TDT4205 Compilers

Exercise 3

Stian Hvatum (hvatum)
MTDT

February 21, 2012

PART 1 Theory

Task 1.1 Parsing

1.1.1 LL(k)

Even if LL(k) in theory can be extended with an $\rightarrow infinitly$ number of lookaheads, this will not resolve the problems with left recursion. For each lookahead-symbol extra, the parse table grows, and at the end it will become humongous. To create and use such a table is both space and time consuming, and we will never have neither space nor time to parse languages and grammars with arbitary use of left recursion, as it requires arbitary much space and time. Also, LL(k) needs the k to be defined, wich sets an upper bound for the number of lookaheads before we begin parsing. We may set the k to 10000000000, and hope that the language will never exeed this number of recursions, but it will not be a *valid* parser for that language, as it can only handle a subset of the possible language constructs.

1.1.2 Left-recursive grammars

The left-recursive grammar:

F $\rightarrow$ f I v A w S x
A $\rightarrow$ P
P $\rightarrow$ P I $|\epsilon$
S $\rightarrow$ S s | s
I $\rightarrow$ i

The equivalent non-left-recursive grammer:

F	→	f I v A w S x
A	→	P
P	→	I P |ϵ
S	→	s S'
S'	→	s S' |ϵ
I	→	i

1.1.3 `FIRST` and `FOLLOW` & LL(1) Parsing table

`FIRST` and `FOLLOW`

Computing `FIRST`

- f is in FIRST(F), since f is the first symbol in production of F, and f is a terminal, and thereby FIRST(f) = f.
- i is in FIRST(I), since FIRST(I) = FIRST(i) = i
- i is in FIRST(P), since FIRST(P) = FIRST(I) = i
- ϵ is in FIRST(P), since P has a production P $\rightarrow \epsilon$
- i, ϵ is in FIRST(A), since FIRST(A) = FIRST(P) = i, ϵ
- s is in FIRST(S) since FIRST(S) = FIRST(s) = s
- s is in FIRST(S') since FIRST(S') = FIRST(s) = s
- ϵ is in FIRST(S'), since S' has a production S' $\rightarrow \epsilon$

Computing `FOLLOW`

- We start by adding $ to F, since F is our *start-symbol*.
- We see from F $\rightarrow$ f I v A w S x that w is in FOLLOW(A), x is in FOLLOW(S) and v is in FOLLOW(I)
- FOLLOW(P) = FOLLOW(A) since A $\rightarrow$ P.
- FOLLOW(S') = FOLLOW(S) since S $\rightarrow$ S'.
- FOLLOW(I) includes FIRST(P) except ϵ since P $\rightarrow$ I P |ϵ, and rule 3, wich makes FOLLOW(I) = i, v

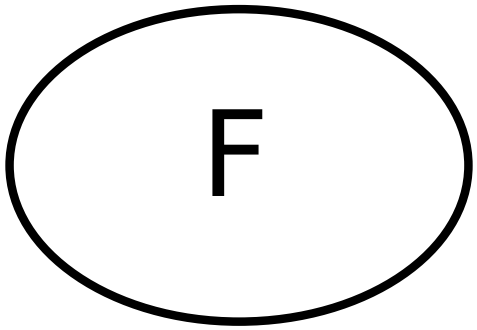
NT	FIRST	FOLLOW
F	f	$
A	i, ϵ	w
P	i, ϵ	w
S	s	x
S'	s, ϵ	x
I	i	i, v, w

LL(1) Parsing table

Followed page 224, Alg. 4.31
IF A $\rightarrow$ b AND FIRST(b) contains c THEN add A $\rightarrow$ b to M(A,c) and
IF FIRST(b) contains ϵ, THEN add A $\rightarrow \epsilon$ to M(A,c) M(A,b) M(A,c)

Non-Terminal	Input Symbol						
	f	i	v	w	s	x	$
F	F $\rightarrow$ f I v A w S x						
A		A $\rightarrow$ P		A $\rightarrow \epsilon$			
P		P $\rightarrow$ I P		P $\rightarrow \epsilon$			
S					S $\rightarrow$ sS'		
S'					S' $\rightarrow$ sS'	S' $\rightarrow \epsilon$	
I		I $\rightarrow$ i					

1.1.4 Parse tree for LL(1)



LL1-Pars