Compiler Construction
WA03: Top-Down Parsing

Terence Henriod

October 11, 2015

Abstract

This assignment asks you to prepare written answers to questions on top-down parsing. Each question has a short answer. You may discuss this assignment with other students and work the problems together. However, your writeup should be your own individual work. Remember written assignments are to be turned in in class on the date due.

Consider the following grammar for a subset of English sentences. The nonterminals are S, NP, VP, AP. The terminals are *the*, *noun*, *adjective*, *verb*. The starts symbol is S.

$$\begin{aligned} S &\to NP\ VP \mid NP\ VP\ NP \\ NP &\to \textit{the}\ AP\ \textit{noun} \\ AP &\to AP\ \textit{adjective} \mid \epsilon \\ VP &\to \textit{verb} \end{aligned}$$

Note: In order to make the notation cleaner for presentation, I will be replacing the symbols of this grammar with a single (presumably reasonable) character replacement:

Original Symbol	Replacement
S	S
NP	N
AP	A
VP	V
the	t
noun	n
adjective	a
verb	v

1. Left-factor this grammar.

   Answer:

   The only left factoring possible is done on S.

$$\begin{aligned} S &\to NVS' \\ S' &\to N \mid \epsilon \\ N &\to tAn \\ A &\to Aa \mid \epsilon \\ V &\to v \end{aligned}$$

2. Eliminate left-recursion from your answer to part (1).

   Answer:

   The only left-recursion to be eliminated is done on AP.

$$\begin{aligned} S &\to NVS' \\ S' &\to N \mid \epsilon \\ N &\to tAn \\ A &\to \epsilon A' = A' \\ A' &\to aA' \mid \epsilon \\ V &\to v \end{aligned}$$

   Note: At this point it is clear that the symbol A' could be eliminated, leaving the production $A \to aA$, and the grammar would be equivalent, but I did not remove it for the sake of being explicit.

3. Give the *First* and *Follow* sets for each grammar symbol for your answer to part (2).

Answer:

First Sets:

$$\begin{array}{|rcl|}\hline First(S \rightarrow\ NVS') & = & First(NVS') \\ & = & First(N) \\ & = & \{t\} \\ \hline First(S) & = & \{t\} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline First(S' \rightarrow N) & = & First(N) \\ & = & \{t\} \\ \hline First(S' \rightarrow \epsilon) & = & First(\epsilon) \\ & = & \{\epsilon\} \\ \hline First(S') & = & \{t\} \cup \{\epsilon\} \\ & = & \{t,\ \epsilon\} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline First(N \rightarrow tAn) & = & First(tAn) \\ & = & First(t) \\ & = & \{t\} \\ \hline First(N) & = & \{t\} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline First(A \rightarrow A') & = & First(A') \\ & = & \{a,\ \epsilon\} \\ \hline First(A) & = & \{a,\ \epsilon\} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline First(A' \rightarrow aA') & = & First(aA')) \\ & = & First(a) \\ & = & \{a\} \\ \hline First(A' \rightarrow \epsilon) & = & First(\epsilon) \\ & = & \{\epsilon\} \\ \hline First(A') & = & \{a\} \cup \{\epsilon\} \\ & = & \{a,\ \epsilon\} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline First(V \rightarrow v) & = & First(v) \\ & = & \{v\} \\ \hline First(V) & = & \{v\} \\ \hline \end{array}$$

Follow Sets:

$$\begin{array}{|rcl|}\hline Follow(S : start\ symbol) & = & \{\$\} \\ \hline Follow(S) & = & \{\$\} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline Follow(S' : S \rightarrow NVS') & = & Follow(S) \\ & = & \{\$\} \\ \hline Follow(S') & = & \{\$\} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline Follow(N : S \to NVS') & = & First(VS') \\ & = & First(V) \\ & = & \{v\} \\ \hline Follow(N : S' \to N) & = & Follow(S') \\ & = & \{\$\} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline Follow(N) & = & \{v\} \cup \{\$\} \\ & = & \{v,\ \$\} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline Follow(A : N \to tAn) & = & First(n) \\ & = & \{n\} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline Follow(A) & = & \{n\} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline Follow(A' : A \to A') & = & Follow(A) \\ & = & \{n\} \\ \hline Follow(A' : A' \to aA') & = & Follow(A') \\ & = & \textit{to be resolved} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline Follow(A') & = & \{n\} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline Follow(V : S \to NVS') & = & (First(S') - \{\epsilon\}) \cup Follow(S) \\ & = & \{t\} \cup \{\$\} \\ & = & \{t,\ \$\} \\ \hline \end{array}$$

$$\begin{array}{|rcl|}\hline Follow(V) & = & \{t,\ \$\} \\ \hline \end{array}$$

4. Give an LL(1) parsing table for your answer to part (2).

   Answer:

	t	n	a	v	$\$$
S	$S \to NVS'$				
S'	$S' \to N$				$S' \to \epsilon$
N	$N \to tAn$				
A			$A \to A'$		
A'		$A' \to \epsilon$	$A' \to aA'$		
V				$V \to v$	

5. Show the sequence of moves of your LL(1) parser on the input:
   The silly professor invented the awkward obscure sentence.
   (Note that you need to tokenize the input first.)

