CMSC 330 Exam #2 practice questions #1– solutions Fall 2012

1. Here are three slightly different solutions:

```
let create_contact_list names =
  let rec search list name =
    match list with first::phone::rest ->
      if name = first
        then phone
        else search rest name
  in search names

let rec create_contact_list names =
  let rec lookup name =
    match names with first::phone::rest ->
      if first = name
        then phone
        else ((create_contact_list rest) name)
  in lookup

let rec create_contact_list names =
  fun name -> match names with
    first::phone::rest ->
      if first = name
        then phone
        else ((create_contact_list rest) name)
```

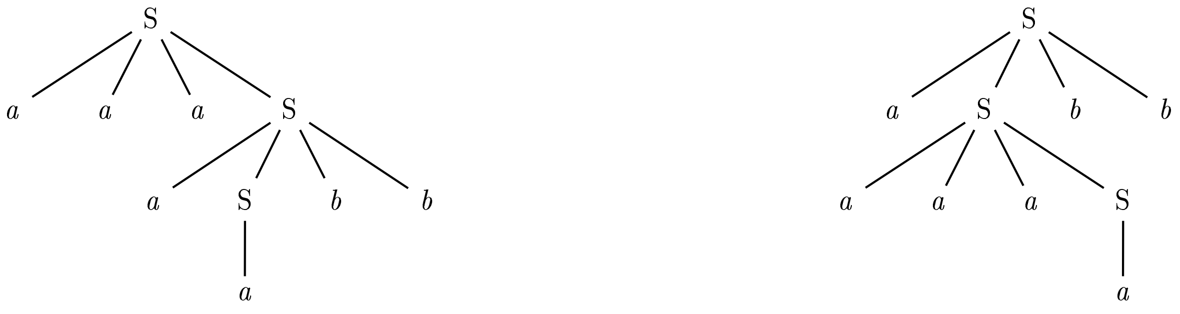
2. One note before giving the solution: recall from CMSC 250 that zero is a multiple of 3 (the multiples of 3 are $\{\ldots, -9, -6, -3, 0, 3, 6, 9, \ldots\}$). Therefore ϵ is not a valid string in this language.

Your first thought might be to write a grammar that looked something like the following, which would generate strings where m + n **is** a multiple of 3 and then generate either one or two extra a's or b's, thereby forcing m + n to **not** be a multiple of 3:

$$\mathrm{S} \rightarrow aaa\mathrm{S} \mid aa\mathrm{S}b \mid a\mathrm{S}bb \mid \mathrm{S}bbb \mid \mathrm{T} \qquad\qquad \mathrm{S} \rightarrow a\mathrm{T} \mid aa\mathrm{T} \mid \mathrm{T}b \mid \mathrm{T}bb \mid a\mathrm{T}b$$

$$\mathrm{T} \rightarrow a \mid aa \mid b \mid bb \mid ab \qquad\qquad \mathrm{T} \rightarrow aaa\mathrm{T} \mid aa\mathrm{T}b \mid a\mathrm{T}bb \mid \mathrm{T}bbb \mid \epsilon$$

The problem with these grammars is that they're ambiguous. For example, in the grammar on the left:



Here are several unambiguous versions:

$$\begin{array}{lcl}
\mathrm{S} & \rightarrow & aaa\mathrm{S} \mid \mathrm{T} \\
\mathrm{T} & \rightarrow & aa\mathrm{T}b \mid \mathrm{U} \\
\mathrm{U} & \rightarrow & a\mathrm{U}bb \mid \mathrm{V} \\
\mathrm{V} & \rightarrow & \mathrm{V}bbb \mid \mathrm{W} \\
\mathrm{W} & \rightarrow & a \mid aa \mid b \mid bb \mid ab
\end{array}$$

$$\begin{array}{lcl}
\mathrm{S} & \rightarrow & \mathrm{S}bbb \mid \mathrm{T} \\
\mathrm{T} & \rightarrow & a\mathrm{T}bb \mid \mathrm{U} \\
\mathrm{U} & \rightarrow & aa\mathrm{U}b \mid \mathrm{V} \\
\mathrm{V} & \rightarrow & aaa\mathrm{V} \mid \mathrm{W} \\
\mathrm{W} & \rightarrow & a \mid aa \mid b \mid bb \mid ab
\end{array}$$

$$\begin{array}{lcl}
\mathrm{S} & \rightarrow & a\mathrm{T} \mid aa\mathrm{T} \mid \mathrm{T}b \mid \mathrm{T}bb \mid a\mathrm{T}b \\
\mathrm{T} & \rightarrow & aaa\mathrm{T} \mid \mathrm{U} \\
\mathrm{U} & \rightarrow & aa\mathrm{U}b \mid \mathrm{V} \\
\mathrm{V} & \rightarrow & a\mathrm{V}bb \mid \mathrm{W} \\
\mathrm{W} & \rightarrow & \mathrm{W}bbb \mid \epsilon
\end{array}$$

Any completely–correct answer should have the following properties:

p1: (completeness) It generates every valid string.

p2: (correctness) It generates only valid strings (it does not generate any invalid strings).

p3: (ambiguity) It's unambiguous.

3. `13 17`

4. a.

```
f1
f1
f2
g1
g2
g2
```

b. Only `z`.

c. Parametric polymorphism.