Sample Exam I
Sample Solutions

Problem 1

(a) Not equivalent — non-null strings described by the BNF must end in 'b', so string 'a' is one defined by the syntax diagram, but not the BNF.

(b) Not equivalent — the syntax diagram yields string ababa, but the BNF only allows $(ab)^k a$ where k is odd.

Problem 2

The “delSpaces problem” was recently discussed in several articles[1] about what a horrible job of writing the program (in C++) many students do. Despite this being a small simple program, many individuals “lose sight of the forest because of the trees”. If you doubt the potential for going astray, try writing it yourself in Java or C++. With Haskell the simplicity of the problem carries through to the solution too. The simplest solution is iterative using the pre-defined function 'filter'.

```
delSpaces xs = filter (/=' ') xs          -- select all non-spaces from xs
```

A recursive definition is also quite straightforward.

```
delSpaces [ ] = [ ]
delSpeace (x:xs)
      | x=' '= delSpaces xs       -- drop space and continue
      | otherwise = x: delSpaces xs    -- retain non-space
```

Problem 4

Using recursion is quite natural for this problem, so it’s the preferred approach. The idea is to extract the segment up to the first delimiter, delete these items and the delimiter, and then repeat the process.

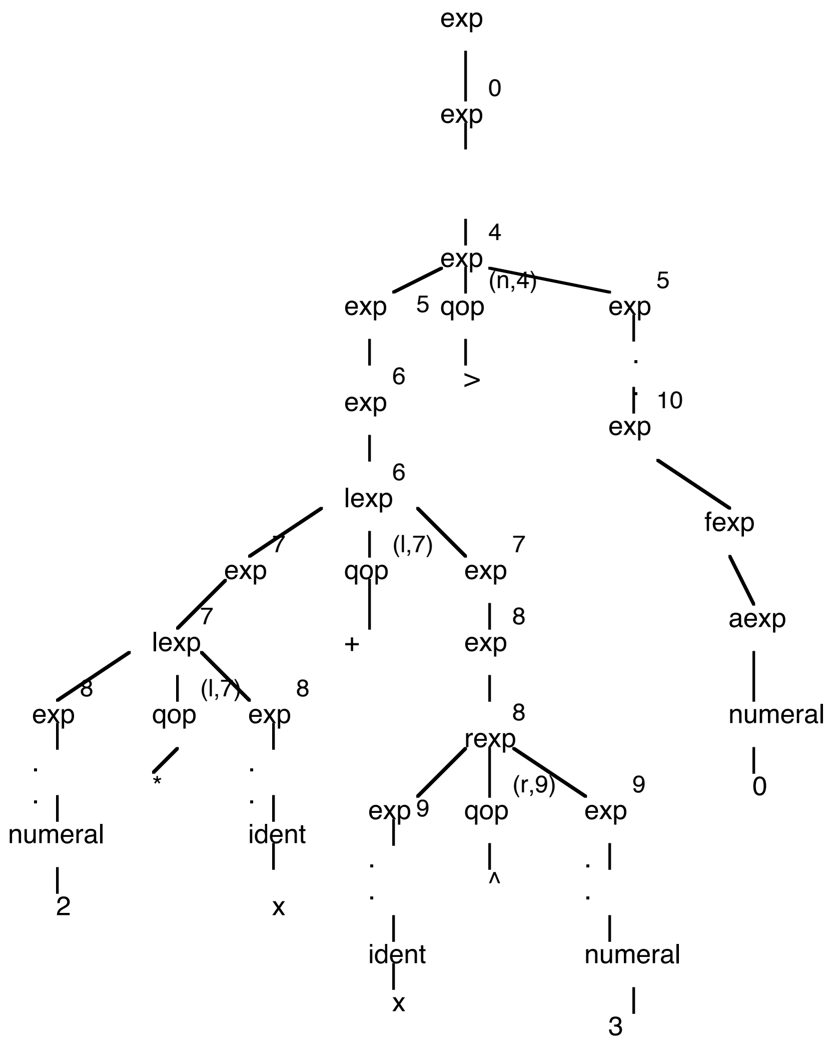
```
replace x y [ ] =[ ]
replace x y (z:zs)
    | x==z          = y:replace x y zs
    | otherwise   = z: replace x y zs
```

An iterative alternative can be written using an auxiliary function and map

```
replace x y zs = map rpl zs
                    where rpl z
                            | z==x        = y
                            | otherwise  = z
```

[1] “A Really Good Idea”, *IEEE Computer* Dec. 1999; “Why Johnny Can’t Program”, *IEEE Computer* Dec. 2000; “Johnny’s Craft Requires Training”, *IEEE Computer* Feb. 2001.

Problem 3


exp
0
exp
4
exp
exp
5
qop
(n,4)
exp
5
>
6
exp
.
.
10
exp
6
lexp
fexp
exp
7
qop
(l,7)
exp
7
aexp
7
lexp
+
8
exp
numeral
exp
8
qop
(l,7)
exp
8
8
rexp
0
*
numeral
ident
exp
9
qop
(r,9)
exp
9
2
x
^
ident
numeral
x
3