TA: Jade Cheng

ICS 312

Homework Solution #19

Due Date: November 05, 2009

Grammar Homework

Question: Devise a grammar for the set of all strings $\{a^{2n}b^{3n} \mid n > 0\}$.

Answer:

```
S --> aa S bbb | epsilon
```

Question: Devise a grammar for the set of all strings on the alphabet {a, b, c} which contain either 2 or 4 b's.

Answer: Convert the CFG from RE following the procedures below.

Regular Expression:

```
RE: <ac>*b<ac>*b<ac>*(b<ac>*b)?<ac>*
ac: a | c
```

Operands:

```
<ac>     --> a | c
```

Operator *:

```
<star>   --> <ac> <star> | epsilon
```

Operator ?:

```
<ques>   --> b <star> b | epsilon
```

Final context-free grammer:

```
<string> --> <star> b <star> b <star> <ques> <star>
<ques>   --> b <star> b | epsilon
<star>   --> <ac> <star> | epsilon
<ac>     --> a | c
```

Question: Devise a grammar for the set of all even integers without leading zeros.

Answer: Convert the CFG from RE following the procedures below.

Regular Expression:

```
RE:      -? (evenD | allD (allDz)* evenDz)
evenD:   2 | 4 | 6 | 8
allD:    1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
evenDz:  0 | 2 | 4 | 6 | 8
allDz:   0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Operands:

```
<evenD>  --> 2 | 4 | 6 | 8
<allD>   --> 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<evenDz> --> 0 | 2 | 4 | 6 | 8
<allDz>  --> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Operator *:

```
<star>   --> <allDz> <star> | epsilon
```

Operator |:

```
<pNum>   --> <evenD> | <allD> <star> <evenDz>
```

Final context-free grammar:

```
<string> --> - <pNum> | <pNum>
<pNum>   --> <evenD> | <allD> <star> <evenDz>
<star>   --> <allDz> <star> | epsilon
<evenD>  --> 2 | 4 | 6 | 8
<allD>   --> 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<evenDz> --> 0 | 2 | 4 | 6 | 8
<allDz>  --> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Question: Supply derivations of `a, a * a, a * a + a, a * ( a + a )` for the following grammar:

```
E --> E + T | E - T | T
T --> T * P | P
P --> ( E ) | a
```

Answer: `a:`

```
E --> T
  --> P
  --> a
```

`a * a:`

```
E --> T
```

```
                --> T * P
                --> P * P
                --> a * P
                --> a * a

        a * a + a:
                E --> E + T
                --> T + T
                --> T * P + T
                --> P * P + T
                --> a * P + T
                --> a * a + T
                --> a * a + P
                --> a * a + a

        a * ( a + a ):
                E --> T
                --> T * P
                --> P * P
                --> a * P
                --> a * ( E )
                --> a * ( E + T )
                --> a * ( T + T )
                --> a * ( P + T )
                --> a * ( a + T )
                --> a * ( a + P )
                --> a * ( a + a )
```

Question: The following grammar provides precedence for multiplication over addition.

```
E --> E + T | T

T --> T * a | a
```

Write grammars build up of `+` `*` and `a`, such that:

a. neither addition nor multiplication have precedence, but operations are preformed from left to right in the same order they occur.

Answer:

```
E --> E + a | E * a | a
```

b. same as a. but from right to left.

Answer:

```
E --> a + E | a * E | a
```

Question: Devise a grammar for the set of all strings $\{a^i b^j c^k \mid i = 2*(j+k)\}$.

Answer:

```
U --> aaUc | V | e
V --> aaVb | e
```