
Midterm Exam (30 pts)

1 Type Inference (3 + 4pts)

Derive, wherever possible, the type of the function defined by the following SML-97 definitions.

1. `fun f g (x::xs) = g(f g xs) x;`
2. `fun h g a b = (g 0 (g a b));`

2 Function Definitions (4 + (3+1)*3 pts)

```
fun accumulate f a [] = a
  | accumulate f a (x::xs) = accumulate f (f(x,a)) xs;
```

- What is the signature of `accumulate`?
- Fill in the named-blanks so that the following equivalences hold among list functions.

```
(fn (x: 'a list) => 1)) = accumulate ___f1___ ___a1___
    all p = accumulate ___f2___ ___a2___
    reverse = accumulate ___f3___ ___a3___
```

Recall that `all` takes a unary predicate and a list, and returns true iff the predicate is true of all the elements in the list. `reverse` takes a list and returns the list obtained by reversing it.

3 SML features (3 + 4 pts)

Explain clearly the software engineering benefits that accrue out of using (1) *Polymorphic Type System* and (2) *Higher-Order Functions* in SML from a programmer's perspective. Your description should be clear, concise, and technically sound.