   Answer:

   Assumption: The '.' character can be replaced with \$.

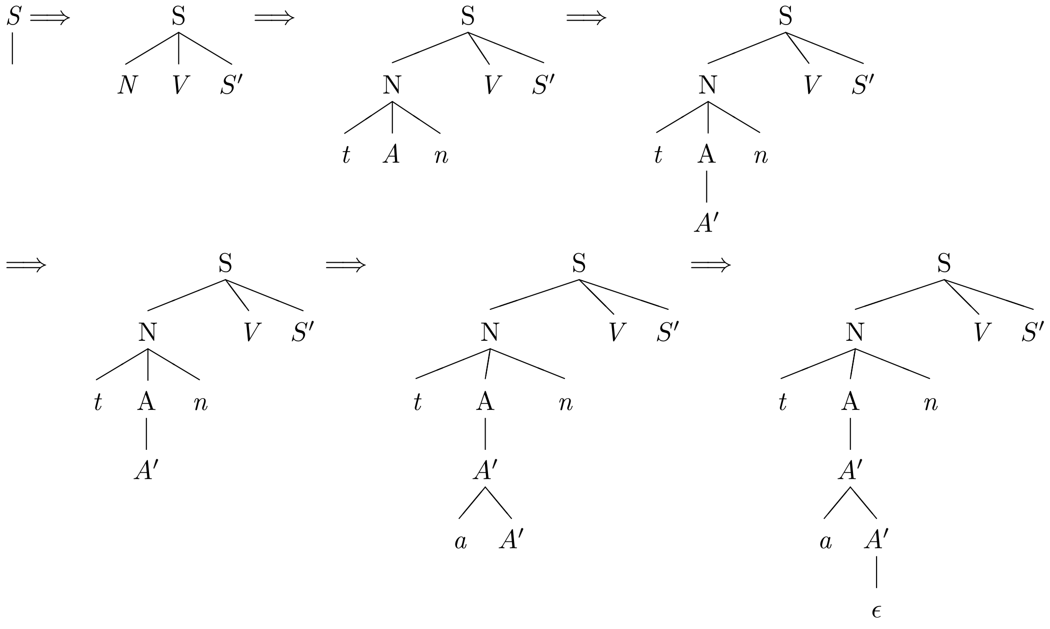
   Tokens: $\{The,\ silly,\ professor,\ invented,\ the,\ awkward,\ obscure,\ sentence,\ \$\}$ becomes $\{t,\ a,\ n,\ v,\ t,\ a,\ a,\ n,\ \$\}$

