HW 5 – Polymorphic Type Inference

CS 421 – Fall 2012

Revision 1.0

Assigned October 2, 2012
Due October 16, 2012, 11:59 pm
Extension 48 hours (20% penalty)

1 Change Log

1.0 Initial Release.

2 Turn-In Procedure

Answer the problem below, save your work as a PDF (either scanned if handwritten or converted from a program), and hand in the PDF. Your file should be named `hw5.pdf`.

3 Objectives and Background

The purpose of this HW is to test your understanding of how to use typing rules to perform polymorhic type derivations in a funtional programming language (here with OCaml syntax). Another purpose of HWs is to provide you with experience answering non-programming written questions of the kind you may experience on the midterms and final.

4 Problems

1. (25 points) Give a complete type derivation for the following typing judgment's.

$$\left\{\begin{array}{l}\texttt{tl}:\forall\texttt{'a. 'a list -> 'a list}\\ \texttt{(::)}:\forall\texttt{'a. 'a -> 'a list -> 'a list}\\ \texttt{(=)}:\forall\texttt{'a. 'a -> 'a -> bool}\end{array}\right\}\vdash$$

```
(let rec length =
  fun lst -> if lst = [] then 0
             else 1 + length (tl(lst))
in
 length (("a" :: []) :: [])
): int
```

As a suggestion for formatting, you may want to name subtrees of the proof and write them out separately. Note, we are asking for a type judgment not the intermediate state of a type inferencing algorithm.

Solution:

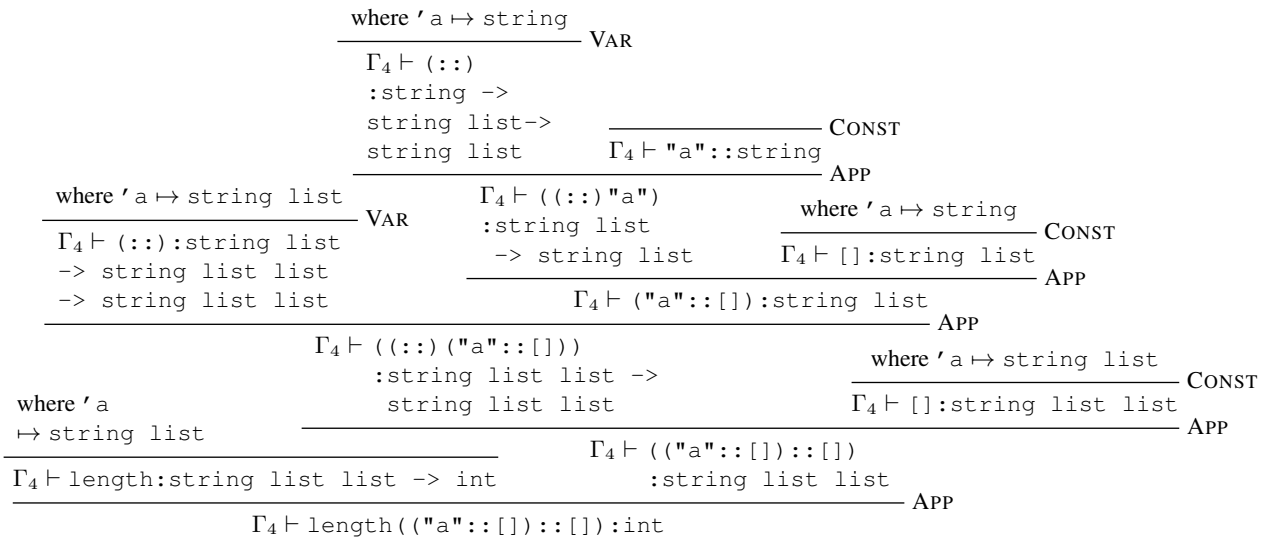
Let Γ_1 = {`tl : ∀'a. 'a list -> 'a list; (::) : ∀'a. 'a -> 'a list -> 'a list; (=) : ∀'a. 'a -> 'a -> bool`}

Γ_2 = {`length : 'a list -> int; tl : ∀'a. 'a list -> 'a list; (::) : ∀'a. 'a -> 'a list -> 'a list; (=) : ∀'a. 'a -> 'a -> bool`}

Γ_3 = {`lst : 'a list; length : 'a list -> int; tl : ∀'a. 'a list -> 'a list; (::) : ∀'a. 'a -> 'a list -> 'a list; (=) : ∀'a. 'a -> 'a -> bool`}

Γ_4 = {`length : ∀'a. 'a list -> int; tl : ∀'a. 'a list -> 'a list; (::) : ∀'a. 'a -> 'a list -> 'a list; (=) : ∀'a. 'a -> 'a -> bool`}

Also let $InTree =$



The proof is then: