

**Control Flow Analysis and Logic Programming
Section and Practice Problems**

Monday April 27, 2015

1 Control Flow Analysis

Consider the following lambda calculus program.

$$(\lambda f. (f\ 76) + (f\ 77)) (\lambda a. a)$$

- Add labels to the program. That is, make it an expression in the labeled lambda calculus of Lecture 22, where every label is unique.
- Let e be your labeled lambda calculus program. Write out $\mathcal{C}[[e]]_e$, i.e., the set of constraints for the program e . (Hint, you should have 20 constraints in total. In particular, for each of the 3 applications, you should have 4 constraints, 2 for each of the lambda terms in the program.)
- Find C^* and r^* , the smallest functions that satisfy the constraints you generated in the question above.
- Check that your functions C^* and r^* make sense. That is, if an expression labeled l can evaluate to an expression labeled l' , do you have $l' \in C^*(l)$?
- Consider adding the expression $(\text{let } x = e_1 \text{ in } e_2)^l$ to the language. Define $\mathcal{C}[(\text{let } x = e_1 \text{ in } e_2)^l]_e$. Try rewriting the program above using one or more let expressions, and make sure that the constraints you generate for the modified program produce the same solution C^* and r^* .

2 Logic Programming

- Consider the following Prolog program (where $[]$ is a constant representing the empty list, $[t]$ is shorthand for $\text{cons}(t, [])$ and $[t_1, t_2|t_3]$ is shorthand for $\text{cons}(t_1, \text{cons}(t_2, t_3))$).

```
foo([], []).
foo([X], [X]).
foo([X, Y|S], [Y, X|T]) :- foo(S, T)
```

For each of the following queries, compute the substitutions that Prolog will generate, if any. (Note that there is a difference between an empty substitution, and no substitution.) If the query evaluation will not terminate, explain why.

- $\text{foo}([a, b], X)$.
- $\text{foo}([a, b, c], X)$.
- $\text{foo}([a, b], [a, b])$
- $\text{foo}(X, [a])$
- $\text{foo}(X, Y)$.

(b) Consider the following Datalog program.

```
bar(a, b, c).  
bar(X, Y, Z) :- bar(Y, X, Z).  
bar(X, Y, Z) :- bar(Z, Y, X), quux(X, Z).  
quux(b, c).  
quux(c, d).  
quux(X, Y) :- quux(Y, X).  
quux(X, Z) :- quux(X, Y), quux(Y, Z).
```

Find all solutions to the query `bar(X, Y, Z)`.