

コンパイラ

第11回 上昇型構文解析(1)

<http://www.info.kindai.ac.jp/compiler>

E館3階E-331 内線5459

takasi-i@info.kindai.ac.jp

コンパイラの構造

- 字句解析系
- 構文解析系
- 制約検査系
- 中間コード生成系
- 最適化系
- 目的コード生成系

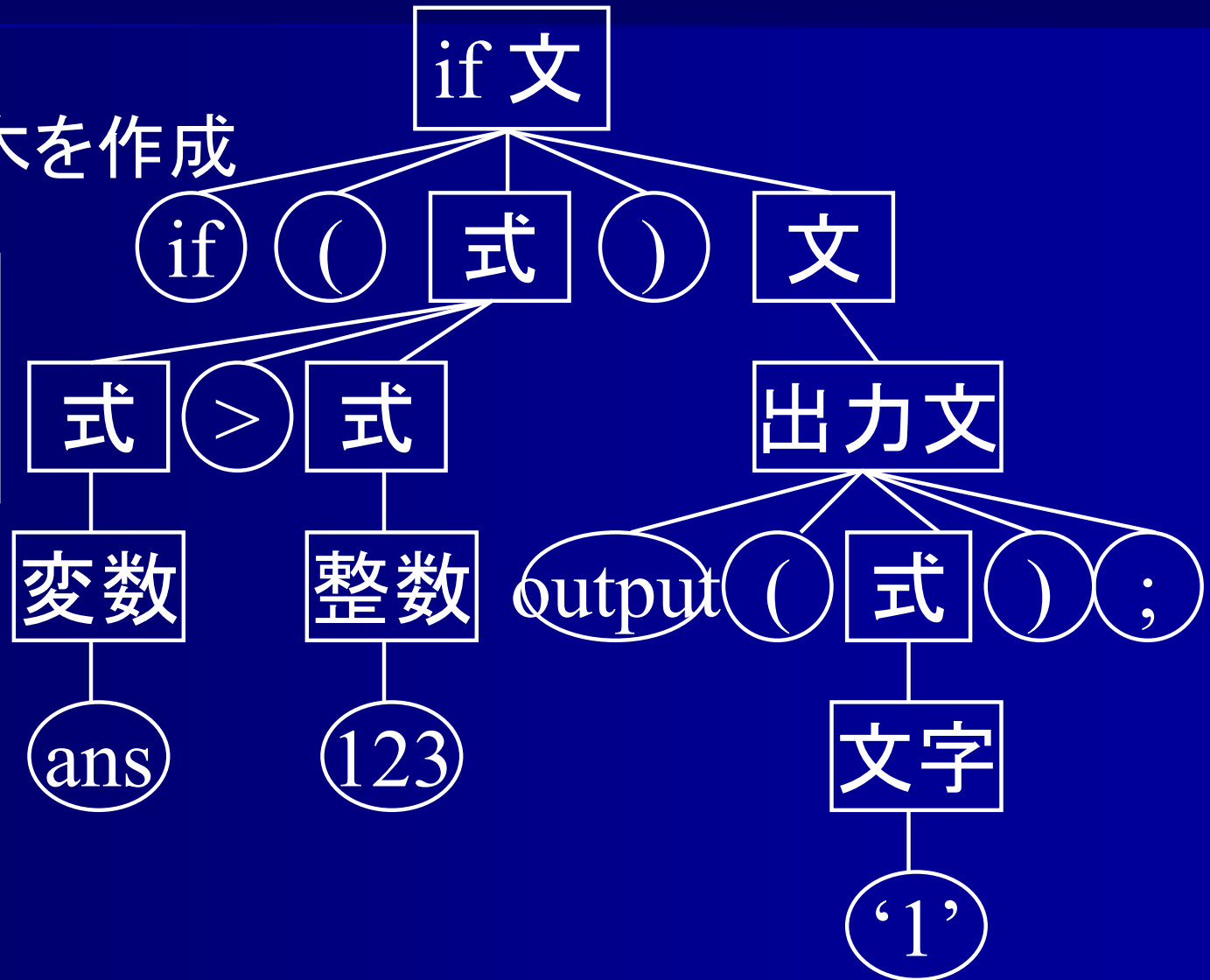
構文解析系

(syntax analyzer, parser)

■ 構文解析系

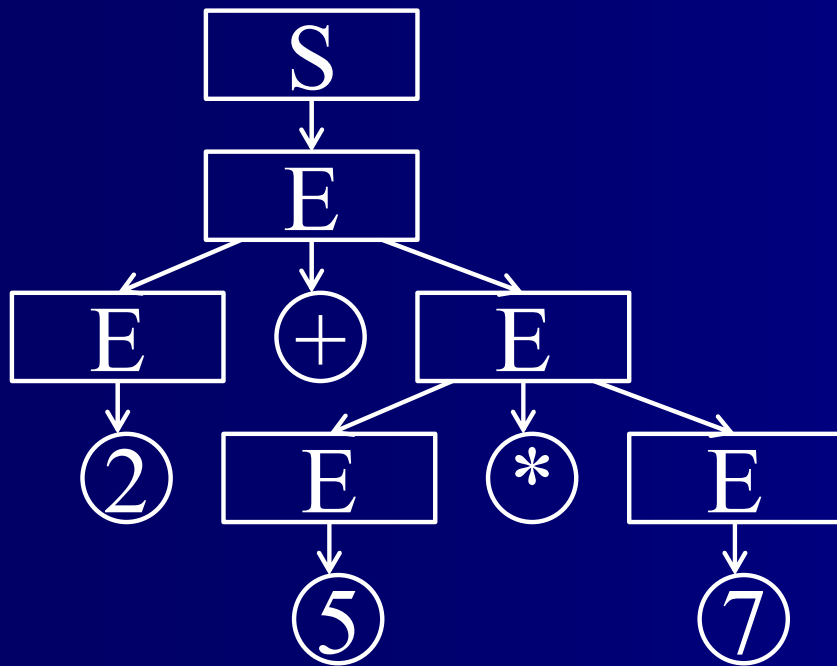
– 構文解析木を作成

```
if (ans > 123 )  
output ('1');
```

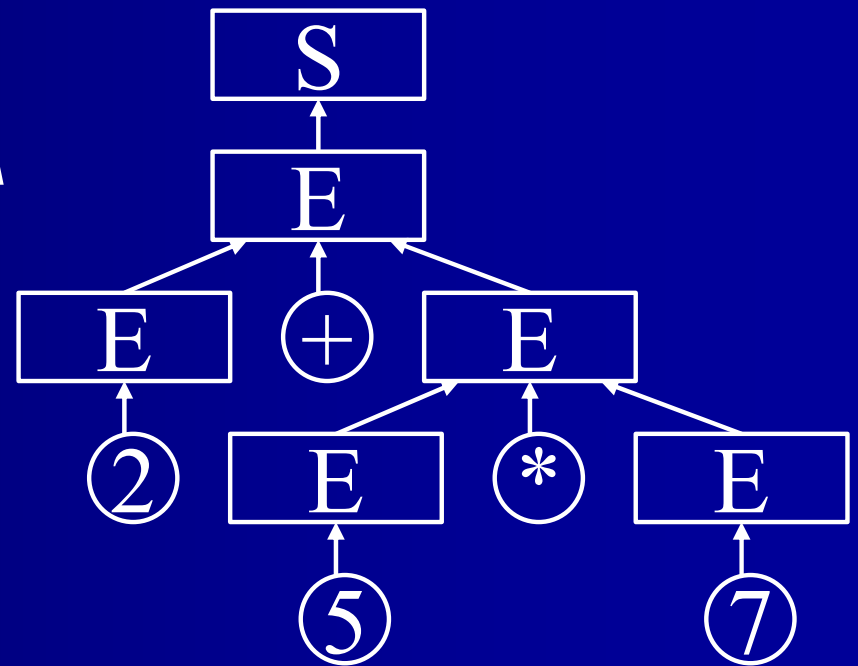


構文解析

- 文法 $G = \{N, T, S, P\}$ が与えられたとき、 $\omega \in T^*$ に対して $S \Rightarrow \omega$ であるか判定、その導出木を得る



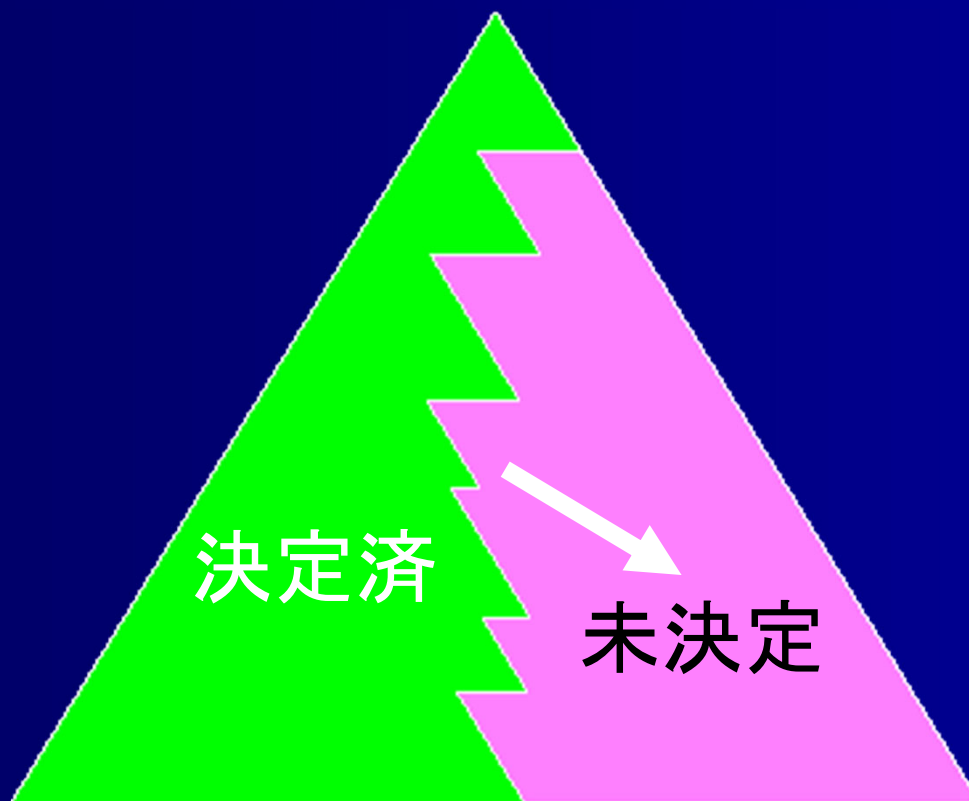
下降型解析



上昇型解析

下降型解析(top-down parsing)

構文解析木



入力記号列

既読

未読

下降型解析の例

$\langle \text{namelist} \rangle ::= \langle \text{name} \rangle \mid \langle \text{name} \rangle \text{“,”} \langle \text{namelist} \rangle$

$\langle \text{name} \rangle ::= \text{“a”} \mid \text{“b”} \mid \text{“c”}$

a, b, c $\langle \text{namelist} \rangle$

$\rightarrow \langle \text{name} \rangle \text{“,”} \langle \text{namelist} \rangle$

$\rightarrow \text{“a”} \text{“,”} \langle \text{namelist} \rangle$

$\rightarrow \text{“a”} \text{“,”} \langle \text{name} \rangle \text{“,”} \langle \text{namelist} \rangle$

$\rightarrow \text{“a”} \text{“,”} \text{“b”} \text{“,”} \langle \text{namelist} \rangle$

$\rightarrow \text{“a”} \text{“,”} \text{“b”} \text{“,”} \langle \text{name} \rangle$

$\rightarrow \text{“a”} \text{“,”} \text{“b”} \text{“,”} \text{“c”}$

$\langle \text{namelist} \rangle \Rightarrow \text{a,b,c}$

上昇型解析(bottom-up parsing)

構文解析木



入力記号列

既読

未読

上昇型解析の例

$\langle \text{namelist} \rangle ::= \langle \text{name} \rangle \mid \langle \text{namelist} \rangle \text{“,”} \langle \text{name} \rangle$

$\langle \text{name} \rangle ::= \text{“a”} \mid \text{“b”} \mid \text{“c”}$

a, b, c “a” “,” “b” “,” “c”

→ $\langle \text{name} \rangle \text{“,”} \text{“b”} \text{“,”} \text{“c”}$

→ $\langle \text{namelist} \rangle \text{“,”} \text{“b”} \text{“,”} \text{“c”}$

→ $\langle \text{namelist} \rangle \text{“,”} \langle \text{name} \rangle \text{“,”} \text{“c”}$

→ $\langle \text{namelist} \rangle \text{“,”} \text{“c”}$

→ $\langle \text{namelist} \rangle \text{“,”} \langle \text{name} \rangle$

→ $\langle \text{namelist} \rangle$

$\langle \text{namelist} \rangle \Rightarrow \text{a,b,c}$

構文解析の種類

情報システムプロジェクトIの構文解析

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)

再帰下降構文解析の欠点

- 受理できる言語の範囲が狭い
 - 受理できる原始言語の文法に制限がある
- 左再帰性を除去すると構文木が変わる
 - 演算子の結合性の情報が消失
 - ⇒ 左再帰性の除去時に情報を記録
- 再帰が必要
 - 記述言語が再帰可能な場合のみ使用可能
 - ⇒ スタックを用いる

LL解析

- Left to right scan & Left most derivation 解析
 - 下降型解析
 - スタックと解析表を用いて解析
 - 再帰下降型解析と本質的に同じ

LL解析

ファイル末

入力記号列

スタック

3 * (6 + 4) \$

開始記号

解析表

\$
E

N \ T	i (整数)	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ϵ	ϵ
T	FT'			FT'		
T'		ϵ	*FT'		ϵ	ϵ
F	i			(E)		

解析表

■ 非終端記号 × 終端記号の表

N \ T	i (整数)	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ϵ	ϵ
T	FT'			FT'		
T'		ϵ	*FT'		ϵ	ϵ
F	i			(E)		

項目 $M[T, i]$: 生成規則 $T \rightarrow FT'$

意味 : 非終端記号 T の解析時に記号 i を読めば
次は FT' を解析する

空欄は構文解析エラー

生成規則→解析表

$E \rightarrow TE'$ $\text{First}(TE') = \{i, \text{"("}\}$

$E' \rightarrow +TE' \mid \varepsilon$ $\text{First}(+TE') = \{\text{"+"}\}$

$T \rightarrow FT'$ $\text{First}(FT') = \{i, \text{"("}\}$

$T' \rightarrow *FT' \mid \varepsilon$ $\text{First}(*FT') = \{\text{"*"}\}$

$F \rightarrow i \mid (E)$ $\text{First}(i) = \{i\}, \text{First}((E)) = \{\text{"("}\}$

$N \setminus T$	i (整数)	$+$	$*$	$($	$)$	$\$$
E	TE'			TE'		
E'		$+TE'$			ε	ε
T	FT'			FT'		
T'		ε	$*FT'$		ε	ε
F	i			(E)		

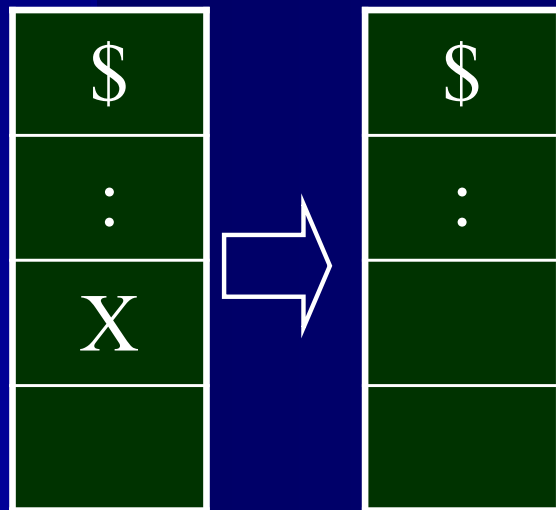
LL解析の手順

X : スタックトップ $a \in T$: 現在の入力記号

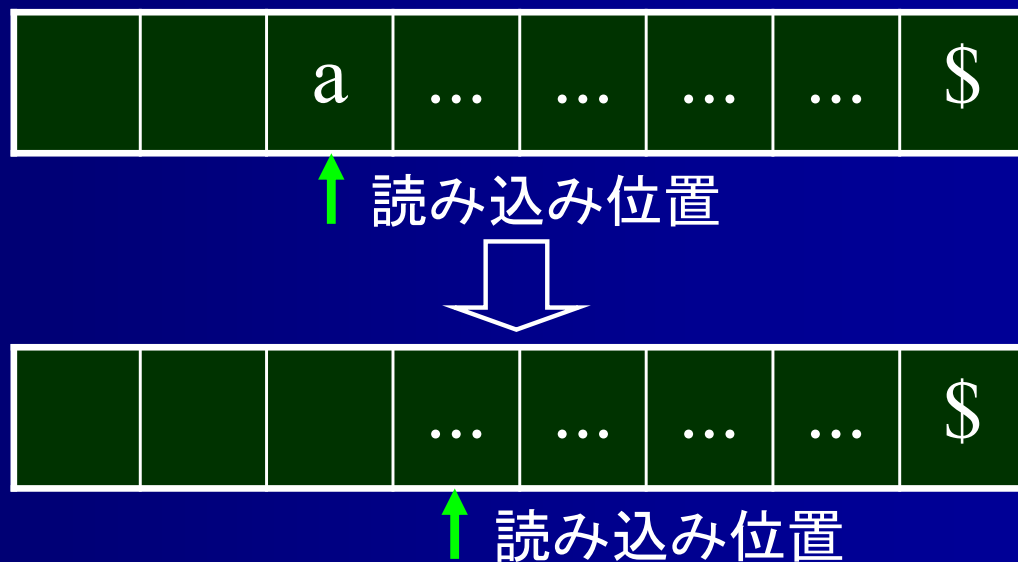
■ X が終端記号のとき

- $X = a \Rightarrow X$ をポップ, 次の文字を読み込む
- $X \neq a \Rightarrow$ 構文解析エラー

スタック



入力記号列



LL解析の手順

X : スタックトップ $a \in T$: 現在の入力記号

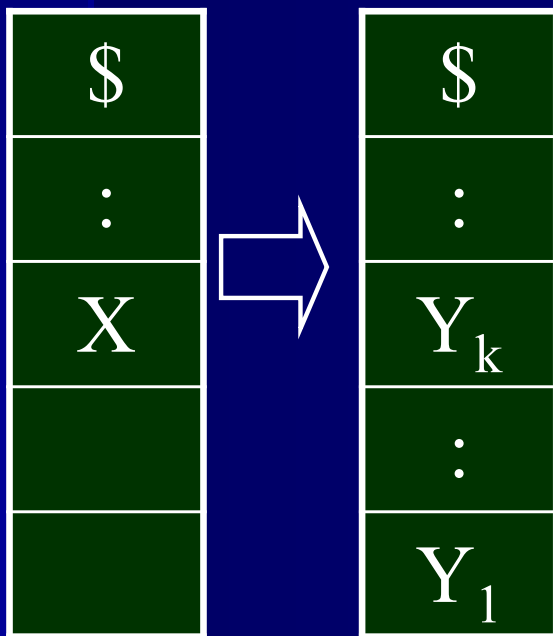
■ X が非終端記号のとき

– 解析表 $M[X, a] = X \rightarrow Y_1 Y_2 Y_3 \dots Y_k$

$\Rightarrow X$ をポップ, $Y_1, Y_2, Y_3, \dots, Y_k$ をプッシュ

スタック

← 後ろから順にプッシュ



解析表

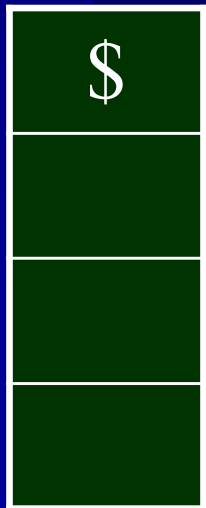
$N \setminus T$		a	
X		$Y_1 Y_2 Y_3 \dots Y_k$	

LL解析の手順

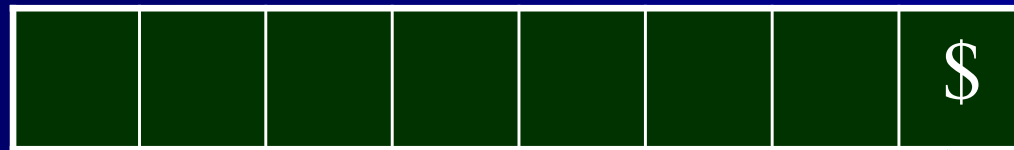
X : スタックトップ $a \in T$: 現在の入力記号

- $X = “\$”$ かつ $a = “\$”$ のとき
 - 解析完了

スタック



入力記号列



↑ 読み込み位置

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$

スタック

\$
E

入力記号列

5	*	(3	+	2)	\$
---	---	---	---	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		TE'			ε	ε
T	FT'					
T'						ε
F	i			(E)		

E → TE' を生成

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$

スタック

\$
E'
T

入力記号列

5	*	(3	+	2)	\$
---	---	---	---	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	ε			
F	i					

T → FT' を生成

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$

スタック

\$
E'
T'
F

入力記号列

5	*	(3	+	2)	\$
---	---	---	---	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'			*FT'			
F	i					

F \rightarrow i (整数) を生成

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$

スタック

\$
E'
T'
i (整数)

入力記号列

5	*	(3	+	2)	\$
---	---	---	---	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	i			TE'		
E'						ε
T						
T'						ε
F	i			(E)		

スタックトップ = 入力記号
⇒ 次の入力を読み込む

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$

スタック

\$
E'
T'

入力記号列

	*	(3	+	2)	\$
--	---	---	---	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'				
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

T' → *FT' を生成

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i|(E)\}$

スタック

\$
E'
T'
F
*

入力記号列

	*	(3	+	2)	\$
--	---	---	---	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	E'			TE'		
E'						ε
T						
T'						ε
F	i			(E)		

スタックトップ = 入力記号
⇒ 次の入力を読み込む

解析例

生成規則

$$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT', \\ T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$$

スタック

\$
E'
T'
F

入力記号列

		(3	+	2)	\$
--	--	---	---	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i|(E)\}$

スタック

\$
E'
T'
)
E
(

入力記号列

		(3	+	2)	\$
--	--	---	---	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$

スタック

\$
E'
T'
)
E

入力記号列

			3	+	2)	\$
--	--	--	---	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i|(E)\}$

スタック

\$
E'
T'
)
E'
T

入力記号列

			3	+	2)	\$
--	--	--	---	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT', \\ T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$$

スタック

\$
E'
T'
)
E'
T'
F

入力記号列

			3	+	2)	\$
--	--	--	---	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i|(E)\}$

スタック

\$
E'
T'
)
E'
T'
i

入力記号列

			3	+	2)	\$
--	--	--	---	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$

スタック

\$
E'
T'
)
E'
T'

入力記号列

				+	2)	\$
--	--	--	--	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \epsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \epsilon, F \rightarrow i | (E)\}$

スタック

\$
E'
T'
)
E'

入力記号列

				+	2)	\$
--	--	--	--	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ϵ	ϵ
T	FT'			FT'		
T'		ϵ	*FT'		ϵ	ϵ
F	i			(E)		

解析例

生成規則

$$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT', \\ T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$$

スタック

\$
E'
T'
)
E'
T
+

入力記号列

				+	2)	\$
--	--	--	--	---	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT', \\ T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$$

スタック

\$
E'
T'
)
E'
T

入力記号列

					2)	\$
--	--	--	--	--	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT', \\ T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$$

スタック

\$
E'
T'
)
E'
T'
F

入力記号列

					2)	\$
--	--	--	--	--	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$

スタック

\$
E'
T'
)
E'
T'
i

入力記号列

					2)	\$
--	--	--	--	--	---	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT', \\ T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$$

スタック

\$
E'
T'
)
E'
T'

入力記号列

)	\$
--	--	--	--	--	--	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i|(E)\}$

スタック

\$
E'
T'
)
E'

入力記号列

)	\$
--	--	--	--	--	--	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT', \\ T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$$

スタック

\$
E'
T'
)

入力記号列

)	\$
--	--	--	--	--	--	---	----

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i|(E)\}$

スタック

\$
E'
T'

入力記号列

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

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT', \\ T' \rightarrow *FT' | \varepsilon, F \rightarrow i | (E)\}$$

スタック

\$
E'

入力記号列

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

解析表

N \ T	i	+	*	()	\$
E	TE'			TE'		
E'		+TE'			ε	ε
T	FT'			FT'		
T'		ε	*FT'		ε	ε
F	i			(E)		

解析例

生成規則

$P = \{E \rightarrow TE', E' \rightarrow +TE' | \varepsilon, T \rightarrow FT',$
 $T' \rightarrow *FT' | \varepsilon, F \rightarrow i|(E)\}$

スタック

\$

入力記号列

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

解析表

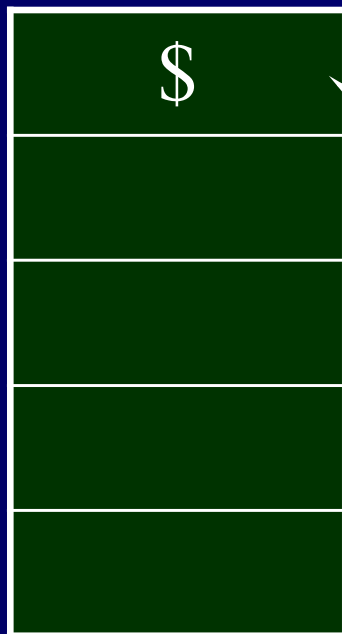
$N \setminus T$	i	$+$	$*$	$($	$)$	$\$$
E	TE'			TE'		
E'		$+TE'$			ε	ε
T	FT'			FT'		
T'		ε	$*FT'$		ε	ε
F	i			(E)		

スタック	入力記号列	生成	スタック	入力記号列	生成
E\$	5*(3+2)\$	$E \rightarrow TE'$	T'E')T'E'\$	+2)\$	$T' \rightarrow \epsilon$
TE'\$	5*(3+2)\$	$T \rightarrow FT'$	E')T'E'\$	+2)\$	$E' \rightarrow +TE'$
FT'E'\$	5*(3+2)\$	$F \rightarrow i$	+TE')T'E'\$	+2)\$	
iT'E'\$	5*(3+2)\$		TE')T'E'\$	2)\$	$T \rightarrow FT'$
T'E'\$	*(3+2)\$	$T' \rightarrow *FT'$	FT'E')T'E'\$	2)\$	$F \rightarrow i$
*FT'E'\$	*(3+2)\$		iT'E')T'E'\$	2)\$	
FT'E'\$	(3+2)\$	$F \rightarrow (E)$	T'E')T'E'\$)\$	$T' \rightarrow \epsilon$
(E)T'E'\$	(3+2)\$		E')T'E'\$)\$	$E' \rightarrow \epsilon$
E)T'E'\$	3+2)\$	$E \rightarrow TE'$)T'E'\$)\$	
TE')T'E'\$	3+2)\$	$T \rightarrow FT'$	T'E'\$	\$	$T' \rightarrow \epsilon$
FT'E')T'E'\$	3+2)\$	$F \rightarrow i$	E'\$	\$	$E' \rightarrow \epsilon$
iT'E')T'E'\$	3+2)\$		\$	\$	

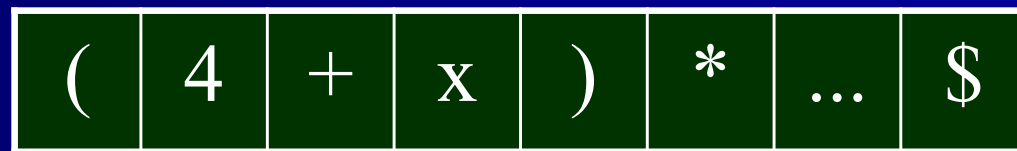
移動還元構文解析 (shift reduce parsing)

- 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



トップに“\$”

末尾に“\$”

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 (スタックトップが開始記号 かつ 入力記号が “\$”)
解析終了

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

$\langle \text{namelist} \rangle ::= \langle \text{name} \rangle \mid \langle \text{namelist} \rangle \text{ “,” } \langle \text{name} \rangle$

$\langle \text{name} \rangle ::= \text{ “a” } \mid \text{ “b” } \mid \text{ “c” }$

“a” “,” “b”

→ ⟨name⟩ “,” “b”

→ ⟨namelist⟩ “,” “b”

⟨name⟩ → “a” の還元

⟨namelist⟩ → ⟨name⟩

の還元

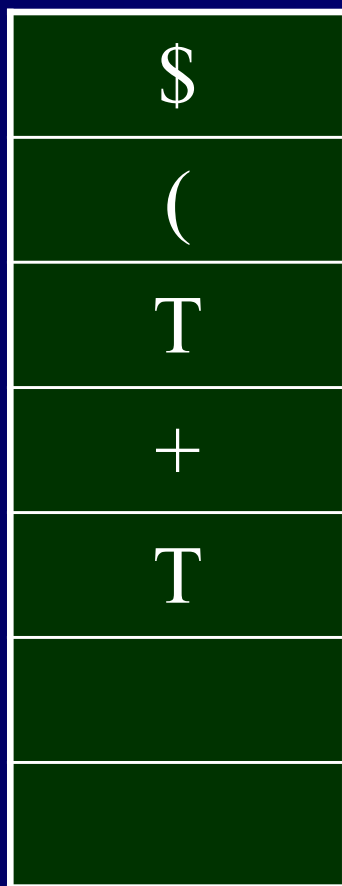
ハンドル

還元

生成規則

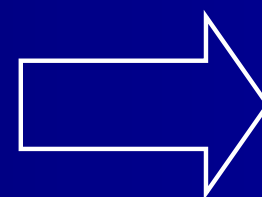
$$P = \{E \rightarrow T+T, T \rightarrow F * F, F \rightarrow i, F \rightarrow (E)\}$$

スタック



右辺 $T+T$ をポップ
左辺 E をプッシュ

$E \rightarrow T+T$ の
右辺と一致
= ハンドル



出力

ADD



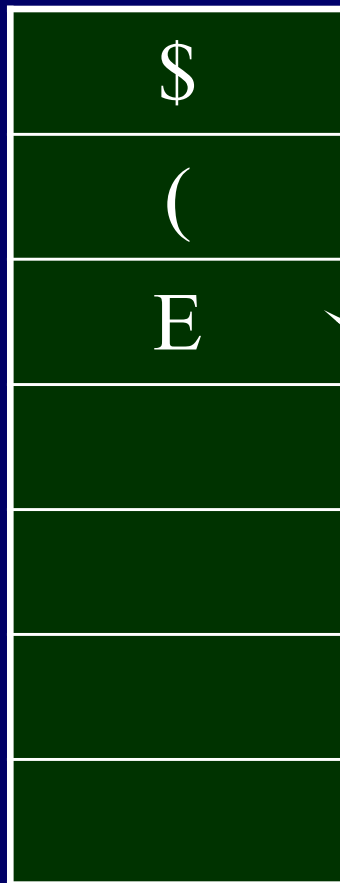
対応する
コードを出力

移動(shift)

