2018 年 6 月 19 日

LR 構文解析概説

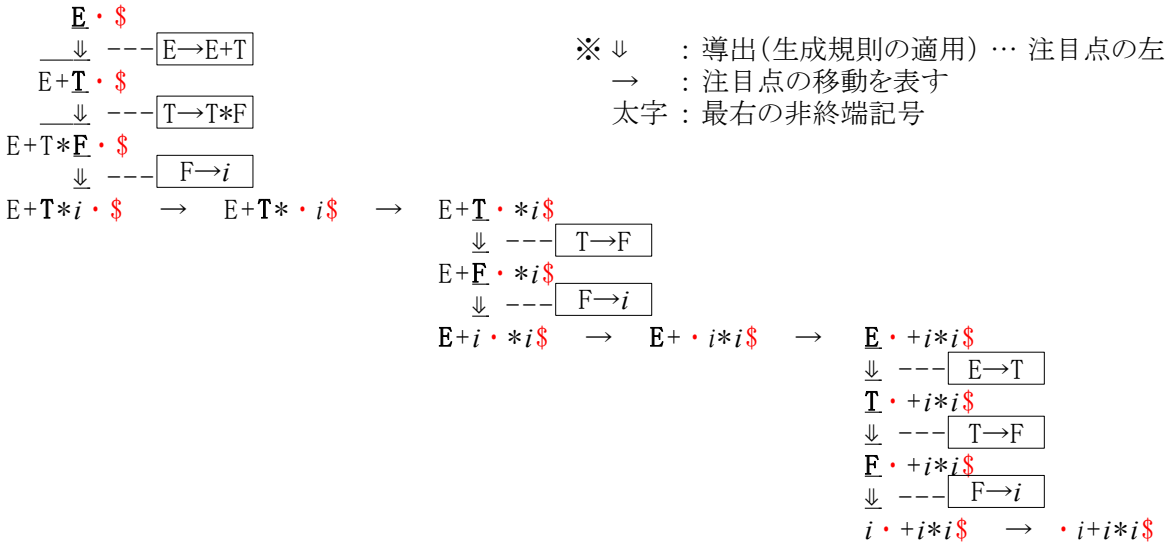
三浦 欽也

基本原理

最右導出の逆を行なう.

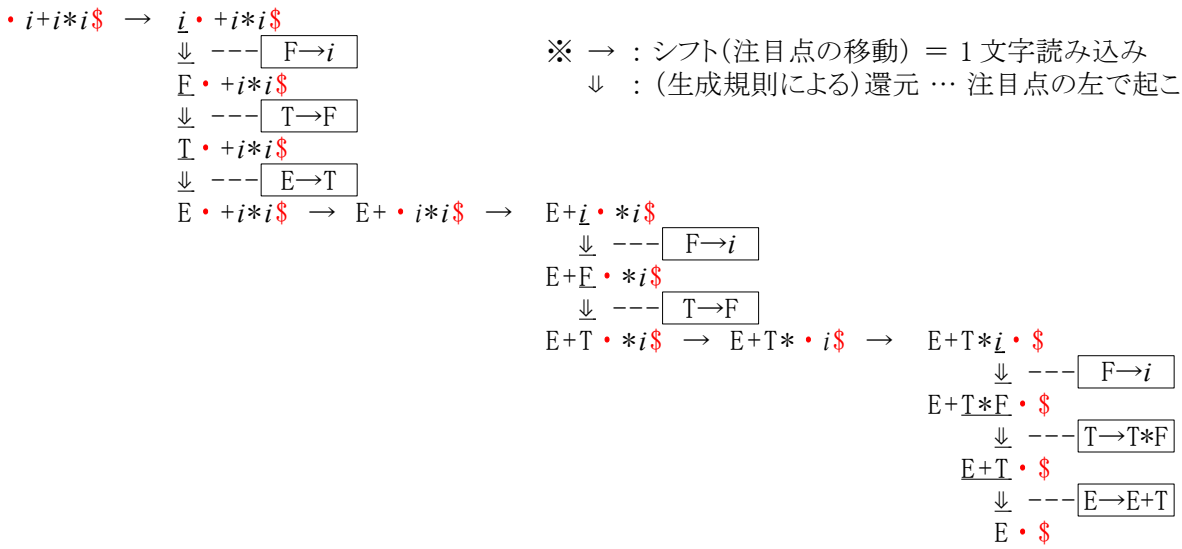
(P.63, 例 3.1 より)
G1 = ⟨P1, E⟩
P1 = { E → E+T | T
T → T*F | F
F → (E) | *i* }

