angel brown
CSIS 618
Homework 2

2.1

Integer → *Integer* *Digit*
→ *Integer* *Digit* *Digit*
→ *Integer* *Digit* *Digit* *Digit*
→ *Digit* *Digit* *Digit* *Digit*
→ 4 *Digit* *Digit* *Digit*
→ 4 5 *Digit* *Digit*
→ 4 5 2 *Digit*
→ 4 5 2 0

If we have just one digit *x*, we can derive it as follows:

Integer → *Digit*
→ *x*

That is, we can derive *x* in just two steps. Since there is no production rule which takes the start symbol *Integer* to a terminal symbol, there is no valid derivation of length 1.

Now suppose that any string of digits of length *d-1* requires at least *2(d-1)* steps. Then a string of digits of length *d* may be written *yx* where *x* is one digit and *y* is a string of digits of length *d-1*. Since *y* is length *d-1*, a derivation of *y* is at least *2(d-1)* steps. Then we know from above that a derivation of one digit requires at least *2* steps. So a derivation of *yx* (of length d) must be at least *2 + 2(d-1) = 2d* steps.

2.4

Identifier → *Letter Digit Letter*
→ *Letter Digit* *i*
→ *Letter* *2* *i*
→ *a* *2* *i*

2.5

(a)

```
                    Assignment
                 /      |      \
          Identifier    =    Expression
              |                  |
              x              Conjunction
                                 |
                              Equality
                                 |
                              Relation
                                 |
                              Addition
                                 |
            Term   AddOp   Term   AddOp   Term
              |      |       |      |       |
              x      +       a      -       1
```

(b)

```
                    Assignment
                 /      |       \
          Identifier    =     Expression
              |                   |
              a               Conjunction
                                  |
                               Equality
                                  |
                               Relation
                                  |
                               Addition
                                  |
                                 Term
                                  |
      Factor    MulOp    Factor    MulOp    Factor
        |         |        |         |        |
     Primary      *     Primary      /     Primary
        |                  |                  |
    Identifier         Identifier         Identifier
        |                  |                  |
      Letter             Letter             Letter
        |                  |                  |
        b                  c                  d
```

(c)

```
                 Assignment
              /       |       \
       Identifier     =     Expression
           |                    |
           i                Conjunction
                                |
                             Equality
                                |
                             Relation
                                |
                             Addition
                                |
        Term                  AddOp            Term
          |                     |                |
        Factor                  +       Factor  MulOp    Factor
          |                               |       |        |
        Primary                        Primary    *      Primary
          |                               |                |
        Identifier                     Identifier        Expression
          |                               |                |
        Letter                          Letter           Conjunction
          |                               |                |
          i                               j              Equality
                                                           |
                                                         Relation
                                                           |
                                                         Addition
                                                           |
                                                 Term     AddOp     Term
                                                   |        |         |
                                                 Factor     -       Factor
                                                   |                  |
                                                   k                  3
```

2.20

(a)

```
        =
      /   \
     x     +
          / \
         x   -
            / \
           a   1
```

(b)

```
        =
      /   \
     a     *
          / \
         b   /
            / \
           c   d
```

(c)

```
        =
       / \
      i   +
         / \
        i   -
           / \
          *   3
         / \
        j   k
```