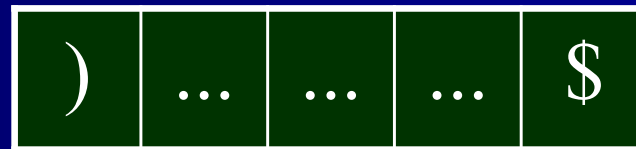
生成規則

スタック

$P = \{E \rightarrow T+T, T \rightarrow F * F, F \rightarrow i, F \rightarrow (E)\}$

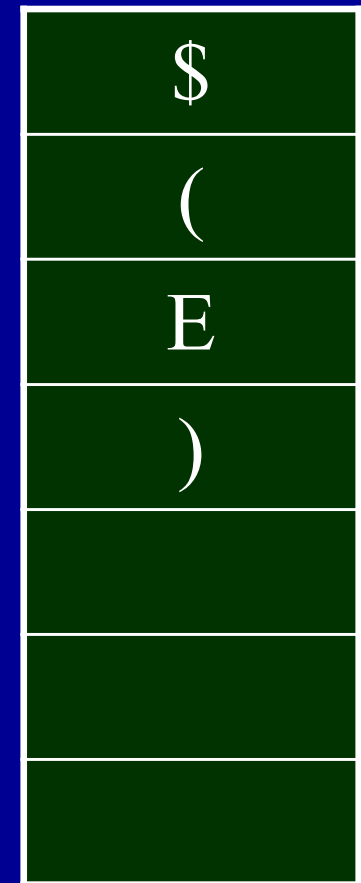


入力記号列



右辺と
不一致

入力記号を
プッシュ



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
↑
読み込み位置

$E \rightarrow T + T$ \Rightarrow 読み込み位置を移動
↑
読み込み位置

$E \rightarrow T + T$ \Rightarrow 読み込み位置を移動
↑
読み込み位置

$E \rightarrow T + T$ \Rightarrow 読み込み完了, 還元する
↑
読み込み位置

解析例

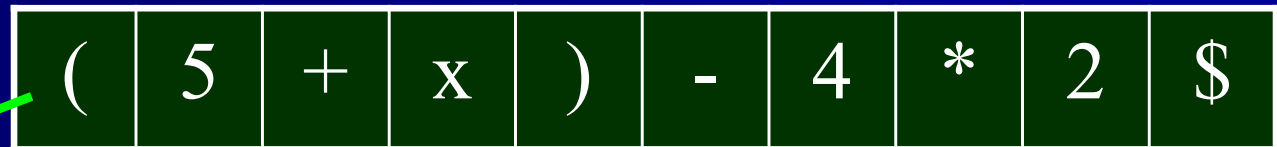
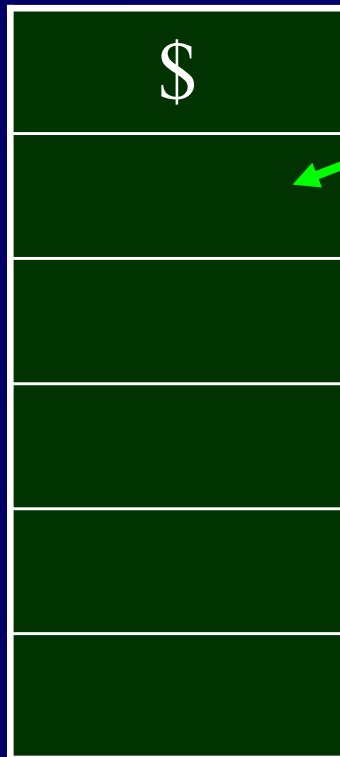
$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

入力記号列



移動



解析例

$(5 + x) - 4 * 2$ の解析

生成規則

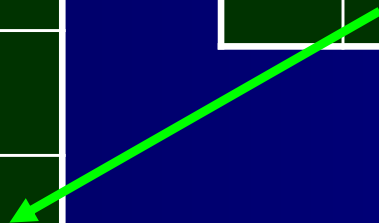
$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

入力記号列

\$
(

	5	+	x)	-	4	*	2	\$
--	---	---	---	---	---	---	---	---	----



解析例

$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

入力記号列

\$
(
5

		+	x)	-	4	*	2	\$
--	--	---	---	---	---	---	---	---	----

スタックトップが
 $E \rightarrow i$ (整数) に
一致



解析例

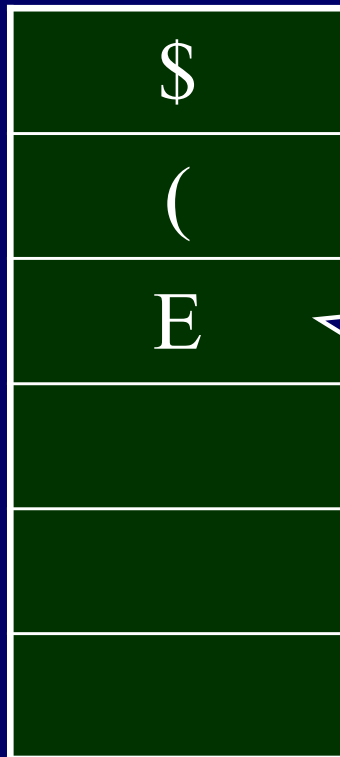
$(5 + x) - 4 * 2$ の解析

生成規則

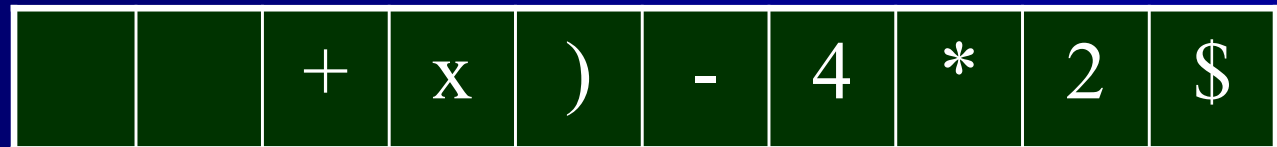
$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

入力記号列



整数を
E に還元



PUSHI 5

整数に対応する
コードを出力

解析例

$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

入力記号列

\$
(
E
+

			x)	-	4	*	2	\$
--	--	--	---	---	---	---	---	---	----

PUSHI 5

解析例

$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

入力記号列

\$
(
E
+
x

)	-	4	*	2	\$
--	--	--	--	---	---	---	---	---	----

$E \rightarrow n$ (変数) に
一致

PUSH 5

解析例

(5 + x) - 4 * 2 の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

入力記号列

\$
(
E
+
E

)	-	4	*	2	\$
--	--	--	--	---	---	---	---	---	----

$E \rightarrow E + E$ に
一致

PUSHI 5
PUSH &x

解析例

$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

\$
(
E

入力記号列

)	-	4	*	2	\$
--	--	--	--	---	---	---	---	---	----

```
PUSHI 5  
PUSH &x  
ADD
```


解析例

$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

入力記号列

\$
(
E
)

					-	4	*	2	\$
--	--	--	--	--	---	---	---	---	----

$E \rightarrow (E)$ に
一致

PUSHI 5
PUSH &x
ADD

解析例

$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

入力記号列

\$
E

					-	4	*	2	\$
--	--	--	--	--	---	---	---	---	----

PUSHI 5
PUSH &x
ADD

解析例

$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

入力記号列

\$
E
-

						4	*	2	\$
--	--	--	--	--	--	---	---	---	----

```
PUSHI 5  
PUSH &x  
ADD
```

解析例

$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

\$
E
-
4

入力記号列

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

$E \rightarrow i$ (整数)

```
PUSHI 5  
PUSH &x  
ADD
```

解析例

$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

\$
E
-
E

入力記号列

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

```
PUSHI 5  
PUSH &x  
ADD  
PUSHI 4
```

解析例

$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

\$
E
-
E
*

入力記号列

								2	\$
--	--	--	--	--	--	--	--	---	----

```
PUSHI 5
PUSH &x
ADD
PUSHI 4
```

解析例

$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

\$
E
-
E
*
2

入力記号列

								2	\$
--	--	--	--	--	--	--	--	---	----

$E \rightarrow i$ (整数)

```
PUSHI 5
PUSH &x
ADD
PUSHI 4
```

解析例

$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

入力記号列

\$
E
-
E
*
E

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

$E \rightarrow E * E$

```
PUSHI 5
PUSH &x
ADD
PUSHI 4
PUSHI 2
```

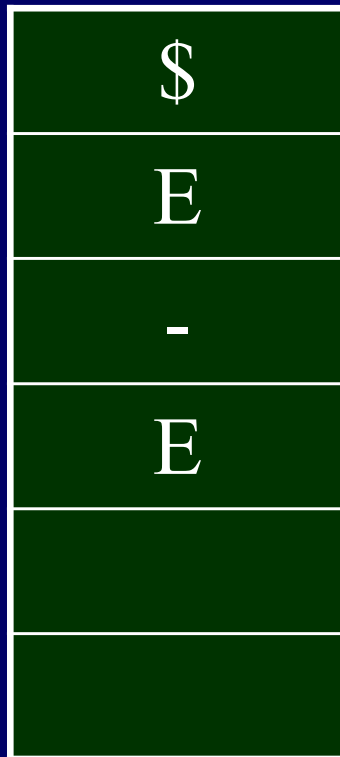

解析例

$(5 + x) - 4 * 2$ の解析

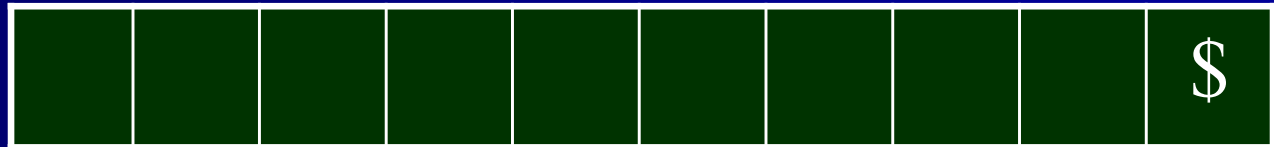
生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック



入力記号列



$E \rightarrow E - E$

```
PUSHI 5
PUSH &x
ADD
PUSHI 4
PUSHI 2
MUL
```

解析例

$(5 + x) - 4 * 2$ の解析

生成規則

$P = \{E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid i \mid n\}$

スタック

入力記号列

\$
E

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

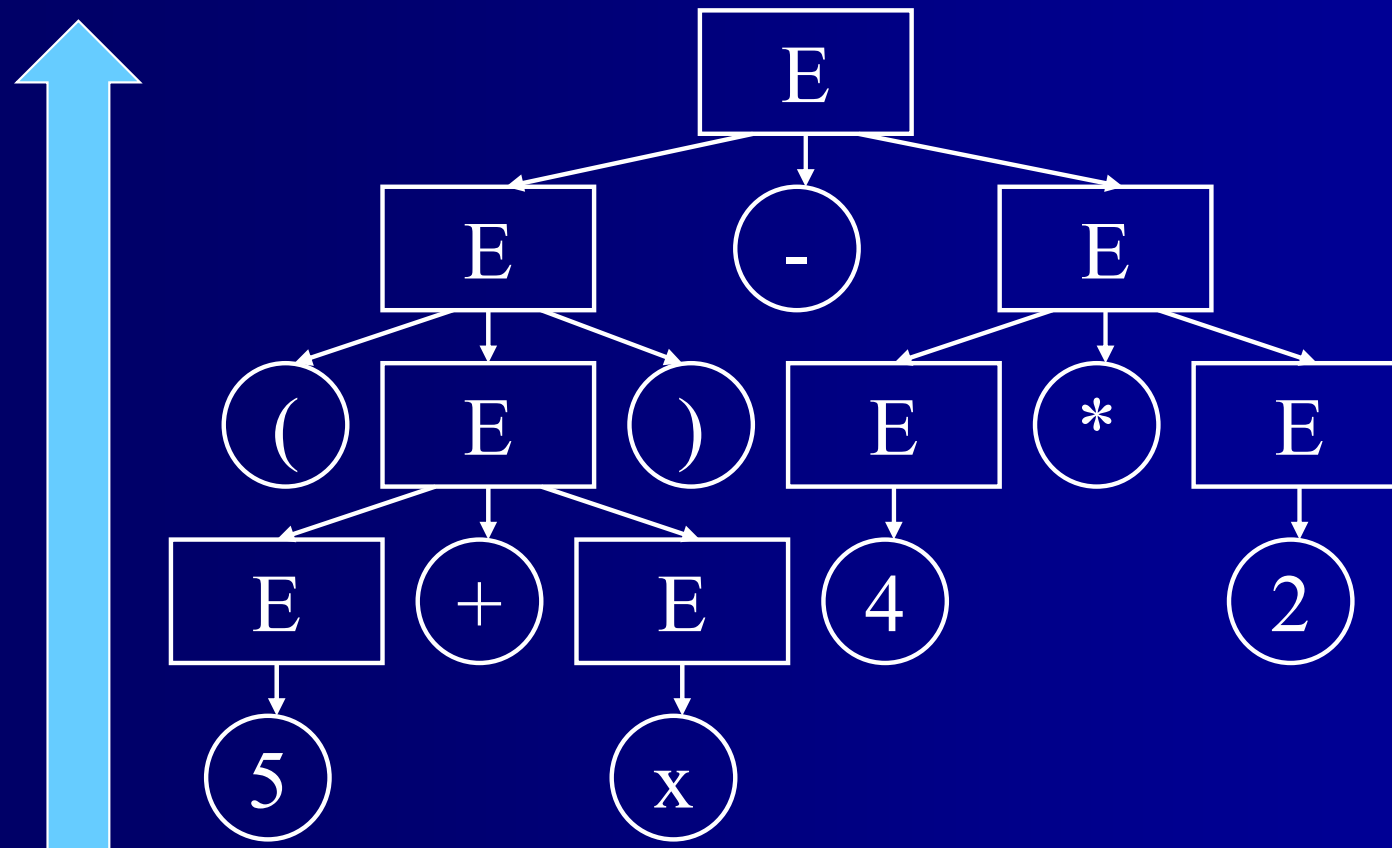
```
PUSHI 5
PUSH &x
ADD
PUSHI 4
PUSHI 2
MUL
SUB
```

スタックトップが開始記号
かつ 入力記号が “\$”
⇒ 解析完了

移動還元構文解析の導出木

- 最右導出の導出木が生成される

入力記号列 : $(5 + x) - 4 * 2$



スタック	入力列	操作	出力
\$	(5 + x) - 4 * 2 \$	移動	
\$ (5 + x) - 4 * 2 \$	移動	
\$ (5	+ x) - 4 * 2 \$	還元 $E \rightarrow i$ (整数)	PUSHI 5
\$ (E	+ x) - 4 * 2 \$	移動	
\$ (E +	x) - 4 * 2 \$	移動	
\$ (E + x) - 4 * 2 \$	還元 $E \rightarrow n$ (変数)	PUSH &x
\$ (E + E) - 4 * 2 \$	還元 $E \rightarrow E + E$	ADD
\$ (E) - 4 * 2 \$	移動	
\$ (E)	- 4 * 2 \$	還元 $E \rightarrow (E)$	
\$ E	- 4 * 2 \$	移動	
\$ E -	4 * 2 \$	移動	
\$ E - 4	* 2 \$	還元 $E \rightarrow i$ (整数)	PUSHI 4
\$ E - E	* 2 \$	移動	
\$ E - E *	2 \$	移動	
\$ E - E * 2	\$	還元 $E \rightarrow i$ (整数)	PUSHI 2
\$ E - E * E	\$	還元 $E \rightarrow E * E$	MUL
\$ E - E	\$	還元 $E \rightarrow E - E$	SUB

ここは還元 $E \rightarrow E - E$ ではなく移動

演算子順位構文解析 (operator precedence parsing)

■ 演算子間の優先順位を定義

- $A \ll B$: A の優先順位 $<$ B の優先順位
- $A \gg B$: A の優先順位 $>$ B の優先順位
- $A == B$: A と B は同じハンドル内

例 : $* \gg +$ $*$ は $+$ よりも優先順位が高い

例 : $(==)$ $($ と $)$ 同じハンドル内

記号の優先順位 (演算子と被演算子)

- 記号 A が B よりも優先順位が高い
= 逆ポーランド記法にしたときに A が先に来る

例 : $x + 5$ $x, 5, +$ + より $x, 5$ が先

(x の優先順位) > (+ の優先順位)

(+ の優先順位) < (5 の優先順位)

$x \gg +, + \ll 5$

被演算子と演算子とでは常に被演算子優先

記号の優先順位 (+と*)

例 : $x + y * z$

$x, y, z, *, +$ $+ \ll *$

例 : $x * y + z$

$x, y, *, z, +$ $* \gg +$

* と + とでは常に * 優先

記号の優先順位 (+同士)

例 : $x +_1 y +_2 z$

$x, y, +_1, z, +_2 \quad +_1 \gg +_2$

例 : $x + y - z$

$x, y, +, z, - \quad + \gg -$

例 : $x - y + z$

$x, y, -, z, + \quad - \gg +$

+ 同士, - 同士, + と - とでは
先に来た方が優先 (左結合的)

記号の優先順位 (=同士)

例 : $x =_1 y =_2 z$

$x, y, z, =_2, =_1 \quad =_1 \ll =_2$

= 同士では後に来た方優先 (右結合的)

記号の優先順位 (\$)

例 : \$ (数式) \$

全ての処理が終われば \$ を処理

⇒ \$ は優先順位最低

\$ << (全て), (全て) >> \$

\$ ** \$: \$ 同士ならば受理

記号の優先順位

例 : $(5 + 2) * (7 - 6)$

5, 2, +, 7, 6, -, *

() がある場合は () 内を優先

(<< (全て), (全て) << (
) >> (全て), (全て) >>)

演算子の優先順位

■ 演算子 f, g

– f が g より優先順位が高い $\Rightarrow f \gg g, g \ll f$

– f と g の優先順位が同じ

■ f, g が左結合的 $\Rightarrow f \gg g, g \gg f$

■ f, g が右結合的 $\Rightarrow f \ll g, g \ll f$

優先順位表

右側 左側	整数	変数	+, -	*, /	=	()	\$
整数			>>	>>			>>	>>
変数			>>	>>	>>		>>	>>
+, -	<<	<<	>>	<<		<<	>>	>>
*, /	<<	<<	>>	>>		<<	>>	>>
=	<<	<<	<<	<<	<<	<<	>>	>>
(<<	<<	<<	<<	<<	<<	==	
)			>>	>>			>>	>>
\$	<<	<<	<<	<<	<<	<<		**

空欄は構文解析エラー

解析手順

1. 入力記号列から非終端記号を取り除く
2. 入力記号列に優先順位を挿入する
3. 左から見て最初の \gg を探す
4. 3.の位置から最も手前の \ll を探す
5. \ll から \gg までを還元する

$\$ ** \$$ になれば受理

解析例

- 入力列 $\$(5 + x) - 4 * 2 \$$

$\$ \langle \langle \langle \underline{5} \rangle \rangle + \langle \langle x \rangle \rangle \rangle - \langle \langle 4 \rangle \rangle * \langle \langle 2 \rangle \rangle \$$

$\langle \langle \text{と} \rangle \rangle$ の間がハンドル

$\Rightarrow \rangle \rangle$ があればその左側を還元

5 を $E \rightarrow i$ で還元

$\$(E + x) - 4 * 2 \$$

解析例

優先順位判定では
非終端記号は削除

還元後の記号列: $\$ (E + x) - 4 * 2 \$$

$\$ \langle \langle (\langle \langle + \langle \langle x \rangle \rangle) \rangle \rangle - \langle \langle 4 \rangle \rangle * \langle \langle 2 \rangle \rangle \$$

x を $E \rightarrow n$ で還元 $\$ (E + E) - 4 * 2 \$$

$\$ \langle \langle (\langle \langle + \rangle \rangle) \rangle \rangle - \langle \langle 4 \rangle \rangle * \langle \langle 2 \rangle \rangle \$$

$+$ を $E \rightarrow E + E$ で還元 $\$ (E) - 4 * 2 \$$

$\$ \langle \langle (=) \rangle \rangle - \langle \langle 4 \rangle \rangle * \langle \langle 2 \rangle \rangle \$$

$=$ は同じハンドル内

解析例

還元後の記号列: $\$ (E) - 4 * 2 \$$

$\$ \underline{\langle \langle (==) \rangle \rangle} - \langle \langle 4 \rangle \rangle * \langle \langle 2 \rangle \rangle \$$

() を $E \rightarrow (E)$ で還元 $\$ E - 4 * 2 \$$

$\$ \langle \langle - \underline{\langle \langle 4 \rangle \rangle} * \langle \langle 2 \rangle \rangle \rangle \$$

4 を $E \rightarrow i$ で還元 $\$ E - E * 2 \$$

$\$ \langle \langle - \langle \langle * \underline{\langle \langle 2 \rangle \rangle} \rangle \rangle \$$

