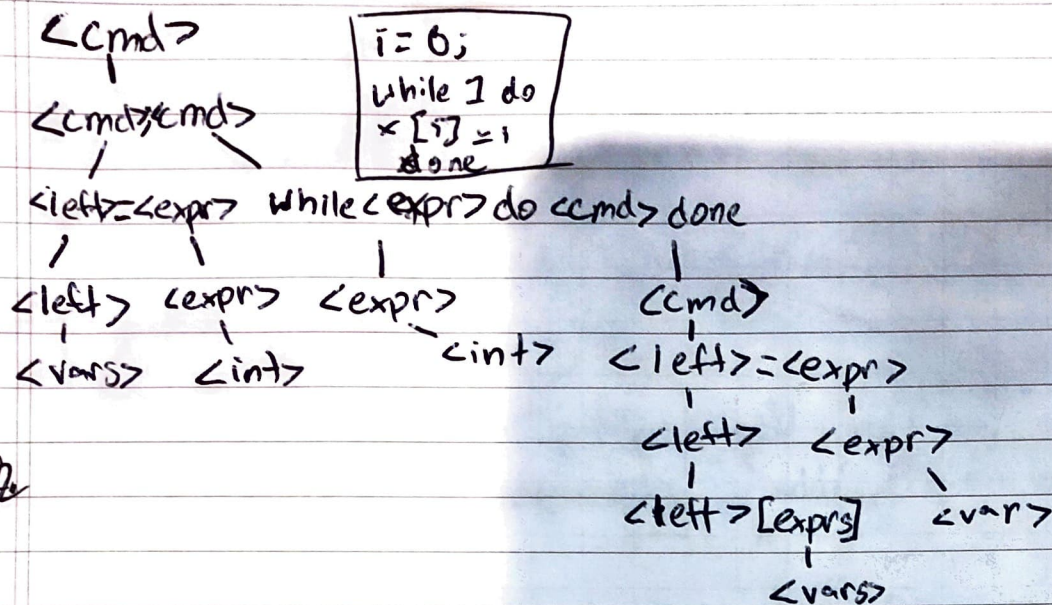


1. This grammar is unambiguous because you can't make more than one parse tree from one expression. No



12

$i = 0;$

if $i = 0$

then skip else $x[i]$

Not accepted by the grammar because "if $i = 0$ " is a $\text{if } \langle \text{expr} \rangle \text{ then } \langle \text{cmd} \rangle$ but a $\langle \text{cmd} \rangle$ only takes $\langle \text{expr} \rangle$, and this is another $\langle \text{cmd} \rangle$ not a var or a int. "if $i = 0$ " doesn't work.

2. ab^*c^*E

3. a^*bb^*cE

4. $\langle A \rangle ::= a \langle B \rangle$

$\mid c \langle C \rangle$
 $\mid E$

$\langle B \rangle ::= b \langle A \rangle$

$\mid b \langle C \rangle$

$\langle C \rangle ::= c \langle C \rangle$

$\mid E$

$$5. \text{add}(1,5) \rightarrow 6$$

$$\text{add}(1, \text{add}(2,3)) \rightarrow \text{add}(1,5)$$

$$\text{add}(1, \text{add}(2,3)) \rightarrow 2 \ 6$$

~~$$\text{eq}(3, \text{add}(2,1)) \rightarrow \text{eq}(3,3)$$~~
~~$$\text{eq}(3, \text{add}(2,1)) \rightarrow \text{eq}(3,3)$$~~
~~$$\text{eq}(3, \text{add}(2,1)) \rightarrow \text{eq}(3,3)$$~~

$$\text{eq}(3,3) \rightarrow \text{true}$$

$$\text{eq}(3, \text{add}(2,1)) \rightarrow \text{eq}(3,3)$$

$$\text{eq}(\text{add}(1,2), \text{add}(2,1)) \rightarrow \text{eq}(3, \text{add}(2,1))$$

$$\text{eq}(\text{add}(1,2), \text{add}(2,1)) \rightarrow \text{eq}(\text{add}(1,2), 3)$$

$$\text{eq}(\text{add}(1,2), \text{add}(2,1)) \rightarrow \text{true}$$

$$\text{add}(3, \text{false}) \rightarrow \text{error}$$

$$\text{add}(\text{add}(1,2), \text{false}) \rightarrow \text{add}(3, \text{false})$$

$$\text{add}(\text{add}(1,2), \text{eq}(1,2)) \rightarrow \text{add}(\text{add}(\text{add}(1,2), \text{false}))$$

$$\text{add}(3, \text{eq}(1,2)) \rightarrow \text{add}(3, \text{false})$$

$$\text{add}(\text{add}(1,2), \text{eq}(1,2)) \rightarrow \text{add}(3, \text{eq}(1,2))$$

$$\text{add}(\text{add}(1,2), \text{eq}(1,2)) \rightarrow \text{error}$$

$$6a) (\text{add}; \text{quit}/2::1::\mathbb{C}) \rightarrow (\text{quit}/3::\mathbb{C})$$

$$(\text{push } 2; \text{add}; \text{quit}/1::\mathbb{C}) \rightarrow (\text{add}; \text{quit}/2::1::\mathbb{C})$$

$$(\text{push } 1; \text{push } 2; \text{add}; \text{quit}/\mathbb{C}) \rightarrow (\text{push } 2; \text{add}; \text{quit}/1::\mathbb{C})$$

$$(\text{push } 1; \text{push } 2; \text{add}; \text{quit}/\mathbb{C})$$

$$b) (\text{sub}; \text{quit}/2::1::\mathbb{C}) \rightarrow (\text{quit}/-1::\mathbb{C})$$

$$(\text{push } 2; \text{sub}; \text{quit}/1::\mathbb{C}) \rightarrow (\text{sub}; \text{quit}/2::1::\mathbb{C})$$

$$(\text{push } 1; \text{push } 2; \text{sub}; \text{quit}/\mathbb{C}) \rightarrow (\text{push } 2; \text{sub}; \text{quit}/1::\mathbb{C})$$

$$(\text{push } 1; \text{push } 2; \text{sub}; \text{quit}/\mathbb{C})$$

$$c) (\text{pop}; \text{quit}/\mathbb{C}) \rightarrow \text{error}$$

$$(\text{if push else pop and}; \text{quit}/-1::\mathbb{C}) \rightarrow (\text{pop}; \text{quit}/\mathbb{C})$$

$$(\text{sub}; \text{if push else pop end}; \text{quit}/2::1::\mathbb{C}) \rightarrow (\text{if push else pop end}; \text{quit}/-1::\mathbb{C})$$

$$(\text{push } 2; \text{sub}; \text{if push else pop end}; \text{quit}/1::\mathbb{C}) \rightarrow (\text{sub}; \text{if push else pop end}; \text{quit}/2::1::\mathbb{C})$$

$$(\text{push } 1; \text{push } 2; \text{sub}; \text{if push else pop end}; \text{quit}/\mathbb{C}) \rightarrow (\text{push } 2; \text{sub}; \text{if push else pop end}; \text{quit}/1::\mathbb{C})$$

$$(\text{push } 1; \text{push } 2; \text{sub}; \text{if push else pop end}; \text{quit}/\mathbb{C})$$