタック

記号	状態
\$	0
E	1
+	5

入力記号

					\$
--	--	--	--	--	----

還元

					T
--	--	--	--	--	---

action [8, \$] = “r T→T*F”
⇒ (F, 8) (*, 6) (T, 7) をポップ
goto [5, T] = “7”
⇒ (T, 7) をプッシュ
コード生成 MUL

62

解析例

状態	action					goto		
	i	+	*	\$	E	T	F	
0	s4				1	2	3	
1		s5		a				
2		rE→T	s6	rE→T				
3		rT→F	rT→F	rT→F				
4		rF→i	rF→i	rF→i				
5	s4				7	3		
6	s4						8	
7		rE→E+T	s6	rE→E+T				
8		rT→T*F	rT→T*F	rT→T*F				

スタック

記号	状態
\$	0
E	1
+	5
T	7

入力記号

					\$
--	--	--	--	--	----

還元

					E
--	--	--	--	--	---

action [7, \$] = “r E→E+T”
⇒ (T, 7) (+, 5) (E, 1) をポップ

63

解析例

状態	action					goto		
	i	+	*	\$	E	T	F	
0	s4				1	2	3	
1		s5		a				
2		rE→T	s6	rE→T				
3		rT→F	rT→F	rT→F				
4		rF→i	rF→i	rF→i				
5	s4				7	3		
6	s4						8	
7		rE→E+T	s6	rE→E+T				
8		rT→T*F	rT→T*F	rT→T*F				

スタック

記号	状態
\$	0

入力記号

					\$
--	--	--	--	--	----

還元

					E
--	--	--	--	--	---

action [7, \$] = “r E→E+T”
⇒ (T, 7) (+, 5) (E, 1) をポップ
goto [0, E] = “1”
⇒ (E, 1) をプッシュ
コード生成 ADD

64

解析例

状態	action					goto		
	i	+	*	\$	E	T	F	
0	s4				1	2	3	
1		s5		a				
2		rE→T	s6	rE→T				
3		rT→F	rT→F	rT→F				
4		rF→i	rF→i	rF→i				
5	s4				7	3		
6	s4						8	
7		rE→E+T	s6	rE→E+T				
8		rT→T*F	rT→T*F	rT→T*F				

スタック

記号	状態
\$	0
E	1

入力記号

					\$
--	--	--	--	--	----

還元

--	--	--	--	--	--

action [1, \$] = “a”
⇒ 解析完了

65

スタック	入力列	参照	操作	出力
\$0	1 + 2 * 3 \$	a[0,i]	移動 4	
\$0 14	+ 2 * 3 \$	a[4,+], g[0,F]	還元 F→i, 3	PUSHI 1
\$0 F3	+ 2 * 3 \$	a[3,+], g[0,T]	還元 T→F, 2	
\$0 T2	+ 2 * 3 \$	a[2,+], g[0,E]	還元 E→T, 1	
\$0 E1	+ 2 * 3 \$	a[1,+]	移動 5	
\$0 E1 +5	2 * 3 \$	a[5,i]	移動 4	
\$0 E1 +5 24	* 3 \$	a[4,*], g[5,F]	還元 F→i, 3	PUSHI 2
\$0 E1 +5 F3	* 3 \$	a[3,*], g[5,T]	還元 T→F, 7	
\$0 E1 +5 T7	* 3 \$	a[7,*]	移動 6	
\$0 E1 +5 T7 *6	3 \$	a[6,i]	移動 4	
\$0 E1 +5 T7 *6 34	\$	a[4,\$], g[6,F]	還元 F→i, 8	PUSHI 3
\$0 E1 +5 T7 *6 F8	\$	a[8,\$], g[5,T]	還元 T→T*F, 7	MUL
\$0 E1 +5 T7	\$	a[7,\$], g[0,E]	還元 E→E+T, 1	ADD
\$0 E1	\$	a[1,\$]	受理	

66

LR(k) 文法	
<ul style="list-style-type: none"> ■ LR(k) 文法 <ul style="list-style-type: none"> - k 個のトークンを先読みすれば解析可能 ■ LR(1) 文法 <ul style="list-style-type: none"> - 直後のトークン1個を先読みすれば解析可能 ■ LR(0) 文法 <ul style="list-style-type: none"> - 先読み無しで解析可能 	