解析例

還元後の記号列: $\$ E - E * 2 \$$

$\$ \langle \langle - \langle \langle * \langle \langle 2 \rangle \rangle \rangle \rangle \rangle \$$

2 を $E \rightarrow i$ で還元

$\$ E - E * E \$$

$\$ \langle \langle - \langle \langle * \rangle \rangle \rangle \$$

* を $E \rightarrow E * E$ で還元

$\$ E - E \$$

$\$ \langle \langle - \rangle \rangle \$$

解析例

還元後の記号列 : $\$ E - E \$$

$\$ \langle \langle - \rangle \rangle \$$

- を $E \rightarrow E-E$ で還元 $\$ E \$$

$\$ ** \$$

$\$ ** \$$ になったので受理

解析例

■ 入力列 $\$(5 + x) - 4 * 2 \$$

入力列	優先順位付記号列	還元	出力
$\$(\underline{5} + x) - 4 * 2 \$$	$\$ \ll (\ll \underline{5} \gg + \ll x \gg) \gg - \ll 4 \gg * \ll 2 \gg \$$	$E \rightarrow i$	PUSHI 5
$\$(E + \underline{x}) - 4 * 2 \$$	$\$ \ll (\ll + \ll \underline{x} \gg) \gg - \ll 4 \gg * \ll 2 \gg \$$	$E \rightarrow n$	PUSH &x
$\$(\underline{E} + \underline{E}) - 4 * 2 \$$	$\$ \ll (\ll + \gg) \gg - \ll 4 \gg * \ll 2 \gg \$$	$E \rightarrow E + E$	ADD
$\$(\underline{E}) - 4 * 2 \$$	$\$ \ll (\ll (=) \gg) \gg - \ll 4 \gg * \ll 2 \gg \$$	$E \rightarrow (E)$	
$\$ E - \underline{4} * 2 \$$	$\$ \ll - \ll \underline{4} \gg * \ll 2 \gg \$$	$E \rightarrow i$	PUSHI 4
$\$ E - E * \underline{2} \$$	$\$ \ll - \ll * \ll \underline{2} \gg \gg \$$	$E \rightarrow i$	PUSHI 2
$\$ E - \underline{E} * \underline{E} \$$	$\$ \ll - \ll * \gg \gg \$$	$E \rightarrow E * E$	MUL
$\$ \underline{E} - \underline{E} \$$	$\$ \ll - \gg \$$	$E \rightarrow E - E$	SUB
$\$ \underline{E} \$$	$\$ ** \$$		

演算子順位構文解析

■ 演算子間の優先順位を定義

- $A \ll B$: A の優先順位 $<$ B の優先順位
- $A \gg B$: A の優先順位 $>$ B の優先順位
- $A == B$: A と B は同じハンドル内

■ 演算子の優先順位から操作を決定

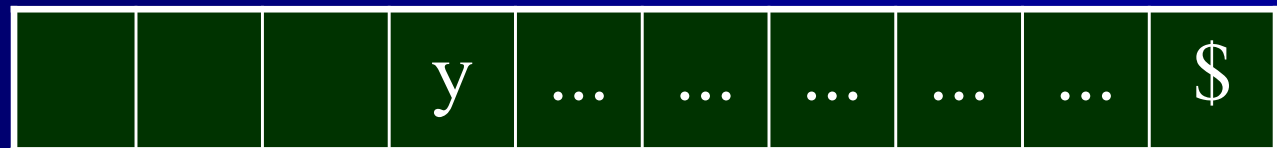
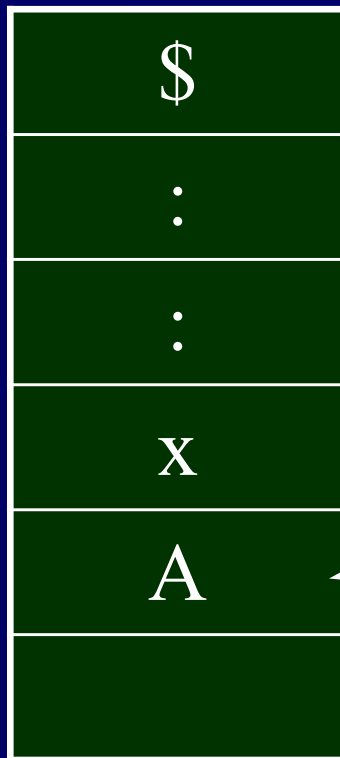
- $A \ll B \Rightarrow B$ をスタックに移動
- $A \gg B \Rightarrow A$ を含むハンドルを還元
- $A == B \Rightarrow B$ をスタックに移動
- $A ** B \Rightarrow$ 受理 ($\$ ** \$$ のみ)
- $A \text{ err } B \Rightarrow$ 構文解析エラー

解析手順

- スタックトップの終端記号 x の優先順位と
入力記号の先頭 y の優先順位を比較

スタック

入力記号列

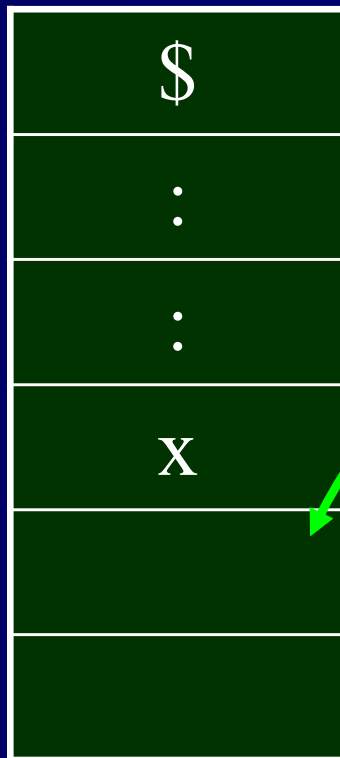


スタックトップが
非終端記号の場合は
スタックの2番目で判定

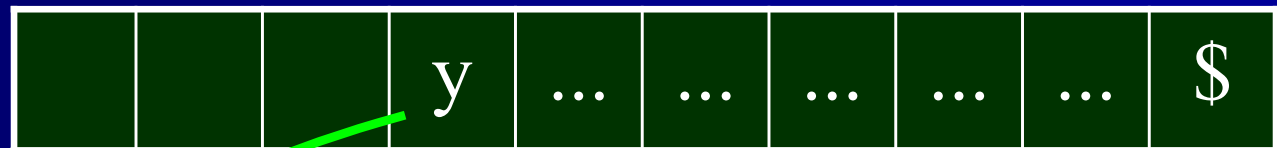
解析手順

- スタックトップの終端記号 x の優先順位と
入力記号の先頭 y の優先順位を比較

スタック



入力記号列



$x \ll y$

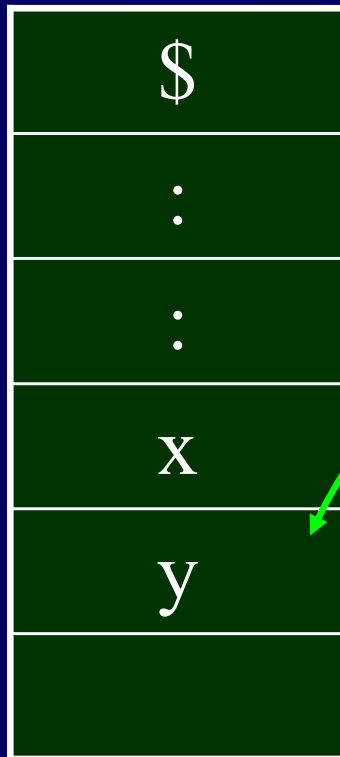
$\Rightarrow y$ を移動



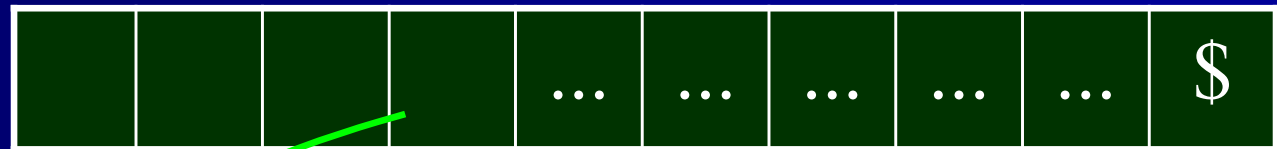
解析手順

- スタックトップの終端記号 x の優先順位と
入力記号の先頭 y の優先順位を比較

スタック



入力記号列



$x \ll y$

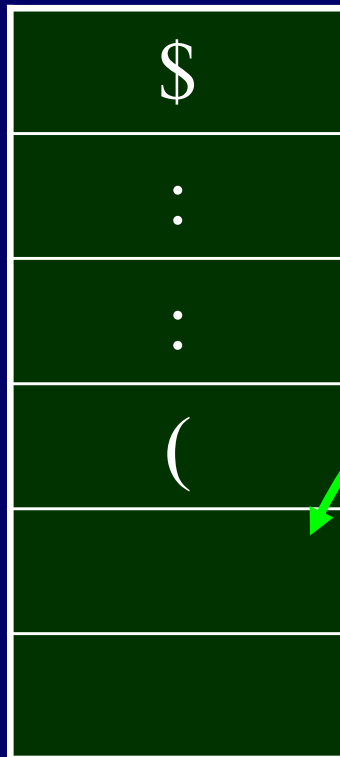
$\Rightarrow y$ を移動



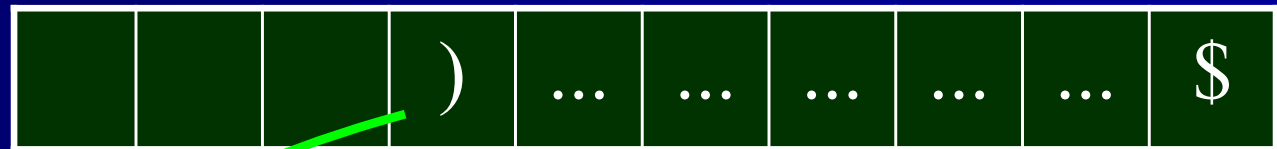
解析手順

- スタックトップの終端記号 x の優先順位と
入力記号の先頭 y の優先順位を比較

スタック



入力記号列



(==)

