CS 312 Data Structures and Functional Programming
Fall 2005

Prelim II

11/15/2005

NAME:______________________________

CU ID:______________________ Net ID:______________________

Section instructor______________________

You have one and a half hours to do this exam.

All programs in this exam must be written in SML.

Problem	Score	Grader
1 (25 pts)		
2 (12 pts)		
3 (8 pts)		
4 (10 pts)		
5 (20 pts)		
6 (25 pts)		
Total (100 pts)		

1. (Type inference in ML, 25 points)

   (a) (5 points) Consider the ML function shown below.

```
fun foldr f a l =
  case l of
    nil => a
  | (h::t) => f(h,foldr f a t)
```

   i. (3 points) Write down a set of type equations for this function. For each equation, write two or three words to explain where the equation comes from.

```
Type variables:
f:t1
a:t2
l:t3
return type of foldr: t4}


Type equations (one possibility):
t3 = t3' list (because l can be nil)
t1 = t3' x t4 -> t5 (application of f)
t5 = t2 (two clauses of case must have same type)
t2 = t4 (type of case expression is return type)
```

   ii. (2 points) Solve this set of equations to determine the most general type for this function.

```
Solve to get
t1 = t3' x t4 -> t4
t2 = t4
t3 = t3' list
t5 = t4
So type of foldr is ((t3' x t4)-> t4) -> t4 -> (t3' list) -> t4.
```

(b) (10 points) Consider the ML function shown below.

```
fun scanr f z = foldr (fn(e,l) => f(e, hd l)::l) [z]
```

i. (5 points) Using the type signature you computed for `foldr`, write down a set of type equations for this function. For each equation, write two or three words to explain where the equation comes from.

```
Type variables:
f:t1
z:t2
return type of scanr: t3
e: t4
l:t5

foldr has type (axb -> b) -> b ->(a list) -> b

Type equations:
t5 = t5' list (hd l)
t1 = t4xt5' -> t7 (application of f)
t5 = t7 list (::)
t4xt5 -> t7 list = axb -> b (type of foldr)
t2 list = b (type of foldr)
t3 = (a list) -> b (type of foldr)
```

ii. (5 points) Solve this set of equations to determine the most general type for this function.

```
t1 = t4xt2 -> t2
t3 = (t4 list) -> (t2 list)
t5 = (t2 list)
t7 = t2
a = t4
b = (t2 list)
t5' = t2

So type of scanr is

(t4xt2->t2) -> t2 -> (t4 list) -> (t2 list)
```

(c) (5 points) Consider the ML functions shown below. You can assume that the type of `id` is `'a` $\rightarrow$ `'a`.

```
fun id x = x

fun foo f x = (id f) (id x)
```

i. (3 points) Using the type signature of `id`, write down a set of type equations for `foo`. For each equation, write two or three words to explain where the equation comes from.

ii. (2 points) Solve this set of equations to determine the most general type for `foo`.

```
Type variables:
f:a
x:b
return type of foo: r

Let first instance of id by instantiated as
c -> c and second as d -> d. So type equations are

c -> c = a -> e
d -> d = b -> f
e = f -> r

which can be solved to give
a = d -> r
b = d
e = d -> r
f = d
c = d -> r

So type of foo is (d -> r) -> d -> r
which makes sense.
```

(d) (5 points) Write down a one-line function for which the ML type inference system is unable to compute a type. Justify your answer briefly.

```
fun foo f g x = (f g) (f x)
is one possibility. It looks similar to the S combinator
but it cannot be typed because the type equations are circular.
Intuitively, it requires the result of applying f to something
to return a function that can take as an argument
the result of applying f to something, which cannot be typed
in ML.
```

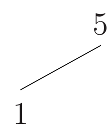
2. (Binary search trees and splay trees, 12 points)

Emil is learning about binary search trees and splay trees.

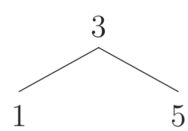
(a) (8 points) Suppose Emil has a splay tree. Draw his splay tree after each step of the following sequence of operations: (insert 1), (insert 5), (insert 3), (insert 2), (insert 4).