67

LR構文解析の作成	
<ul style="list-style-type: none"> ■ LR構文解析 <ul style="list-style-type: none"> - スタックと解析表で解析 ⇒ 解析表の作成が必要 <p>しかし解析表の作成は人力では不可能</p> <p>表が大きくなり過ぎる (終端記号数, 非終端記号数に対して サイズが指数的に)</p>	

68

LR構文解析のクラス				
クラス	特徴	解析表 サイズ	受理 文法	解析表の作成
LR(0)	先読み無し	↑ 小 ↓ 大 ↑ 狭 ↓ 広	↑ 狭 ↓ 広	人力で可能
SLR(1) simple LR(1)	必要時に先読み			人力で可能
LALR(1) look ahead LR(1)	常に先読み 状態を一部統合			コンパイラコン パイラで作成
LR(1)	常に先読み 状態を完全分離			作成は難しい
コンパイラコンパイラ: コンパイラを自動生成するプログラム				

69

LR(0) 構文解析の 解析表作成	
<ol style="list-style-type: none"> 1. 各非終端記号のFollow集合を求める 2. LR(0) 項を求める 3. 閉包(=状態)を求める 4. 各状態からの遷移を求める 	

70

LR(0)項 (LR(0) item)	
<ul style="list-style-type: none"> ■ LR(0)項 <ul style="list-style-type: none"> - 生成規則の右辺に解析位置(・)を加えたもの <p>例: $A \rightarrow BCD$</p> <p>$[A \rightarrow \cdot BCD][A \rightarrow B \cdot CD][A \rightarrow BC \cdot D][A \rightarrow BCD \cdot]$</p> <p>解析前 Bまで解析 Cまで解析 全て解析</p> <p>例: $A \rightarrow \epsilon$</p> <p>$[A \rightarrow \cdot]$</p>	

71

LR(0)項	
<p>例: $\{ \\$ \rightarrow E\\$, E \rightarrow E+T , E \rightarrow T , T \rightarrow T * F , T \rightarrow F , F \rightarrow i \}$</p> <p>項 $[\\$ \rightarrow \cdot E\\$]$ $[\\$ \rightarrow \cdot E\\$]$</p> <p style="text-align: center;"> $\begin{matrix} \boxed{E \rightarrow \cdot E+T} & \boxed{F \rightarrow \cdot T} \\ \cdot E+T\\$ & \cdot T\\$ \end{matrix}$ </p> <p>$E\\$ 解析前 = $E+T$ 解析前 $[\\$ \rightarrow \cdot E\\$]$</p> <p style="text-align: center;">= T 解析前 = $[E \rightarrow \cdot E+T]$</p> <p style="text-align: center;">= $[E \rightarrow \cdot T]$</p> <p style="text-align: center;">= $[T \rightarrow \cdot T * F]$</p> <p style="text-align: center;">= $[T \rightarrow \cdot F]$</p> <p style="text-align: center;">= $[F \rightarrow \cdot i]$</p> <p>同じ状態として扱う必要あり 閉包 (closure)</p>	

72

LR(0)項の閉包 (closure)

- LR(0)項集合 I の閉包 closure (I)
 - $i \in I \Rightarrow i \in \text{closure}(I)$
 - I の項は closure (I) に含まれる
 - $[A \rightarrow u \cdot Bv] \in \text{closure}(I) \wedge B \rightarrow w \in P \Rightarrow [B \rightarrow \cdot w] \in \text{closure}(I)$
 - $[A \rightarrow u \cdot Bv]$ が closure (I) に含まれ $B \rightarrow w$ が生成規則 P に含まれるならば $[B \rightarrow \cdot w]$ は closure (I) に含まれる

73

LR(0)閉包の例

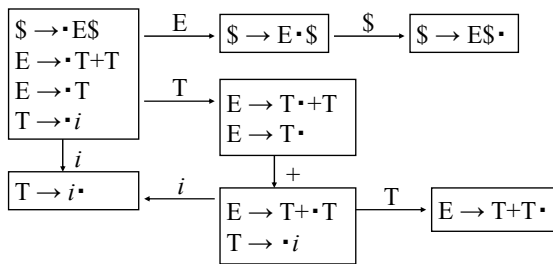
- 例 : $\$ \rightarrow E\$$, $E \rightarrow E+T$, $E \rightarrow T$, $T \rightarrow T*F$, $T \rightarrow F$, $F \rightarrow i$
- $I = [\$ \rightarrow \cdot E\$]$ のとき
- $\text{closure}(I) = \{$
- $[\$ \rightarrow \cdot E\$]$ I 自身
 - $[E \rightarrow \cdot E+T]$ E から生成
 - $[E \rightarrow \cdot T]$ E から生成
 - $[T \rightarrow \cdot T*F]$ T から生成
 - $[T \rightarrow \cdot F]$ T から生成
 - $[F \rightarrow \cdot i]$ F から生成
- $\}$

