CS4124

Answers to Selected Problems of HW #5

Problem 3.2.2

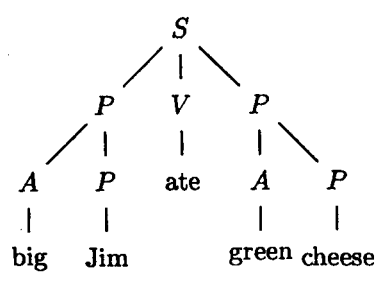
Two distinct leftmost derivations of () are given by

1. $S \Rightarrow SS \Rightarrow (S)S \Rightarrow ()S \Rightarrow ()$
2. $S \Rightarrow (S) \Rightarrow ()$

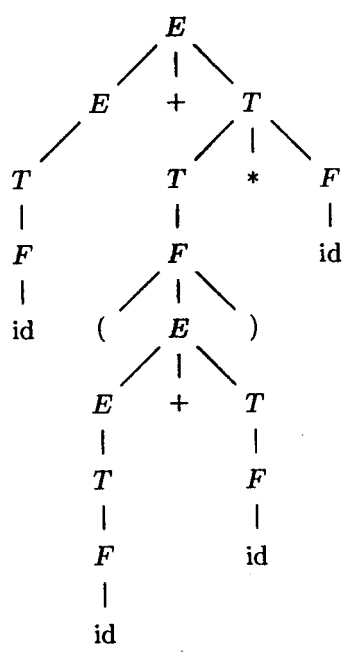
The problem with this grammar is that the rule $S \to e$ allows us to produce superfluous Ss which then disappear. Let us replace this grammar with $G' = (\{(,), S, T\}, \{(,)\}, R, S)$, where

$$\begin{aligned} R \;=\; & \{S \to T \\ & S \to e \\ & T \to () \\ & T \to (T) \\ & T \to TT\}. \end{aligned}$$

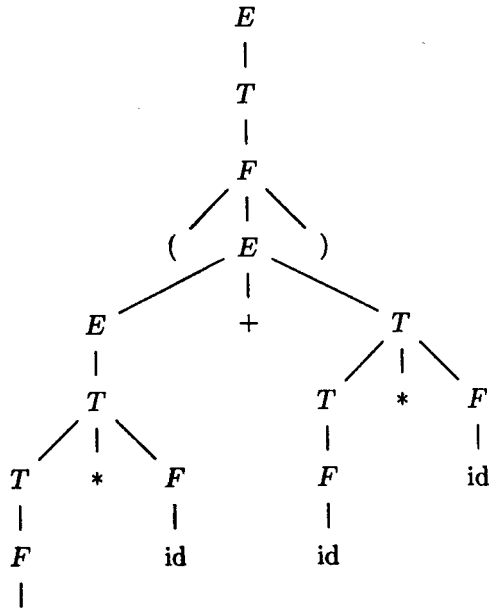
Problem 3.2.4



(a)



(b)



(c) id

Problem 3.3.2

(a) $M = (K, \Sigma, \Gamma, \Delta, s, F)$, where

$$\begin{aligned}
K &= \{s\} \\
\Sigma &= \{(,),[,],\} \\
\Gamma &= \{(,[\} \\
F &= \{q\} \\
\Delta &= \{((q,(,e)(q,))), \\
&\quad ((q,),(),(q,e)), \\
&\quad ((q,[,e),(q,[)), \\
&\quad ((q,],[),(q,e))\}.
\end{aligned}$$

(b) $M = (K, \Sigma, \Gamma, \Delta, q, F)$, where

$$\begin{aligned}
K &= \{q,r\} \\
\Sigma &= \{a,b\} \\
\Gamma &= \{a\} \\
F &= \{r\} \\
\Delta &= \{((q,a,e),(q,aa)), \\
&\quad ((q,a,e),(q,a)), \\
&\quad ((q,e,e),(r,e)), \\
&\quad ((r,b,a),(r,e))\}.
\end{aligned}$$

(c) $M = (K, \Sigma, \Gamma, \Delta, s, F)$, where

$$\begin{aligned}
K &= \{q, r\} \\
\Sigma &= \{a, b\} \\
\Gamma &= \{a, b\} \\
F &= \{r\} \\
\Delta &= \{((q, a, e), (q, a)), \\
&\quad ((q, b, e), (q, b)), \\
&\quad ((q, e, e), (r, e)), \\
&\quad ((q, a, e), (r, e)), \\
&\quad ((q, b, e), (r, e)), \\
&\quad ((r, a, a), (r, e)), \\
&\quad ((r, b, b), (r, e))\}
\end{aligned}$$

(d) $M = (K, \Sigma, \Gamma, \Delta, s, F)$, where

$$\begin{aligned}
K &= \{q\} \\
\Sigma &= \{a, b\} \\
\Gamma &= \{A, a, b\} \\
F &= \{q\} \\
\Delta &= \{((q, a, e), (q, A)), \\
&\quad ((q, b, e), (q, b)), \\
&\quad ((q, a, b), (q, a)), \\
&\quad ((q, b, A), (q, a)), \\
&\quad ((q, b, a), (q, e))\}
\end{aligned}$$

Problem 3.3.3

(a) Given a pushdown automaton $M = (K, \Sigma, \Gamma, \Delta, s, F)$, create another pushdown automaton $M' = (K \cup \{f\}, \Sigma, \Gamma, \Delta', s, \{f\})$, where

$$\begin{aligned}
\Delta' &= \Delta \\
&\quad \cup \{((q, e, e), (f, e)) : q \in F\} \\
&\quad \cup \{((f, e, \alpha), (f, e)) : \alpha \in \Gamma\}.
\end{aligned}$$

Then $L_f(M) = L(M')$. M' acts just like M except that any point when M is in a final state, M' is able to e-transition to the special final state f, from which it is allowed to clear its stack. If M was able to accept by final state, then it had scanned all its input and M' will be able to accept normally, and vice-versa.

