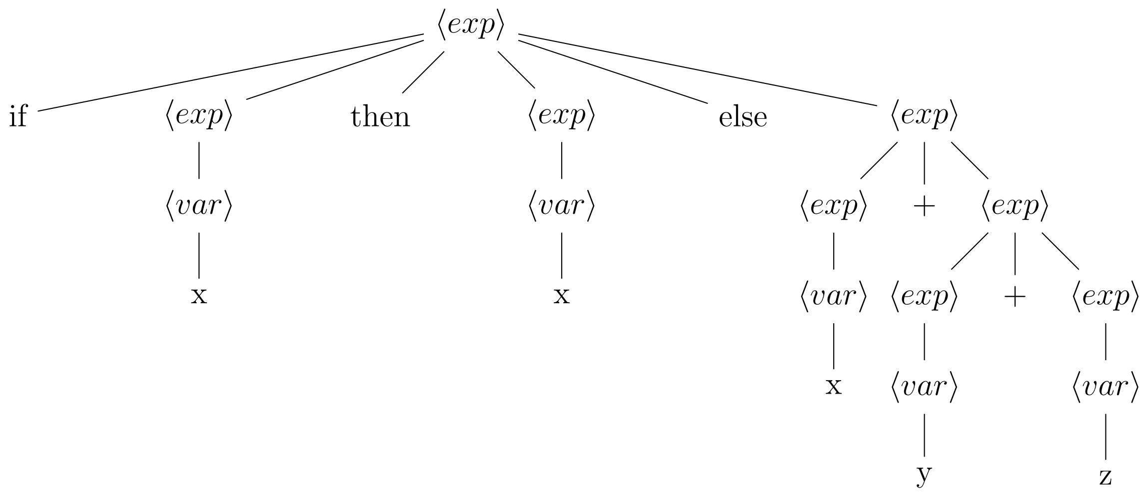
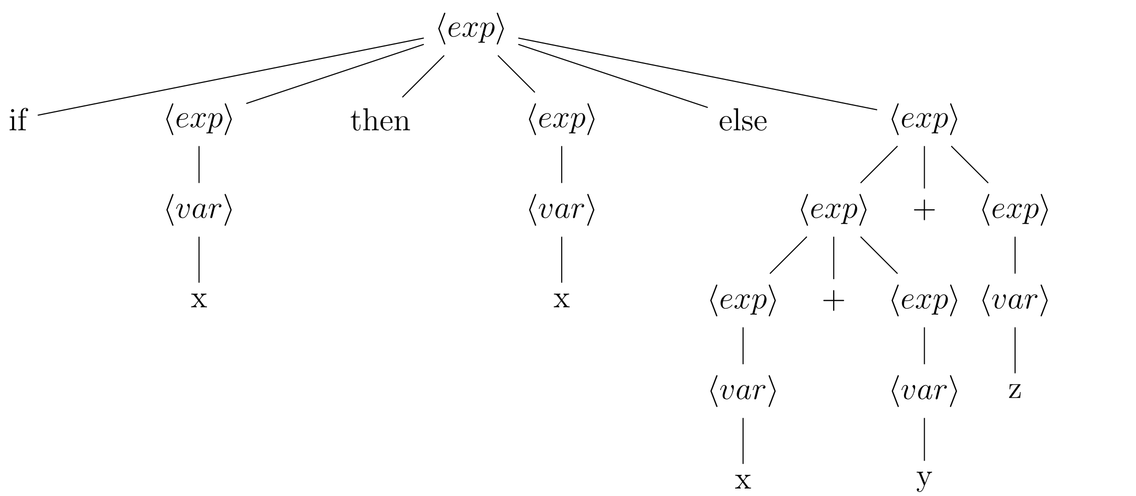
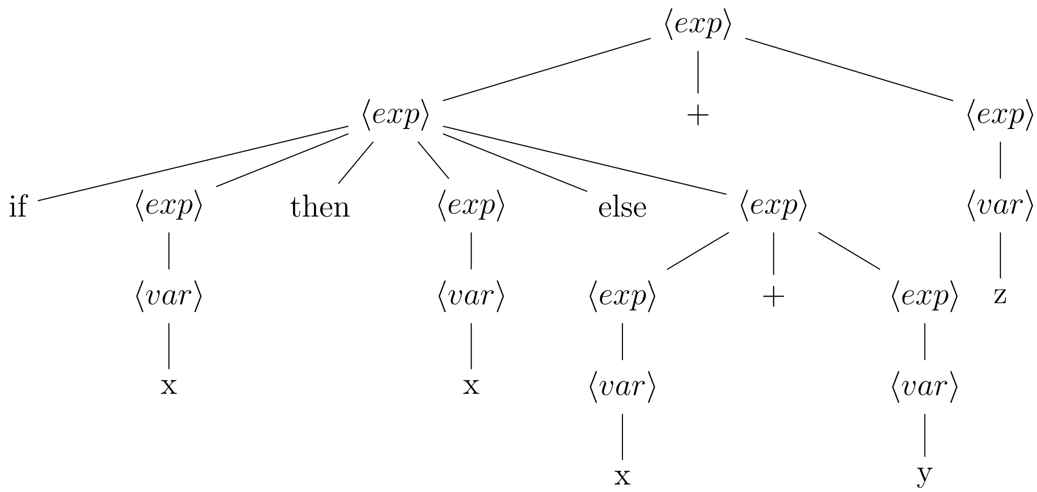
CS421 : hw7	**Summer 2015**
Dan McQuillan	*Handed In: July 27, 2015*