74

LR(0)閉包の例

- 例 : $\$ \rightarrow E\$$, $E \rightarrow T+T$, $E \rightarrow T$, $T \rightarrow i$

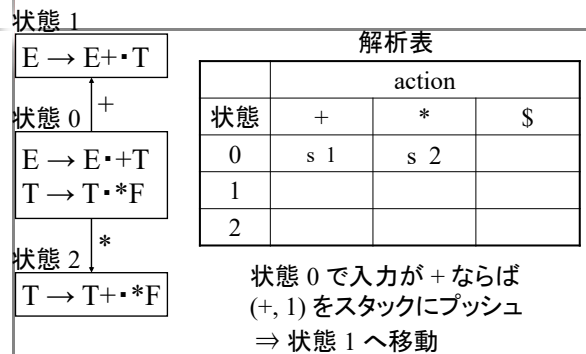
Follow (E) = { $\$$ }, Follow (T) = { $\$, +$ }



各閉包を1つの状態として状態遷移を求める

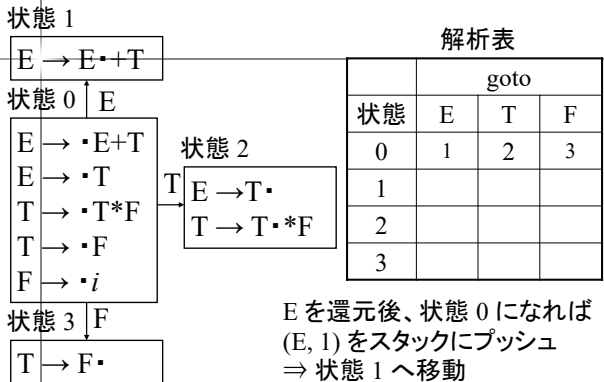
75

構文解析表の作成 (移動)



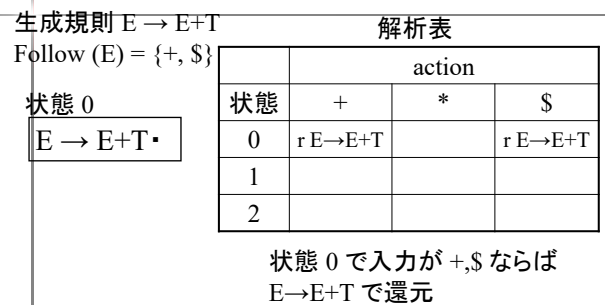
76

構文解析表の作成 (還元)

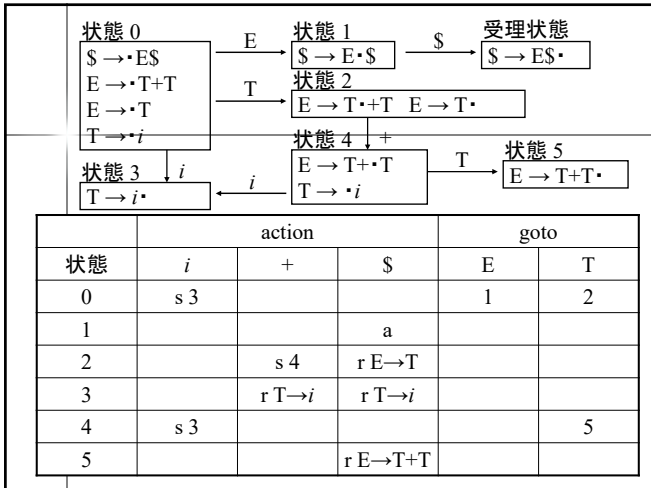


77

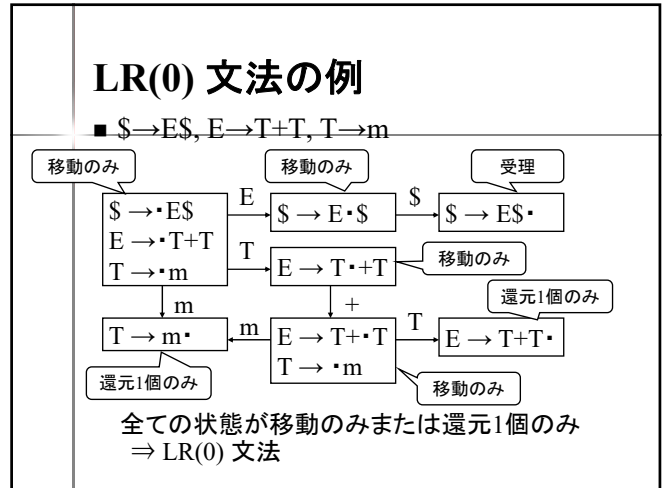
構文解析表の作成 (還元)



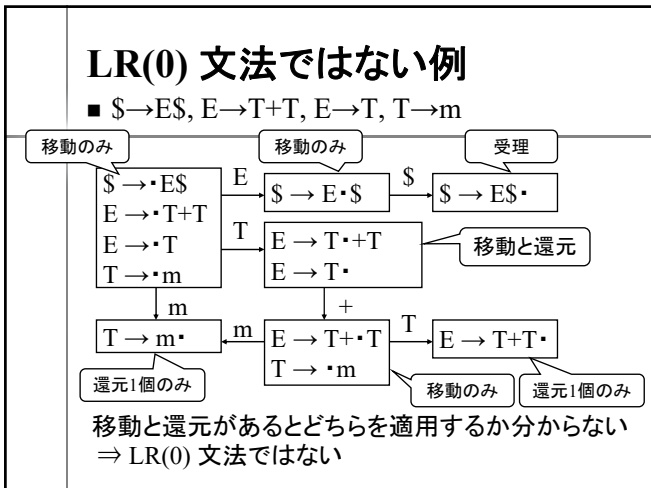
78



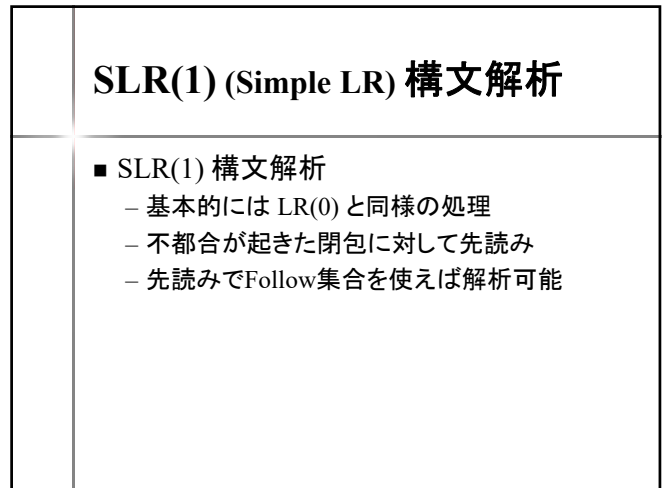
79



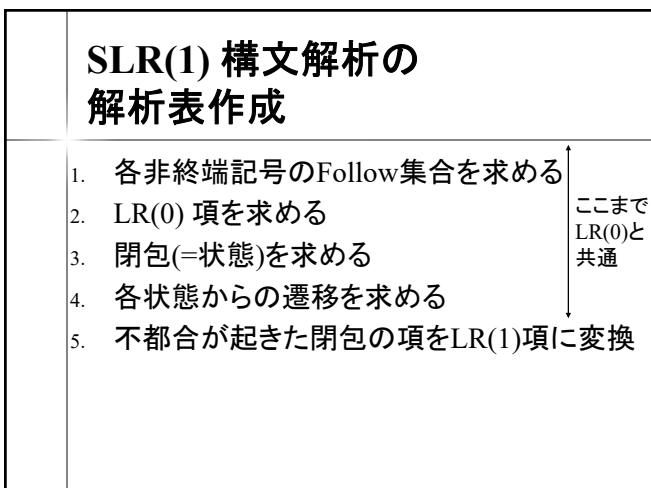
80



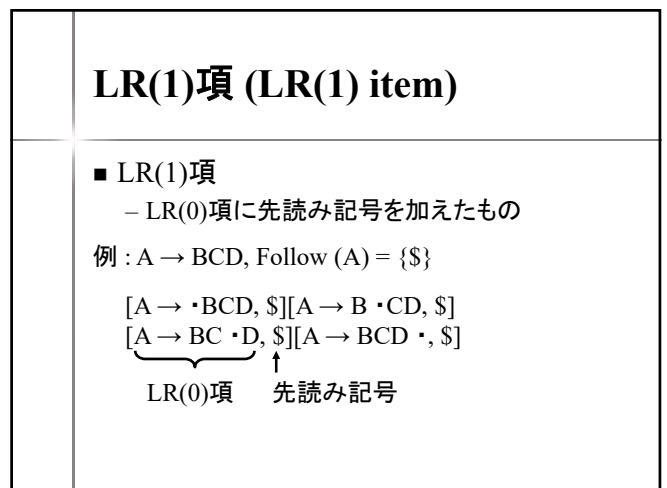
81



82



83



84

LR(1)項の閉包 (closure)

- LR(1)項集合 I の閉包 $\text{closure}(I)$
 - $i \in I \Rightarrow i \in \text{closure}(I)$
 - I の項は $\text{closure}(I)$ に含まれる
 - $[A \rightarrow u \cdot Bv, a] \in \text{closure}(I) \wedge B \rightarrow w \in P \Rightarrow [B \rightarrow \cdot w, \text{First}(va)] \in \text{closure}(I)$
 - $[A \rightarrow u \cdot Bv, a]$ が $\text{closure}(I)$ に含まれ $B \rightarrow w$ が生成規則 P に含まれるならば $[B \rightarrow \cdot w, \text{First}(va)]$ は $\text{closure}(I)$ に含まれる

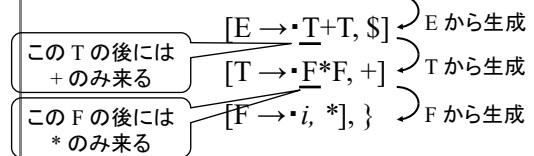
85

LR(1)閉包の例

- 例: $\$ \rightarrow E\$$, $E \rightarrow T+T$, $T \rightarrow F*F$, $F \rightarrow i$
 $\text{Follow}(E) = \{\$, \}$, $\text{Follow}(T) = \{+, \$\}$, $\text{Follow}(F) = \{+, *, \$\}$

$I = [\$ \rightarrow \cdot E\$]$ のとき

$\text{closure}(I) = \{ [\$ \rightarrow \cdot E\$], [E \rightarrow \cdot T+T], [T \rightarrow \cdot F*F], [F \rightarrow \cdot i] \}$



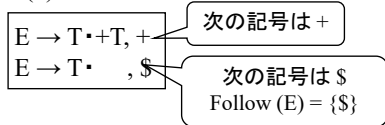
86

LR(1)閉包の例

- 例: $\$ \rightarrow E\$$, $E \rightarrow T+T$, $E \rightarrow T$, $T \rightarrow i$

$\text{Follow}(E) = \{\$, \}$, $\text{Follow}(T) = \{+, \$\}$

LR(1)閉包



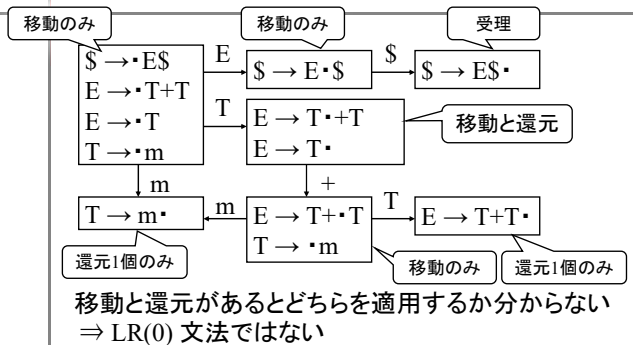
次の記号が $+$ $\Rightarrow E \rightarrow T+T$ の解析
 次の記号が $\$$ $\Rightarrow E \rightarrow T$ の解析

87

SLR(1) 文法の例

① LR(0)と同様に処理

- $\$ \rightarrow E\$$, $E \rightarrow T+T$, $E \rightarrow T$, $T \rightarrow m$