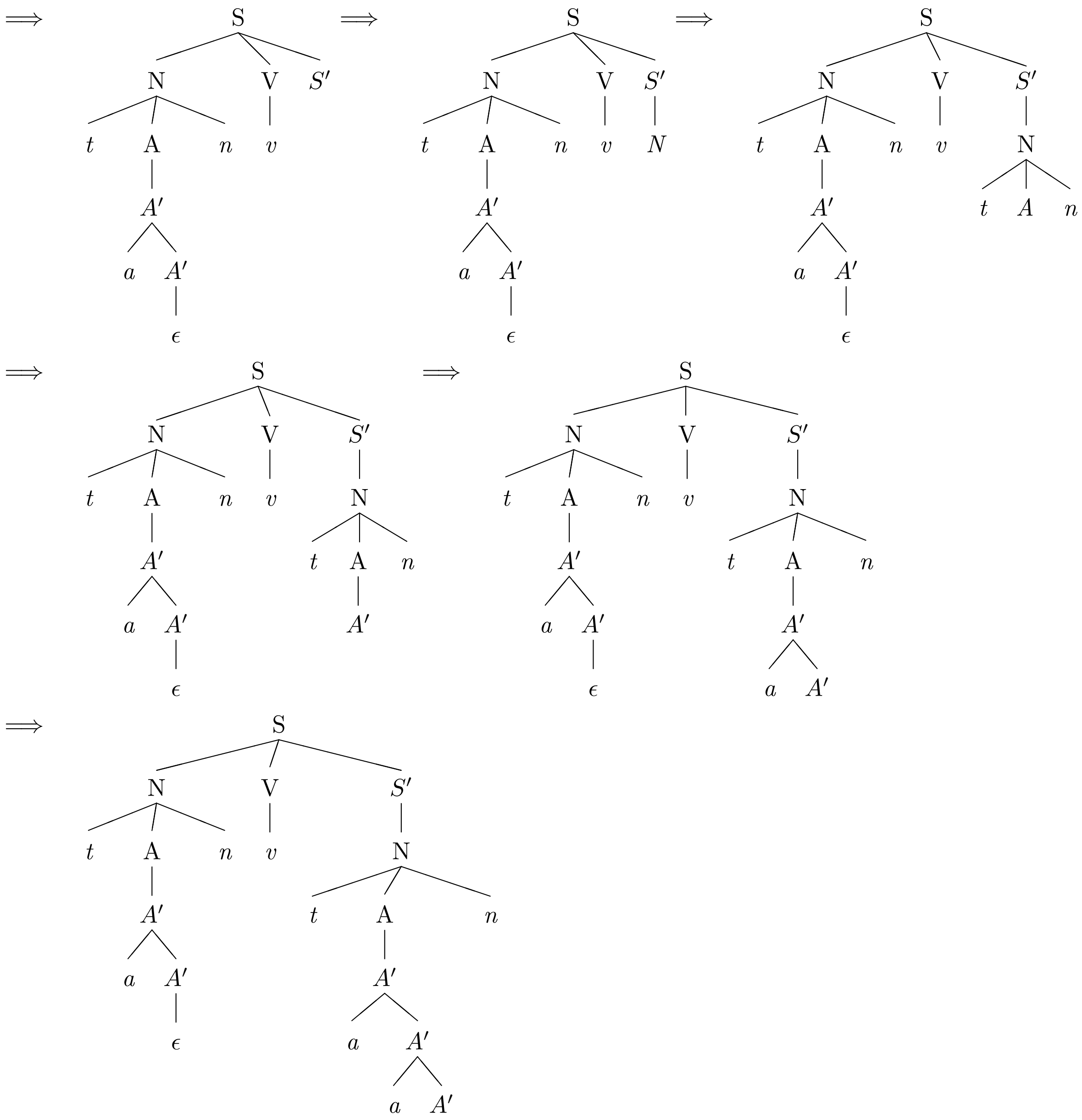
Stack	Input	Production	Derivation
$\$S$	$tanvtaan\$$	$S \to NVS'$	S
$\$S'VN$	$tanvtaan\$$	$N \to tAn$	NVS'
$\$S'VnAt$	$tanvtaan\$$	[pop and go to next token]	$tAnVS'$
$\$S'VnA$	$anvtaan\$$	$A \to A'$	$tA'nVS'$
$\$S'VnA'$	$anvtaan\$$	$A^{prime} \to aA'$	$taA'nVS'$
$\$S'VnA'a$	$anvtaan\$$	[pop and go to next token]	
$\$S'VnA'$	$nvtaan\$$	$A' \to \epsilon$	$tanVS'$
$\$S'Vn$	$nvtaan\$$	[pop and go to next token]	
$\$S'V$	$vtaan\$$	$V \to v$	$tanvS'$
$\$S'v$	$vtaan\$$	[pop and go to next token]	
$\$S'$	$taan\$$	$S' \to N$	$tanvN$
$\$N$	$taan\$$	$N \to tAn$	$tanvtAn$
$\$nAt$	$taan\$$	[pop and go to next token]	
$\$nA$	$aan\$$	$A \to A'$	$tanvtA'n$
$\$nA'$	$aan\$$	$A' \to aA'$	$tanvtaA'n$
$\$nA'a$	$aan\$$	[pop and go to next token]	
$\$nA'$	$an\$$	$A' \to aA'$	$tanvtaaA'n$
$\$nA'a$	$an\$$	[pop and go to next token]	
$\$nA'$	$n\$$	$A' \to \epsilon$	$tanvtaan$
$\$n$	$n\$$	[pop and go to next token]	
$\$$	$\$$	[pop and go to next token]	
		Successful parse.	

6. Show the parse tree that your parser traces for this sentence.

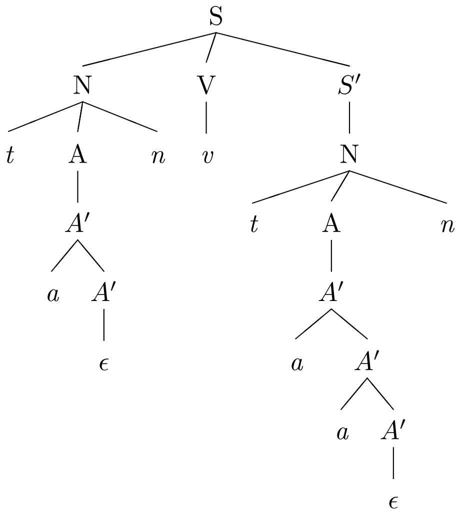
   Answer:

   (See the “Derication” column in the table in the previous exercise to see the string representations of the trees, sentential and otherwise.)




⟹
S
N
V
S′
t
A
n
v
A′
a
A′
ϵ
N
t
A
n
a
A′
a
A′

(Finally) $\Longrightarrow$


S
N
V
S'
t
A
n
v
N
A'
t
A
n
a
A'
A'
ϵ
a
A'
a
A'
ϵ