course: CSC 135-01 - Computing Theory and Programming Languages

instructor: Ted Krovetz

related notes: 2022-04-07 2022-04-05-CSC135-01-LEC-parsing

Parsing, LL(1) Parsing, Pushdown Automata (PDA)

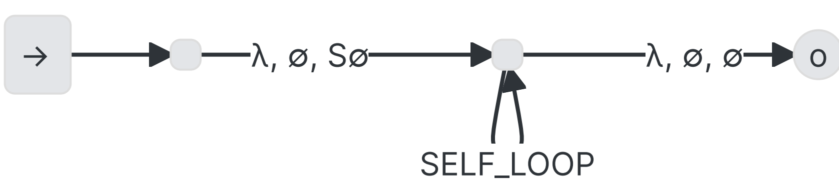
W14.4 | Thursday, April 7, 2022 | 08:58 AM

Say that we have a grammar

$$
\begin{aligned}
S &\rightarrow ABCd \\
A &\rightarrow aA \mid \lambda \\
B &\rightarrow bB \mid \lambda \\
C &\rightarrow cC \mid \lambda
\end{aligned}
$$

If we wanted to change to a Pushdown Automata (PDA)

SELF_LOOP INPUTS		
a, a, λ	λ, S, ABcd	λ, B, bB
b, b, λ	λ, A, aA	λ, B, λ
c, c, λ	λ, A, λ	λ, C, cC
d, d, λ		λ, c, λ



> TOK = next
> TOP = pop
> ⋮
> if TOP = X; and TOK ∈ first(w_i) // $X_i \rightarrow w_i$
> push(w_i)

- $X \rightarrow \lambda$: nothing
- $X \rightarrow xw$: $x \in$ first(X)
- $X \rightarrow yw$: first(Y) $\leq$ first(X)

> if Y is nullable ...
> first(w) $\leq$ first(X)

Logic	
$A \rightarrow\ aA$	
$A \rightarrow\ \lambda$	
$B \rightarrow\ bB$	
$B \rightarrow\ \lambda$	
$C \rightarrow\ cC$	
$C \rightarrow\ \lambda$	
$S \rightarrow\ ABCd$	

Improved Pushdown Automata (PDA)

Follow(A) is the set of all terminals that can appear immediately after **A** in a derivation

Methodically:

1. Find set constraints
2. Seed sets with $\in$ statements
3. Copy Left-Hand-Side (LHS) at $\leq$ statements to Right-Hand-Side (RHS)
4. Repeat Until no more changes

Patterns to look for:

Pattern	Meaning
$X \rightarrow ....Yz...$	**z $\in$ follow(Y)**
$X \rightarrow ....YZ...$	**first(Z) $\leq$ follow(Y)**
$X \rightarrow ....Y$	**follow(X) $\leq$ follow(Y)**

- $X \rightarrow ....Y$
  - Where **Y** is at the end of production

Example01 $S \rightarrow \ aSa\ |bSb|x$

1. $S \rightarrow aSa \rightarrow abSba$
    1. aSa: $a \in \ follow(S)$
    2. abSba: $b \in \ follow(S)$
2. $follow\,(S)\ =\ \{a,\ b\}$

Impossible: $....Sx..$

Example02

1. $A \rightarrow \ aA$
2. $A \rightarrow \ \lambda$
3. $B \rightarrow \ bB$
4. $B \rightarrow \ \lambda$
5. $C \rightarrow \ cC$
6. $C \rightarrow \ \lambda$
7. $S \rightarrow \ ABCd$

What is our set constraints?

$S \rightarrow \ ABCd$

- $first(b)\ \le\ follow(A)$
- $first(c)\ \le\ follow(B)$
- $d\ \in\ follow(C)$
- $d\ \in\ follow(B)$
- $d\ \in\ follow(A)$
- $first(C)\ \le\ follow(A)$

FIRST	FOLLOW
$S\; d, a, b, c$	
$A\; a$	d,b,c
$B\; b$	d,c
$C\; c$	d

What to do with these? Build prediction table

$LHS \rightarrow \ RHS$	**FIRST RHS**	**IF RHS nullable follow of LHS**	**Predictor (The union of the two other columns)**

$LHS \to\ RHS$	**FIRST RHS**	**IF RHS nullable follow of LHS**	**Predictor (The union of the two other columns)**
$A \to\ aA$	a		a
$A \to\ \lambda$		b, c, d	a, c, d
$B \to\ bB$	b		b
$B \to\ \lambda$		c, d	c, d
$C \to\ cC$	c		c
$C \to\ \lambda$		d	d
$S \to\ ABCd$	a, b, c, d		a, b, c, d

Pushdown Automata (PDA) parsing, sometimes called **LL(1) parsing**, is appropriate for this grammar if...

1. Each non-terminal group has disjoint predictors
2. Grammar is **NOT** left-recursive
3. Grammar is **NOT** ambiguous

Example03

Grammar:

$$T \to\ aTc$$
$$T \to\ R$$
$$R \to\ bR$$
$$R \to\ \lambda$$

STEP 01 - WRITE DOWN OUR CONSTRAINTS

GRAMMAR	CONSTRAINTS
$T \to\ aTc$	$a \in\ first(T)$, $c \in\ follow(T)$
$T \to\ R$	$\large $
$R \to\ bR$	$\large $
$R \to\ \lambda$	$\large $

STEP 02 - SEED SETS WITH $\in$ STATEMENTS

Take our non-terminals

STEP

GRAMMAR	CONSTRAINTS	IF RHS nullable follow of LHS	Predictor
$T \rightarrow\ aTc$	a		a
$T \rightarrow\ R$	b	c	b, c
$R \rightarrow\ bR$	b		b
$R \rightarrow\ \lambda$		c	c