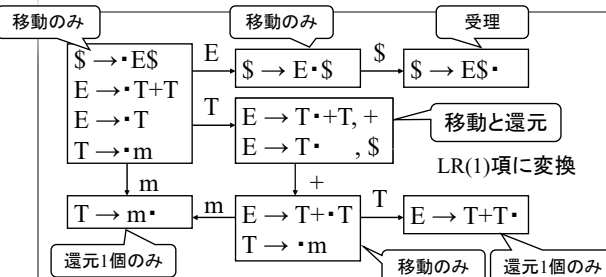


88

SLR(1) 文法の例

② 不都合な項を LA(1)に変換

- $\$ \rightarrow E\$$, $E \rightarrow T+T$, $E \rightarrow T$, $T \rightarrow m$

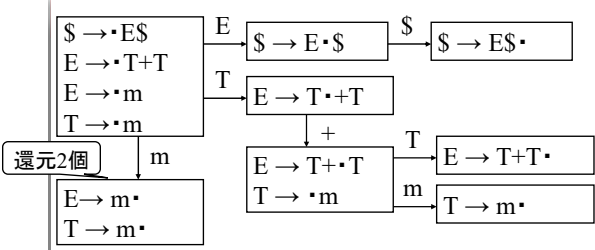


89

SLR(1) 文法ではない例

- $\$ \rightarrow E\$$, $E \rightarrow T+T$, $E \rightarrow m$, $T \rightarrow m$

$\text{Follow}(E) = \{\$, \}$, $\text{Follow}(T) = \{+, \$\}$



90

SLR(1) 文法ではない例

■ $S \rightarrow ES, E \rightarrow T+T, E \rightarrow m, T \rightarrow m$
 $\text{Follow}(E) = \{\$, \}, \text{Follow}(T) = \{\$, +\}$

$S \rightarrow \cdot ES$
 $E \rightarrow \cdot T+T$
 $E \rightarrow \cdot m$
 $T \rightarrow \cdot m$

$E \rightarrow E \cdot \$$ $\$ \rightarrow \$ \cdot$
 $E \rightarrow T \cdot +T$

還元2個 m

$E \rightarrow m \cdot, \$$
 $T \rightarrow m \cdot, \$+$

$\text{Follow}(E) = \{\$\}$
 $E \rightarrow \text{Follow}(T) = \{\$, +\}$
 $T \rightarrow$ 次の記号が $\$$ の場合
 どちらの還元か判定不可能

不都合発生部分をLR(1)項に変換しても解析不可能
 \Rightarrow SLR(1) 文法ではない

91

LALR(1) (Look Ahead LR) 構文解析

■ LALR(1) 構文解析

- まずLR(0)と同様の処理
- その後全ての閉包に対して先読み
- 先読みでFollow集合を使えば解析可能

92

LALR(1) 構文解析の解析表作成

- 各非終端記号のFollow集合を求める
- LR(0) 項を求める
- 閉包(=状態)を求める
- 各状態からの遷移を求める
- 初期状態から順に全ての閉包の項をLR(1)項に変換

ここまでLR(0)と共通

93

LALR(1) 文法の例 ① LR(0)と同様に処理

■ $S \rightarrow ES, E \rightarrow T+T, E \rightarrow m, T \rightarrow m$
 $\text{Follow}(E) = \{\$, \}, \text{Follow}(T) = \{+, \$\}$

$S \rightarrow \cdot ES$
 $E \rightarrow \cdot T+T$
 $E \rightarrow \cdot m$
 $T \rightarrow \cdot m$

$E \rightarrow E \cdot \$$ $\$ \rightarrow \$ \cdot, \epsilon$
 $E \rightarrow T \cdot +T$
 $E \rightarrow T+ \cdot T$ $T \rightarrow T \cdot$
 $T \rightarrow \cdot m$ $T \rightarrow m \cdot$

還元2個 m

$E \rightarrow m \cdot, \$$
 $T \rightarrow m \cdot, \$+$

不都合発生部分をLR(1)に変換しても解析不可能
 \Rightarrow SLR(1) 文法ではない

94

LALR(1) 文法の例 ② 全ての項をLA(1)に変換

■ $S \rightarrow ES, E \rightarrow T+T, E \rightarrow m, T \rightarrow m$
 $\text{Follow}(E) = \{\$, \}, \text{Follow}(T) = \{+, \$\}$

$S \rightarrow \cdot ES, \epsilon$
 $E \rightarrow \cdot T+T, \$$
 $E \rightarrow \cdot m, \$$
 $T \rightarrow \cdot m, +$

