

PROBLEM SHEET 2

Alex Kavvos

The following questions are about the language of numbers and strings.

1. Write down the abstract syntax tree for the pre-term `plus(let(len(x); i . plus(i ; n)); num[2])`.
2. Assume $\Sigma \stackrel{\text{def}}{=} \{0, 1\}$. Write a program that
 - has a free variable x of type `Str`,
 - appends the string `0110` to x ,
 - computes the length of the compound string, and
 - adds that number to itself.

Your program should not mention the string literal `str[0110]` more than once.

3. Produce a typing derivation for the following terms, assuming that $\Sigma \stackrel{\text{def}}{=} \{0, 1\}$.
 - (i) $x : \text{Str} \vdash x : \text{Str}$
 - (ii) $\vdash \text{plus}(\text{num}[1]; \text{num}[1]) : \text{Num}$
 - (iii) $x : \text{Str} \vdash \text{cat}(x; \text{str}[01]) : \text{Str}$
 - (iv) $x : \text{Str}, n : \text{Num} \vdash \text{plus}(\text{let}(\text{len}(x); i. \text{plus}(i; n)); \text{num}[2]) : \text{Num}$
4. Perform the following substitutions, step-by-step.
 - (i) $\text{plus}(\text{let}(\text{len}(x); i. \text{plus}(i; n)); \text{num}[2])[i/x]$
 - (ii) $\text{plus}(\text{let}(\text{len}(x); i. \text{plus}(i; n)); \text{num}[2])[\text{num}[0]/n]$
 - (iii) $\text{plus}(\text{let}(\text{len}(x); i. \text{plus}(i; n)); \text{num}[2])[i/n]$
5. State the cases of the inversion lemma for the following constructs:
 - (i) `len(e)`
 - (ii) `let(e_1 ; x . e_2)`
6. Prove the weakening lemma for the programming language of numbers and strings.
7. (*) Complete the proof of substitution from Lecture 4.

[Hint: In the case of variables, consider various cases: is it the variable I'm substituting for, or is it not? Also, you will have to use weakening, so assume that you have proven that already.]
8. Prove that types are unique, i.e. that for every context Γ and pre-term e there exists at most one τ such that $\Gamma \vdash e : \tau$.

[Hint: assume that there exist two, and prove that they must be the same.]