⇒) を移動

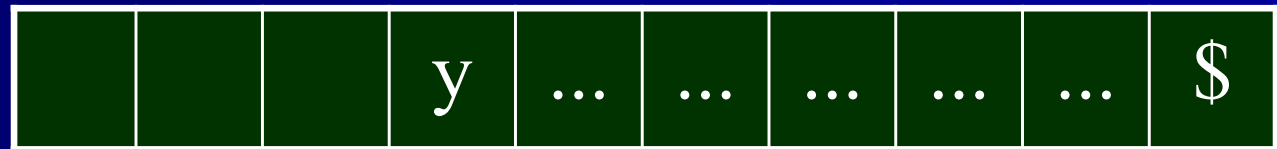
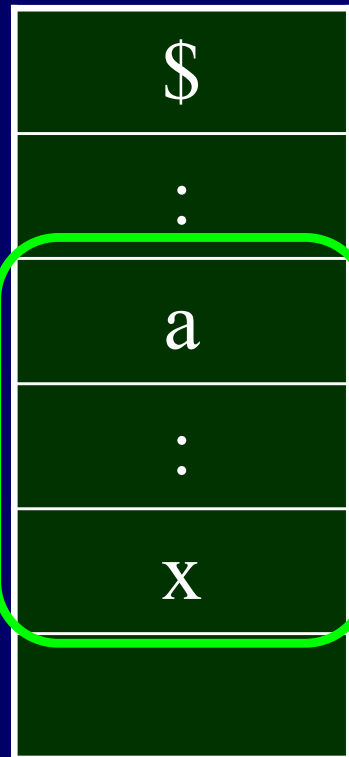