$i+i*i$ の最右導出（最右の非終端記号 を見つけるための注目点・と終端を表す $ を右につける）



※ ⇓ ：導出(生成規則の適用) … 注目点の左で行う
→ ：注目点の移動を表す
太字 ：最右の非終端記号

その逆（= LR 構文解析）（解析対象の文字列を・と $ ではさんで, 逆順の操作(還元)を行う）



※ → ：シフト(注目点の移動) ＝ 1 文字読み込み
⇓ ：(生成規則による)還元 … 注目点の左で起こる

※ 還元時には, 非終端記号に対応する構文木を, 順次生成して行く.
※ 構文木は下から上へ生成されていく ＝ 上向き構文解析

最右導出（注目点移動をともなう）と LR 構文解析の関係

注目点移動をともなう最右導出	⇔	LR 構文解析
注目点	⇔	読み込みポインタ
注目点の左側	⇔	処理中の記号列（終端・非終端記号）
注目点の右側	⇔	読み込み前の記号列（終端記号のみ）
注目点の移動	⇔	シフト（1 字読み込み）
導出	⇔	還元

LR 構文解析の基本的な流れ

入力列 x に対して，$\cdot x\$$ からスタートし，シフトか還元を繰り返して $S\cdot\$$（S は出発記号）となれば，成功して終了．その際に，同時に構文木も生成する．

シフト

注目点（= 読み込みポインタ）をひとつ右へ動かす．= 1 文字読み込む．

$$u\cdot \mathtt{a}v\$ \;\rightarrow\; u\mathtt{a}\cdot v\$ \qquad (u\in V^*,\ \mathtt{a}\in V_T,\ v\in V_T^*)$$

還元

注目点のすぐ左で生成規則（$A\rightarrow\alpha$）を逆に使って記号列を置き換える

$$u\alpha\cdot v\$ \;\Rightarrow\; uA\cdot v\$, \qquad (u,\alpha\in V^*,\ A\in V_N,\ v\in V_T^*)$$

※ このとき，A に対応する（部分）構文木を生成する．
（構文木の下からの生成 = 上向き構文解析）

注

途中の $u\cdot v\$$ という形式は構文解析の各時点での「状態」を表わす．

u ：処理中の記号列（$\in V^*$）– 既知

v ：読み込み前の記号列（$\in V_T^*$）– 未知

シフトするか還元するか？

構文解析の各時点（$u\cdot v\$$）で：

- シフトをするか還元するか
- どの生成規則で還元するか

を決めなければならない．

しかし，実際には，可能な選択肢（ある v に対して $S\cdot\$\Rightarrow u\cdot v\$\Rightarrow\cdot x\$$ なる最右導出が存在するような選択肢）は，u に依存してある程度限られる．

既知の u に対する可能な選択肢（ある v に対して $S\cdot\$\Rightarrow u\cdot v\$\Rightarrow\cdot x\$$ なる最右導出が存在するような選択肢）を得るためには，ある種のオートマトン（正準オートマトン）を生成してそれを用いる．（正準オートマトンの作り方は text を参照すること）

具体的には，その正準オートマトン に u を入力し，到達した状態によって，可能な選択肢が決定される．

各時点で，毎回，u を正準オートマトンに入力するのは効率が悪いなぜなら：

- シフトの時は，u が 1 文字分伸びるだけなので直前の状態から 1 回分だけ状態遷移すれば済む．

- 還元の時は，u のうちの，還元されなかった前半部分（$u'\alpha \cdot v\$ \Rightarrow u'A \cdot v\$$ という還元の場合，u' に相当する部分）は還元後も変わらないので，u によって遷移した状態の系列を記憶しておけば，還元された後半部分の遷移（$u'\alpha \cdot v\$ \Rightarrow u'A \cdot v\$$ という還元の場合，A による遷移）をやり直すだけで済む．

「配置」(pp.103-105) はそのような考えに基づき，状態の系列を，入力系列（u）とともに記憶する構造体である．

実際の LR 構文解析のプログラムでは，配置と状態遷移表を用いる．（詳細はテキストを参照すること）

参考

- https://wwws.kobe-c.ac.jp/~miura/stock/LR/LR.pdf （この文書）
- https://wwws.kobe-c.ac.jp/~miura/stock/LR/LR.swf （LR 構文解析例の Flash ムービー）