After 1 step: 1

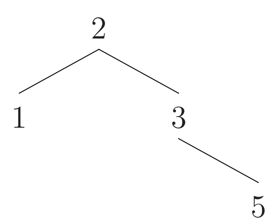
After 2 steps:



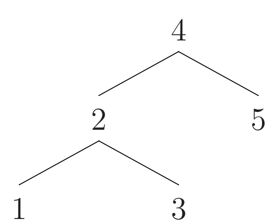
After 3 steps:



After 4 steps:



After all 5 steps:



(b) (4 points) Emil has an empty splay tree while Emily has an empty binary search tree. You will ask them to insert the integers 1 through 5 in some order into their trees. For example, one possible order you may give them is "Start, insert 4, insert 3, insert 1, insert 2, insert 5, stop."

Is there an order of these insertions for which their trees are identical after each insertion in the sequence of five insertions?

If not, is there an order of these insertions for which their trees are identical after all insertions are completed?

Justify your answer.

```
No. In a BST, the root node of tree contains the first
element that is inserted. In a splay tree, the root node always
contains the latest element inserted into the tree. So for any
sequence of insertions greater than 1, the BST and splay tree
must be different.
```

3. (Scope rules, 8 points)

   Consider the following code:

```
val x = ref 0
fun inc(y) = (x := !x + 1; x)
val x = ref 1
fun f(x) = if (!x)*(!x) < 2 then !x else f(x)
```

   (a) In a variant of ML that is eager/static (like SML), what (if anything) is the value of f(inc(x))?

   (b) In a variant of ML that is eager/dynamic, what (if anything) is the value of f(inc(x))?

   (c) In a variant of ML that is lazy/static, what (if anything) is the value of f(inc(x))?

   (d) In a variant of ML that is lazy/dynamic, what (if anything) is the value of f(inc(x))?

   `Answers: 1, infinite loop, infinite loop, infinite loop, infinite loop.`

4. (Asymptotic complexity, 10 points)

(a) Suppose that the function `M` is defined for all powers of 2 and is described by the following recurrence equation and base case.

```
M(n) = n-1+2M(n/2)
M(1) = 0
```

i. Write down the exact solution for `M` when `n` is a power of 2.

```
Use the expansion method. Students get:
M(n)= nlog(n)-n+1
```

ii. What is the asymptotic order of `M(n)`?

```
nlog(n)
```

(b) Find the asymptotic order of the solution to the following recurrence equation.

T(n) = 2T(n/2) + cn^2
T(1) = 1

```
Use the expansion method again. You get:

T(n) = 2cn^2 - (2c-1)n

So asymptotic order of T(n) is O(n^2)
```

5. (Short answers, 20 points)

(a) (5 points) The following outline indicates that `P` is an invariant of the while loop, while `Q` and `R` are the pre- and post-conditions. State in words the four steps involved in proving a loop like the one shown below correct.

```
{Q}
   initialization;
   {invariant P}
   while ( B ) {
       S
   }
{R}
```

```
Answer:
1.  Prove that P is true initially:  {Q} initialization {P}
2.  Prove that P is true upon termination: (P and !B)  => R
3.  Prove that the loop makes progress toward termination (not done forma
4.  Prove that P is indeed a loop invariant: {P and B} S {P}
```

(b) (5 points) Write down the type and value that the following SML fragment evaluates to.

```
let
  datatype 'a t = A of 'a t list
  val a = A [A [A [], A []], A [A [], A []]]
  val m = foldr (fn (x,y) => if x > y then x else y) 0
  val rec d = fn A c => 1 + m (map d c)
in d a
end
```

```
Answer: 3:int
```

(c) (5 points) Explain the difference between polymorphism of Java methods and polymorphism of SML functions.

```
Java: methods are polymorphic in the sense that the type of
the reference is always a supertype of the type of the object.
So the actual parameter can be of many types as long as that type
is a subtype of the declared type of the reference. This is
sometimes inclusion polymorphism.

ML: parameteric polymorphism.
```