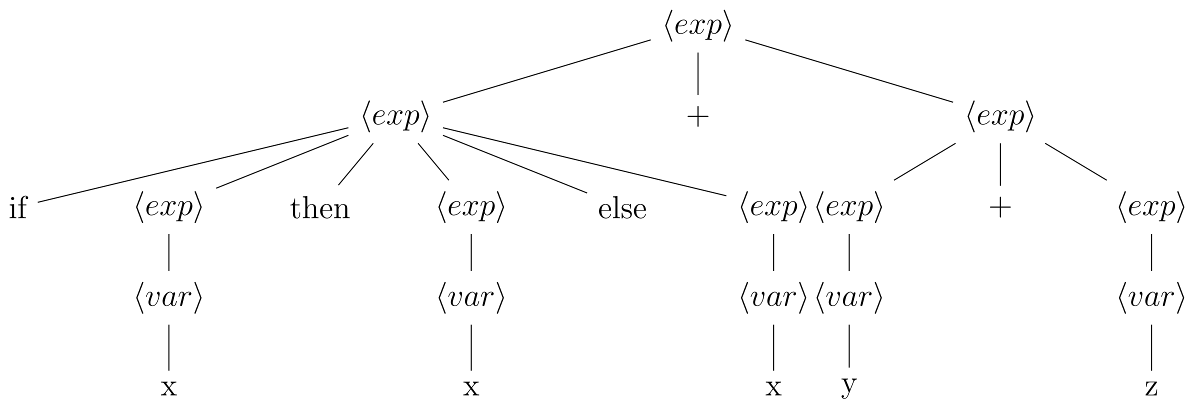
1 Consider the following grammar over the alphabet if, then, else, +, x, y, z, (,):

a Show that the above grammar is ambiguous by showing at least three distinct parse trees for the string ”if x then x else x + y + z”