(b) Given a pushdown automaton $M = (K, \Sigma, \Gamma, \Delta, s, F)$, create another pushdown automaton $M'' = (K \cup \{s'', f''\}, \Sigma, \Gamma \cup \{\perp\}, \Delta', s'', \{f''\})$, where

$$\begin{aligned}
\Delta'' &= \Delta \\
&\quad \cup \{((s'', e, e), (s, \perp))\} \\
&\quad \cup \{((q, e, \perp), (f'', e)) : q \in F\}.
\end{aligned}$$

Then $L_f(M'') = L(M)$. M'' acts just like M except that it is always carrying a $\perp$ at the bottom of its stack. The only way M'' can get into a final state is by clearing it on a transition from a state in F. But unless M reached that state with empty stack, the $\perp$ will not be on top of the stack of M'', so that it can only enter

its final state on an empty stack. On the other hand, any accepting computation of M can be turned into an accepting computation of M'' by adding one step at the start to put $\perp$ on the stack and one at the end to remove it.

Problem 3.3.4

(a) Given a pushdown automaton $M = (K, \Sigma, \Gamma, \Delta, s, F)$, create another pushdown automaton $M' = (K \cup \{S', f'\}, \Sigma, \Gamma \cup \{\perp\}, \Delta', s', \{f'\}$, where

$$\begin{aligned} \Delta' &= \Delta \\ &\quad \cup\{((s', e, e), (s, \perp))\} \\ &\quad \cup\{((q, e, \perp), (f', e)) : q \in F\}. \end{aligned}$$

Then $L_e(M') = L(M)$. M' is only able to clear its stack from a final state of M, so in order to reach a situation of empty store, it must have passed through an accepting configuration in its simulation of M. On the other hand, in any situation when M could have accepted, its stack is empty and it is in a final state, so that M' sees $\perp$ and is in an appropriate state to clear that $\perp$ and then accept.

(b) Given a pushdown automaton $M = (K, \Sigma, \Gamma, \Delta, s, F)$, create another pushdown automaton $M'' = (K \cup \{s'', f''\}, \Sigma, \Gamma \cup \{\perp\}, \Delta'', s'', \{f''\})$, where

$$\begin{aligned} \Delta'' &= \Delta \\ &\quad \cup\{((s'', e, e), (s, \perp))\} \\ &\quad \cup\{((q, e, \perp), (f'', e)) : q \in K\} \end{aligned}$$

Then $L(M'') = L_e(M)$. Whenever M is in a position to accept by empty store, then M'' will be able to pop off the $\perp$, and enter a final state. On the other hand, M'' cannot enter that state unless M has reached an empty stack and is thus ready to accept.

Problem 3.4.1

The new machine is $M = (\{p, q\}, \{(,)\}, \{(,), S\}, \Delta, p, \{q\})$, where

$$\Delta = \{((p, e, e), (q, S)), ((q, e, S), (q, SS)), ((q, e, S), (q, (S))), ((q, e, S), (q, e)), ((q, (, (), (q, e)), ((q,),)), (q, e))\}$$

Then

$$\begin{aligned} (p, (()()), e) &\vdash_M (q, (()()), S) \\ &\vdash_M (q, (()()), (S)) \\ &\vdash_M (q, ()()), S)) \\ &\vdash_M (q, ()()), SS)) \\ &\vdash_M (q, ()()), (S)S)) \\ &\vdash_M (q,)()), S)S)) \\ &\vdash_M (q,)()),)S)) \\ &\vdash_M (q, ()), S)) \\ &\vdash_M (q, ()), (S))) \\ &\vdash_M (q,)), S))) \\ &\vdash_M (q,)),))) \\ &\vdash_M (q,),)) \\ &\vdash_M (q, e, e) \end{aligned}$$

The Last Problem

(a)

$S \rightarrow [S, \#, S]$, $S \rightarrow [S, \#, q]$
$[S, \#, S] \rightarrow a\,[S, 0, S]\;[S, \#, S]$,
$[S, 0, S] \rightarrow a\,[S, 0, S]\;[S, 0, S]$,
$[S, 0, S] \rightarrow [q, 0, q]$ (also $[S, 0, q]$ not used $[q, 0, q]$, it is not used here)
$[S, \#, S] \rightarrow [q, \#, q]$

$[q, 0, q] \rightarrow b$
$[q, \#, q] \rightarrow e$

mirror rules for $[S, \#, q]$

(b)

$S \rightarrow [S, \#, S]$
$\rightarrow a\,[S, 0, S]\,[S, \#, S]$
$\rightarrow a\,a\,[S, 0, S]\,[S, 0, S]\,[S, \#, S]$
$\rightarrow a\,a\,[q, 0, q]\,[S, 0, S]\,[S, \#, S]$
$\rightarrow a\,a\,b\,[S, 0, S]\,[S, \#, S]$
$\rightarrow a\,a\,b\,b\,[S, \#, S]$
$\rightarrow a\,a\,b\,b$