$E \rightarrow E \cdot \$, \epsilon$ $\$ \rightarrow \$ \cdot, \epsilon$
 $E \rightarrow T \cdot +T, \$$

還元2個 m

$E \rightarrow m \cdot, \$$
 $T \rightarrow m \cdot, +$

$E \rightarrow T+ \cdot T, \$$ $T \rightarrow T \cdot, \$$
 $T \rightarrow \cdot m, \$$ $T \rightarrow m \cdot, \$$

次の記号が $\$$ か $+$ かでどちらの還元か判定可能
 \Rightarrow LALR(1) 文法

95

LALR(1) 文法ではない例

■ $S \rightarrow ES, S \rightarrow +XS, E \rightarrow T=, X \rightarrow E=, X \rightarrow T, E \rightarrow m, T \rightarrow m$
 $\text{Follow}(E) = \{\$, =\}, \text{Follow}(X) = \{\$, \}, \text{Follow}(T) = \{\$, =\}$

$S \rightarrow \cdot ES$
 $S \rightarrow \cdot +XS$
 $E \rightarrow \cdot T=$
 $E \rightarrow \cdot m$
 $T \rightarrow \cdot m$

$S \rightarrow E \cdot \$$ $\$ \rightarrow \$ \cdot$
 $S \rightarrow + \cdot XS$ $X \rightarrow +X \cdot \$$ $\$ \rightarrow \$ \cdot, \epsilon$
 $X \rightarrow \cdot E=$ $X \rightarrow E \cdot =$
 $X \rightarrow \cdot T$ $X \rightarrow T \cdot$

還元2個 m

$E \rightarrow T \cdot =$
 $E \rightarrow m \cdot, \$=$
 $T \rightarrow m \cdot, \$=$

不都合発生部分をLA(1)に変換しても解析不可能
 \Rightarrow SLR(1) 文法ではない

96

LALR(1) 文法ではない例

■ $S \rightarrow ES, S \rightarrow +XS, E \rightarrow T=, X \rightarrow E=, X \rightarrow T, E \rightarrow m, T \rightarrow m$
 $\text{Follow}(E) = \{\$, =\}, \text{Follow}(X) = \{\$\}, \text{Follow}(T) = \{\$, =\}$

全ての閉包をLR(1)項に変換しても解析不可能
 \Rightarrow LALR(1) 文法ではない

97

LR(1) 構文解析

■ LR(1) 構文解析
 - 最初からLR(1)項を使って閉包を求める
 - 先読みでFollow集合を使えば解析可能

98

LR(1) 構文解析の解析表作成

1. 各非終端記号のFollow集合を求める
2. LR(1) 項を求める
3. 閉包(=状態)を求める
4. 各状態からの遷移を求める

99

LR(1) 文法の例

■ $S \rightarrow ES, S \rightarrow +XS, E \rightarrow T=, X \rightarrow E=, X \rightarrow T, E \rightarrow m, T \rightarrow m$
 $\text{Follow}(E) = \{\$, =\}, \text{Follow}(X) = \{\$\}, \text{Follow}(T) = \{\$, =\}$

どちらの閉包も次の記号でどちらの還元か判定可能
 \Rightarrow LR(1)文法

100

LR(1)文法ではない例

■ $S \rightarrow ES, E \rightarrow E+a, E \rightarrow T+T, E \rightarrow m, T \rightarrow m$
 $\text{Follow}(E) = \{\$, +\}, \text{Follow}(T) = \{\$, +\}$