(d) (5 points) Using the environment model diagrams shown in class, draw the environment immediately after each invocation of `fib` in the following code. Your diagrams must show the static and dynamic links clearly.

```
let
  fun fib n = if (n < 2) then 1 else fib(n-1) + fib(n-2)
in fib(3) end
```

6. (B-trees, 25 points)

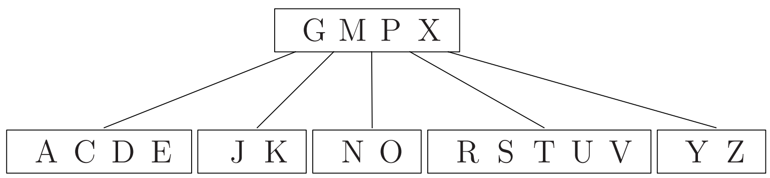
The picture below shows a B-tree B0 whose minimum degree is 3.

(a) Draw the B-tree B1 after the key `B` is inserted into B0.

(b) Draw the B-tree B2 after the key `Q` is inserted into B1.

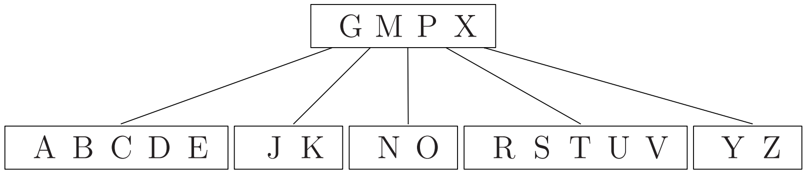
(c) Draw the B-tree B3 after the key `L` is inserted into B2.

(d) Draw the B-tree B4 after the key `F` is inserted into B3.

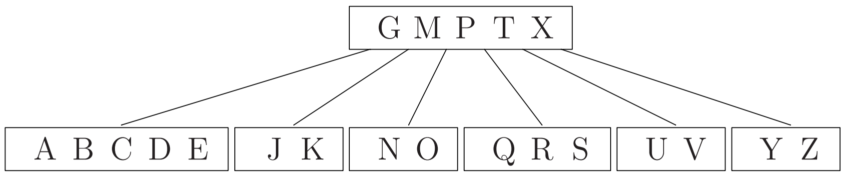


Answer on new page

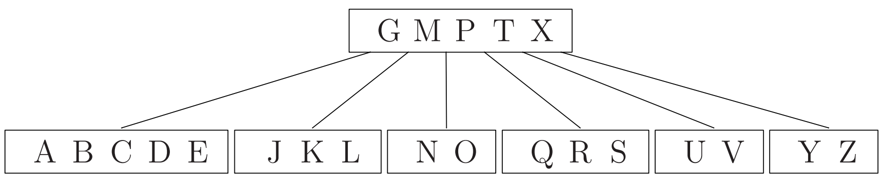
B1:


G M P X
A B C D E
J K
N O
R S T U V
Y Z

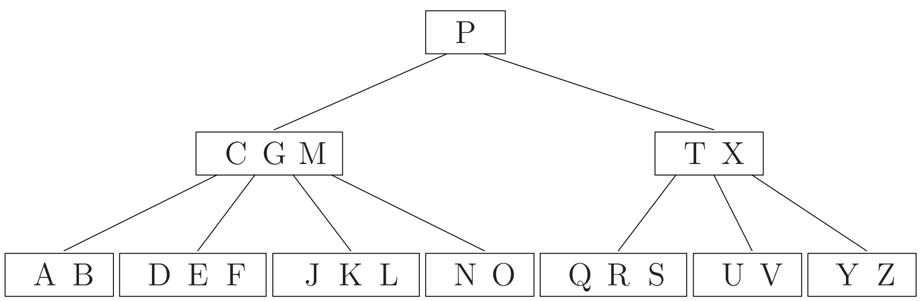
B2:


G M P T X
A B C D E
J K
N O
Q R S
U V
Y Z

B3:


G M P T X
A B C D E
J K L
N O
Q R S
U V
Y Z

B4:


P
C G M
T X
A B
D E F
J K L
N O
Q R S
U V
Y Z