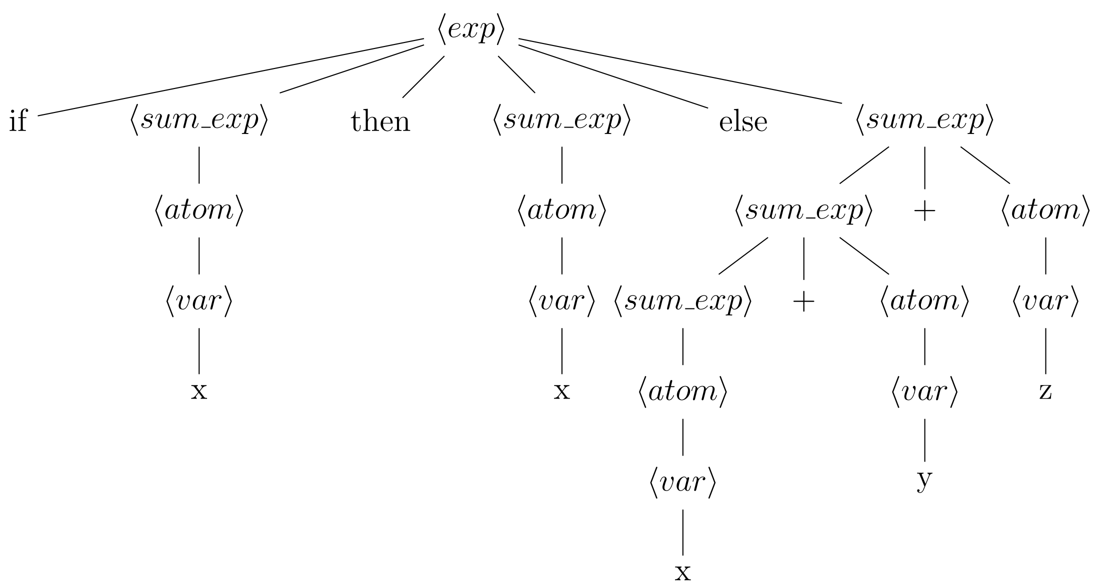




b Write a new grammar accepting the same language that is unabmiguous, and such that addition $\langle exp\rangle + \langle exp\rangle$ has higher precedence than conditional.

$$
\begin{aligned}
\langle exp\rangle &::= \text{if } \langle sum\_exp\rangle \text{ then } \langle sum\_exp\rangle \text{ else } \langle sum\_exp\rangle && (1)\\
&\;\;|\langle sum\_exp\rangle && (2)\\
\langle sum\_exp\rangle &::= \langle sum\_exp\rangle + \langle atom\rangle && (3)\\
&\;\;|\langle atom\rangle && (4)\\
\langle atom\rangle &::= \langle var\rangle && (5)\\
&\;\;|(\langle exp\rangle) && (6)\\
\langle var\rangle &::= x|y|z && (7)
\end{aligned}
$$

c Give the parse tree for ”f x then x else x + y + z” using the grammar you gave in the previous part of this problem.



2 Add a new increment operator ++I to the syntax of expression E and a new do-while operator do C while B od to the syntax of commands C

$I \in$ Identifiers
$N \in$ Numerals
$B ::=$ true | false $|B \;\&\; B|B$ or $B|$ not $B|E < E|E = E$
$E ::= N|I| + +I|E + E|E * E|E - E| - E$
$C ::=$ skip$|C;C|I ::= E|$ if B then C else C fi | while B do C od | do C while B od

a. Add the structural operational semantics (a.k.a. natural semantics) for these operators. Note that the operators work as follows. The semantics of the operator ++I is to add one to the current value of I, then store the new value into I. The execution of do C while B od starts with executing the command C in the body of the loop. The loop is repeated until the boolean expression B is evaluated to false.

Natural Semantics:

++I:

$$\frac{v = m(I) + 1}{(+ + I, m) \Downarrow (v, m[I \to v])}$$

do C while B od:

$$\frac{(C, m) \Downarrow m' \quad (\text{ while } B \text{ do } C \text{ od }, m') \Downarrow m''}{(\text{do } C \text{ while } B \text{ od }, m) \Downarrow m''}$$

b. Add the transition semantics for these operators. They have the same meaning as part a.

Transition Semantics:

++I:

$$\frac{v = m(I) + 1}{(+ + I, m) \to (v, m[I \to v])}$$

do C while B od:

$$(\text{do } C \text{ while } B \text{ od }, m) \to (C; \text{ while } B \text{ do } C \text{ od }, m)$$

c. Using the rules given for natural semantics in class, and the rules written in parts a and b, give a proof that starting with a memory that maps x to 3, do y ::= ++x while x ¡ 5 od evaluates to a memory that maps x and y to 5.

Natural Semantics Proof:

Proof:

let $m' = \{x \to 4\}$
let $m'' = \{x \to 4, y \to 4\}$
let $m''' = \{x \to 5, y \to 4\}$
let $m'''' = \{x \to 5, y \to 5\}$

$$\frac{\overbrace{(y ::= + + x, m) \Downarrow m'[y \to 4]}^{e_1} \quad \overbrace{(\text{ while } x < 5 \text{ do } y ::= + + x \text{ od }, m'') \Downarrow m''''}^{e_2}}{(\text{ do } y ::= + + x \text{ while } x < 5 \text{ od }, m) \Downarrow m''''}$$

$$\dfrac{\dfrac{m(x) + 1 = 4}{(+ + x, m) \Downarrow (4, m[x \to 4])}}{\underbrace{(y ::= + + x, m) \Downarrow m'[y \to 4]}_{e_1}}$$

$$\dfrac{\dfrac{\dfrac{4 < 5 = \text{ true}}{(x, m'') \Downarrow 4 \quad (5, m'') \Downarrow 5}}{(x < 5, m'') \Downarrow \text{ true}} \quad \dfrac{\dfrac{m(x) + 1 = 5}{(+ + x, m'') \Downarrow (5, m''[x \to 5])}}{(y ::= + + x, m'') \Downarrow m'''[y \to 5]} \quad \dfrac{\dfrac{\dfrac{5 < 5 = \text{ false}}{(x, m'''') \Downarrow 5 \quad (5, m'''') \Downarrow 5}}{(x < 5, m'''') \Downarrow \text{ false}}}{(\text{ while } x < 5 \text{ do } y ::= + + x \text{ od }, m'''') \Downarrow m''''}}{\underbrace{(\text{ while } x < 5 \text{ do } y ::= + + x \text{ od }, m'') \Downarrow m''''}_{e_2}}$$

■

Transition Semantics Proof:

Proof:

let $m' = \{x \to 4\}$
let $m'' = \{x \to 4, y \to 4\}$
let $m''' = \{x \to 5, y \to 4\}$
let $m'''' = \{x \to 5, y \to 5\}$

$$(\text{ do } y ::= ++x \text{ while } x < 5 \text{ od }, m) \to$$

$$\cfrac{\cfrac{4 = m(x) + 1}{(++x, m) \to (4, m[x \to 4])}}{\to (y ::= ++x;\ \text{while } x < 5 \text{ do } y ::= ++x \text{ od }, m)}$$

$$\cfrac{(y ::= 4, m') \to m'[y \to 4]}{\to (y ::= 4;\ \text{while } x < 5 \text{ do } y ::= ++x \text{ od }, m')}$$

$$\to (\text{ while } x < 5 \text{ do } y ::= ++x \text{ od }, m'')$$

$$\cfrac{(x, m'') \to (4, m'')}{\to (\text{ if } x < 5 \text{ then } y ::= ++x;\ \text{while } x < 5 \text{ do } y ::= ++x \text{ od } \text{ else } \text{ skip }, m'')}$$

$$\cfrac{\text{true } = 4 < 5}{\to (\text{ if } 4 < 5 \text{ then } y ::= ++x;\ \text{while } x < 5 \text{ do } y ::= ++x \text{ od } \text{ else } \text{ skip }, m'')}$$

$$\to (\text{ if } \text{ true } \text{ then } y ::= ++x;\ \text{while } x < 5 \text{ do } y ::= ++x \text{ od } \text{ else } \text{ skip }, m'')$$

$$\cfrac{\cfrac{5 = m''(x) + 1}{(++x, m'') \to (5, m''[x \to 5])}}{\to (y ::= ++x;\ \text{while } x < 5 \text{ do } y ::= ++x \text{ od }, m'')}$$

$$\cfrac{(y ::= 5, m''') \to m'''[y \to 5]}{\to (y ::= 5;\ \text{while } x < 5 \text{ do } y ::= ++x \text{ od }, m''')}$$

$$\to (\text{ while } x < 5 \text{ do } y ::= ++x \text{ od }, m'''')$$

$$\cfrac{(x, m'''') \to (5, m'''')}{\to (\text{ if } x < 5 \text{ then } y ::= ++x;\ \text{while } x < 5 \text{ do } y ::= ++x \text{ od } \text{ else } \text{ skip }, m'''')}$$

$$\frac{5 < 5 = \text{ false}}{\rightarrow (\text{ if } 5 < 5 \text{ then } y ::= ++x;\ \text{while } x < 5 \text{ do } y ::= ++x \text{ od } \text{ else } \text{ skip }, m'''')}$$

$$\rightarrow (\text{ if } \text{ false } \text{ then } y ::= ++x;\ \text{while } x < 5 \text{ do } y ::= ++x \text{ od } \text{ else } \text{ skip }, m'''')$$

$$\rightarrow (\text{ skip }, m'''') \rightarrow m''''$$

■