

# Programming Languages (COP 4020/6021) [Fall 2016]

## Assignment V

### Objectives

1. To gain experience writing inference rules in deductive systems.
2. To become familiar with definitions of free variables, alpha-equivalence, and capture-avoiding substitution.

**Due Date:** Monday, February 29, 2016 (at the beginning of class, 5:00pm).

### Assignment Description

Do the following by yourself.

Consider the following language L.

expressions  $e ::= x \mid n \mid e_1 + e_2 \mid \text{let val } x = e_1 \text{ in } e_2 \text{ end} \mid \text{if } e_1 \text{ then } e_2 \text{ else } e_3 \mid T \mid F \mid B$

Language L contains variables ( $x$ ), natural numbers ( $n$ ), addition expressions, SML-style let-expressions, if-then-else expressions, and *ternary*-logic values. Instead of simply having *true* and *false* values, L has  $T$ ,  $F$ , and  $B$  values, all of which have type *tern*. These ternary values can be interpreted to mean “true” ( $T$ ), “false” ( $F$ ), and “both” ( $B$ ). That is,  $B$  refers to a logical value that is *both* true *and* false (perhaps because it refers to an assertion that is sometimes, but not always, true).

If-then-else expressions operate in the same way as if-then-else expressions in SML, except that the test expression (i.e.,  $e_1$  in any expression of the form *if*  $e_1$  *then*  $e_2$  *else*  $e_3$ ) must have type *tern*, rather than *bool*. Dynamically, when we evaluate the test expression to a value, execution proceeds as follows:

- When the test expression is *true*, we execute only the then-branch. The overall if-expression evaluates to whatever the then-branch evaluates to.
- When the test expression is *false*, we execute only the else-branch. The overall if-expression evaluates to whatever the else-branch evaluates to.
- When the test expression is *both*, we execute *both* then- and else-branches to values  $v_{\text{then}}$  and  $v_{\text{else}}$ . If  $v_{\text{then}}$  and  $v_{\text{else}}$  are natural numbers then the overall if-expression evaluates to  $v_{\text{then}} + v_{\text{else}}$ . If  $v_{\text{then}}$  and  $v_{\text{else}}$  are ternary-logic values then the overall if-expression evaluates to  $v_{\text{then}} \wedge v_{\text{else}}$ , as defined in the following truth table:

$\wedge$	<b>T</b>	<b>F</b>	<b>B</b>
<b>T</b>	T	B	B
<b>F</b>	B	F	B
<b>B</b>	B	B	B

Provide definitions for (1) free variables, (2) capture-avoiding substitution, and (3) alpha-equivalence in L.

**Submission Notes**

- Turn in a hardcopy (handwritten or printed) version of your solutions. Please do not email solutions or upload them into Canvas.
- Write the following pledge at the end of your submission: “I pledge my Honor that I have not cheated, and will not cheat, on this assignment.” Sign your name after the pledge. Not including this pledge will lower your grade 50%.
- You may submit solutions up to 2 days (48 hours) late with a 15% penalty.
- If you think there’s a chance you’ll be absent or late for class on the date this assignment is due, you’re welcome to submit solutions early by giving them to me or the TA before or after class, or during any of our office hours.