解析手順

- スタックトップの終端記号 x の優先順位と
入力記号の先頭 y の優先順位を比較

スタック

入力記号列



ハンドル

$x \gg y$

$\Rightarrow x$ を含む
ハンドルを還元

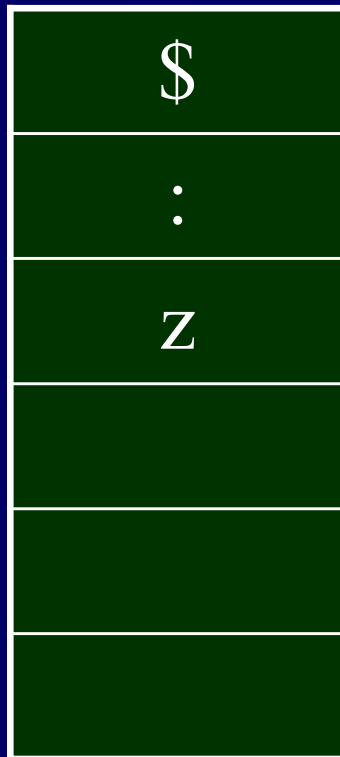
$Z \rightarrow a...x$



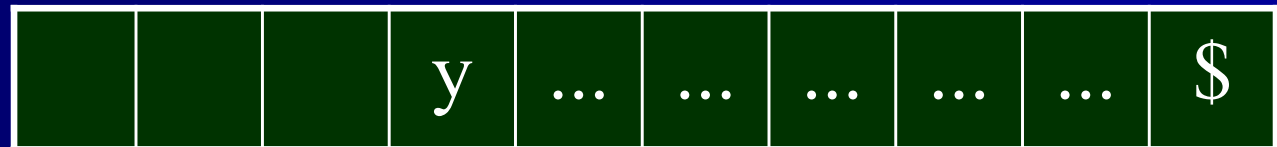
解析手順

- スタックトップの終端記号 x の優先順位と
入力記号の先頭 y の優先順位を比較

スタック



入力記号列



$x \gg y$

⇒ x を含む
ハンドルを還元

$z \rightarrow a...x$

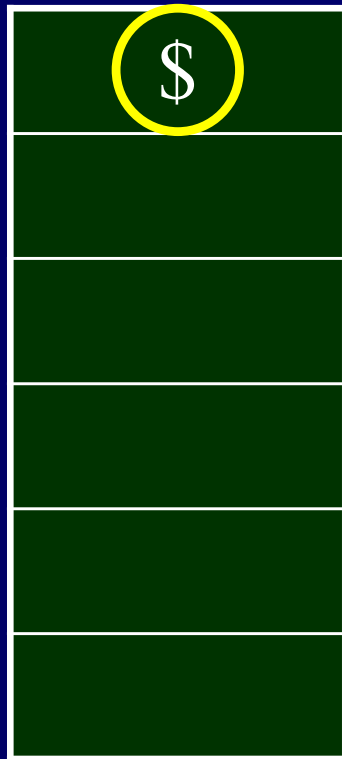
x のコード

対応するコードを
出力

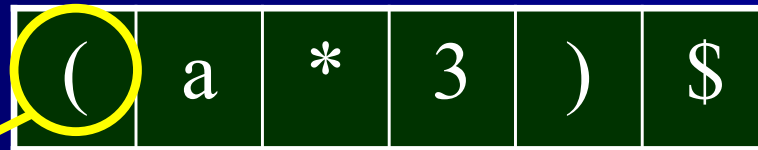
解析例

- 入力列 $\$(a * 3) \$$

スタック

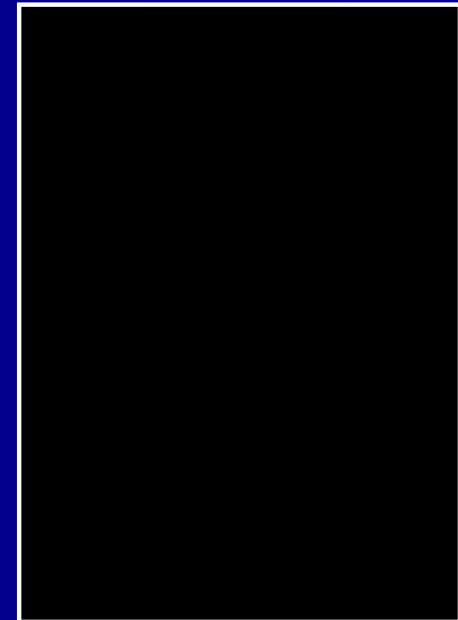


入力記号列



$\$ \ll ($

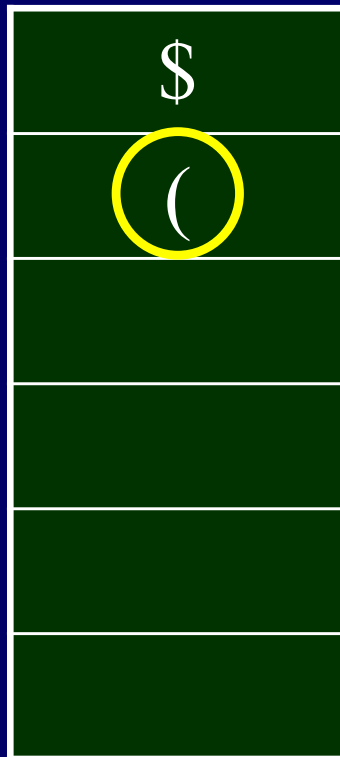
$\Rightarrow ($ を移動



解析例

■ 入力列 $\$(a * 3)\$$

スタック



入力記号列



(<< a

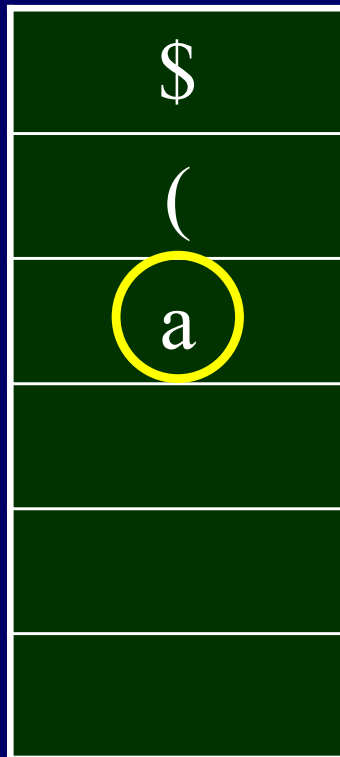
⇒ a を移動



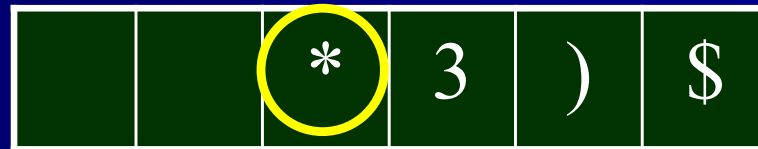
解析例

- 入力列 $\$(a * 3)\$$

スタック

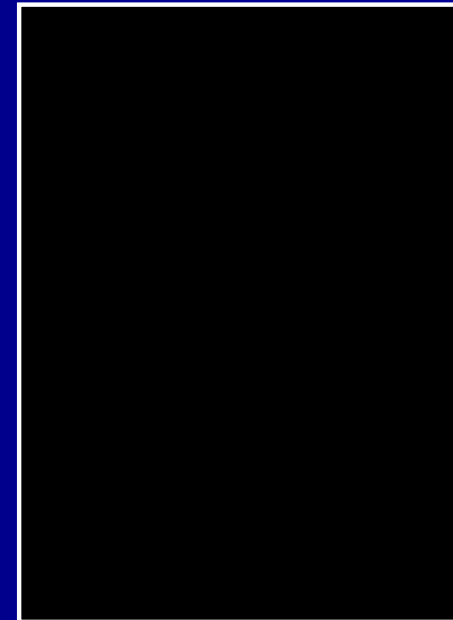


入力記号列



$a \gg *$

$\Rightarrow a$ を還元

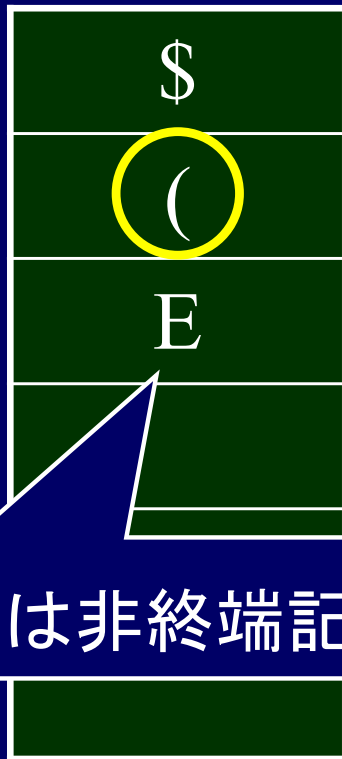


解析例

■ 入力列 $\$(a * 3)\$$

スタック

入力記号列



(<< *
⇒ * を移動

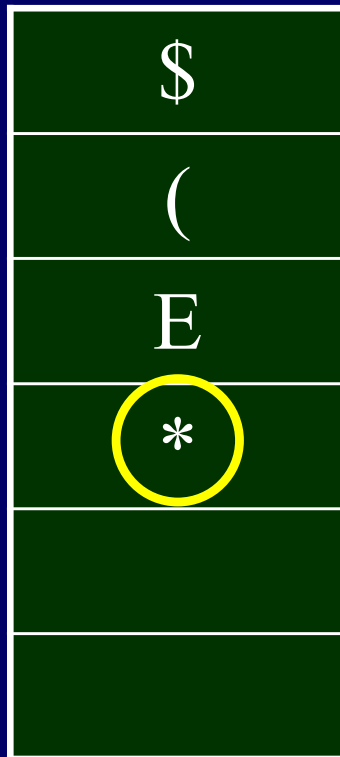
E は非終端記号

PUSH &a

解析例

- 入力列 $\$(a * 3)\$$

スタック



入力記号列



* << 3

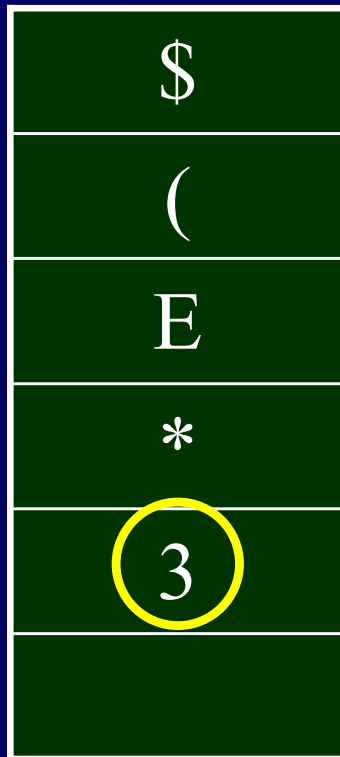
⇒ 3 を移動

PUSH &a

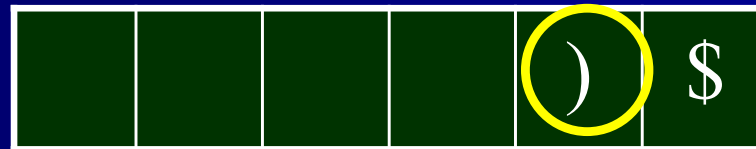
解析例

■ 入力列 \$(a * 3) \$

スタック



入力記号列



3 >>>)

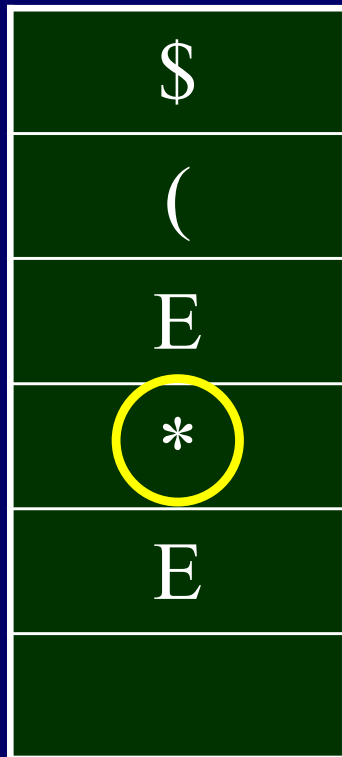
⇒ 3 を還元

PUSH &a

解析例

- 入力列 $\$(a * 3) \$$

スタック



入力記号列



* >>>)

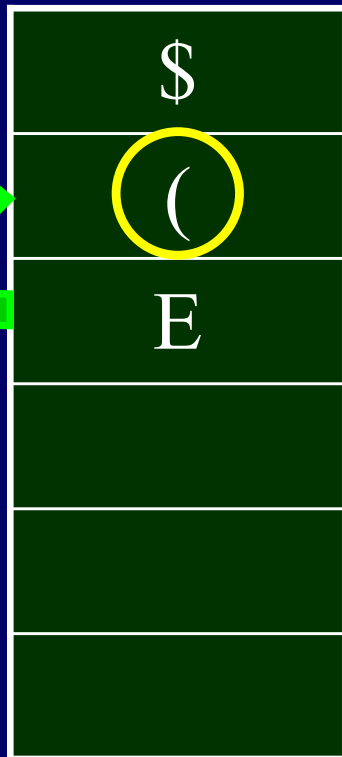
⇒ $E * E$ を還元

```
PUSH &a  
PUSHI 3
```

解析例

- 入力列 \$(a * 3) \$

スタック



入力記号列



(==)

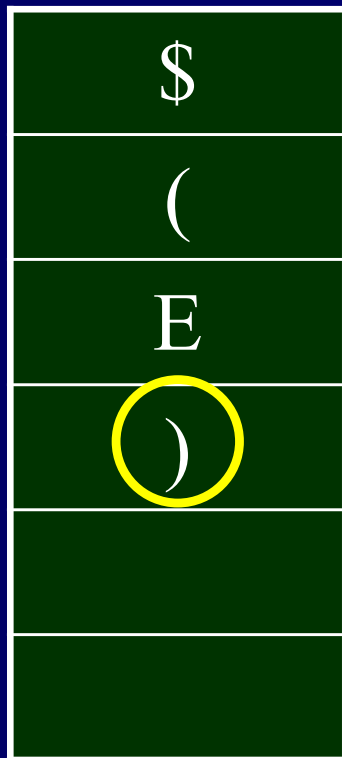
⇒) を移動

```
PUSH &a  
PUSHI 3  
MUL
```

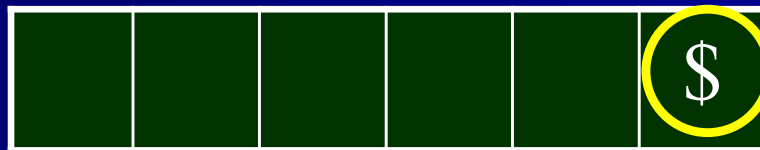
解析例

- 入力列 $\$(a * 3) \$$

スタック



入力記号列



) >>> \$

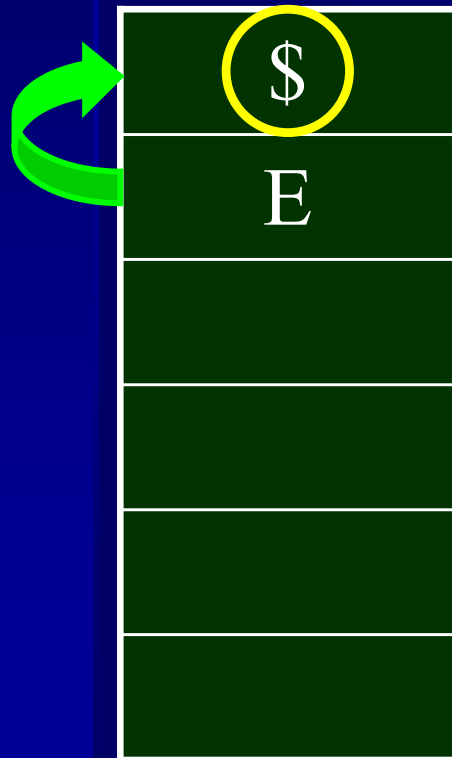
⇒ (E) を還元

```
PUSH &a  
PUSHI 3  
MUL
```

解析例

- 入力列 $\$(a * 3) \$$

スタック



入力記号列



$\$ ** \$$

\Rightarrow 受理

```
PUSH &a  
PUSHI 3  
MUL
```

解析例

■ 入力列 $(a * 3)$

スタック	入力列	優先順位付記号列	判定	操作
\$	(a * 3) \$	\$ << (<< a >> * << 3 >>) >> \$	\$ << (移動
\$ (a * 3) \$	\$ << (<< a >> * << 3 >>) >> \$	(<< a	移動
\$ (a	* 3) \$	\$ << (<< a >> * << 3 >>) >> \$	a >> *	還元
\$ (E	* 3) \$	\$ << (<< * << 3 >>) >> \$	(<< *	移動
\$ (E *	3) \$	\$ << (<< * << 3 >>) >> \$	* << 3	移動
\$ (E * 3) \$	\$ << (<< * << 3 >>) >> \$	x >>)	還元
\$ (E * E) \$	\$ << (<< * >>) >> \$	* >>)	還元
\$ (E) \$	\$ << (= =) >> \$	(= =)	移動
\$ (E)	\$	\$ << (= =) >> \$) >> \$	還元
\$ E	\$	\$ * * \$	\$ * * \$	受理

優先順位の数値化

- 優先順位には半順序関係が成立

- 半順序関係

- $A \ll B \wedge B \ll C \Rightarrow A \ll C$

⇒各記号に整数値を割り当てる
(優先順位が高い方に大きい数値)

例: “+” = 3, “*” = 5

解析手順

- スタックトップの終端記号 x の優先順位 $p(x)$ と
入力記号の先頭 y の優先順位 $q(y)$ を比較

例：スタックトップ： $+$
入力記号の先頭： $*$

$$p(+)=4, q(*)=5 \quad \Rightarrow + \lll *$$

演算子順位構文解析の問題点

- 複数の順位を持つ演算子の存在

- 例：2項演算子の - と単項演算子の -

- * >> 2項演算子の - * << 単項演算子の -

- 両者を区別する必要あり

- 下降型構文解析なら構文解析時に区別可能

- 上昇型構文解析では構文解析時に区別不可能

- ⇒ 字句解析時に区別

演算子の区別

- $\langle \text{Exp} \rangle ::= \langle \text{Term} \rangle \{ (+ | -) \langle \text{Term} \rangle \}$
 - 2項演算子の - は $\langle \text{Term} \rangle$ の後にのみ
 - ⇒ if (一つ前のトークンが $\langle \text{Term} \rangle$ の末尾)
2項演算子の -, else 単項演算子の -

$\langle \text{Term} \rangle$ の末尾:

NAME, INTEGER, “inputint” “]” “)” 等

演算子の区別

■ 字句解析プログラム(の一部)

```
if (currentChar == "-") {  
    if (lookAhead() == "=") {  
        currentChar = nextChar();  
        token = new Token (ASSIGNSUB);  
    } else if (lookAhead == "-") {  
        currentChar = nextChar();  
        token = new Token (DEC);  
    } else if (prevToken == NAME  
        | prevToken == INTEGER  
        | prevToken == "inputint" ... ) {  
        token = new Token (SUB); // 2項演算子  
    } else token = new Token (CSIGN); // 単項演算子  
}
```

直前の
トークンで
判定