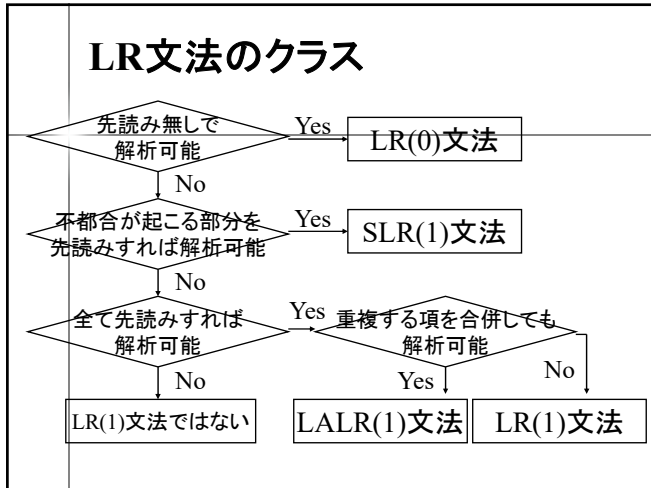
1個先読みしても解析不可能
 \Rightarrow LR(1)文法ではない

101

LR構文解析のクラス

クラス	先読み	状態	解析表 サイズ	受理 文法	その他の特徴
LR(0)	無し	LR(0)項	小	狭	
SLR(1)	有り (必要時のみ)	LR(0)項 \Rightarrow LR(1)項 (必要部分のみ)			
LALR(1)	有り	LR(0)項 \Rightarrow LR(1)項 (全体)	大	広	コンパイラコンパイラで作成
LR(1)	有り	LR(1)項			

102



103

構文解析の種類

下降型解析 (top-down parsing)	再帰下降解析 (recursive descent parsing)
	LL解析 (Left to right scan & Left most derivation)
上昇型解析 (bottom-up parsing)	演算子順位構文解析 (operator precedence parsing)
	LR解析 (Left to right scan & Right most derivation)

104

構文解析の長所と短所

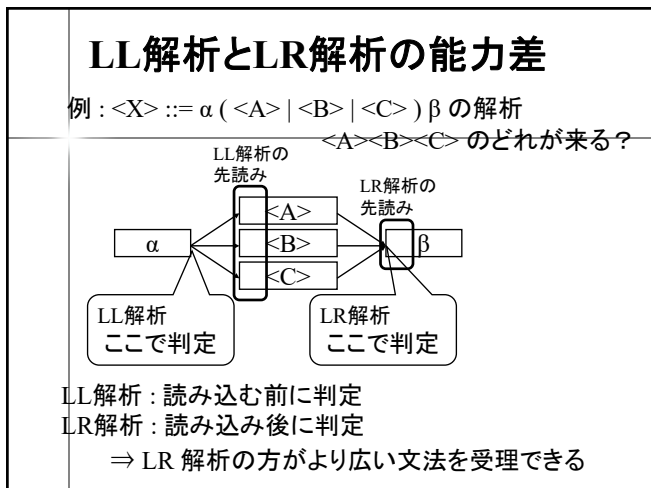
文法	解析表と文法の対応	解析表サイズ	適用可能文法範囲	解析効率	その他の長所
再帰下降	高	小	狭	高	構文エラー発見時のエラーメッセージを作り易い コード生成部を埋め込み易い
LL					
演算子順位	式:高 全体:低	式:小 全体:大	狭	式:最高 全体:低	
LR	低	大	広	低	構文エラー発見が遅い

105

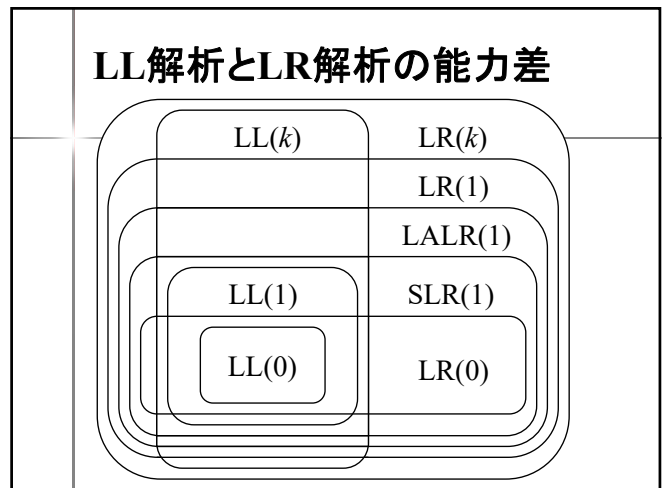
LL解析とLR解析の解析表

- LL解析
 - 文法から直接解析表を作成可能
 - サイズが小さい $LL(k) : |N| \times |T|^k$
 - ⇒ 解析表の作成は容易
- LR解析
 - 解析表の導出作業が必要
 - サイズが大きい $LR(k) : 2^{|P|} \times (|N| + |T|) \times |T|^k$
 - ⇒ 解析表の作成は困難

106



107



108

オンライン試験

- 試験日：7月26日(水)
- 試験時間：60分
- 試験範囲：第1～14回
- 配点：70点満点
- 持ち込み：全て可
 - －ただし外部との通信は禁止