Sample Solutions for Chapter 4

Exercise 1: Understanding Attributed Grammars

Given the following attributed grammar:

```
N <↑n>          (. int n; .)
= D <↑n>
  {  D <↑x>     (. n = 2 * n + x; .)
  } .

D <↑x>
=    "0"        (. x = 0; .)
  |  "1"        (. x = 1; .) .
```

a) What does this grammar do, i.e., what does N return as an output attribute?

b) Translate the two productions into parser methods that also contain the attributes and semantic actions. The terminal symbols "0" and "1" can be denoted by the token names zero and one in the parser.

Solution

a) The grammar calculates the value of a binary number that is given as a sequence of binary digits. The nonterminal symbol D returns the value of a single binary digit and the nonterminal symbol N the value of the entire binary number.

b) Translation into parser methods (attributes and semantic actions are printed in gray):

```
static int N() {
   int n = D();
   while (sym == zero || sym == one) {
      int x = D();
      n = 2 * n + x;
   }
   return n;
}

static int D() {
   int x;
   if (sym == zero) {
      scan(); x = 0;
   } else if (sym == one) {
      scan(); x = 1;
   } else {
      error("0 or 1 expected"); x = 0;
   }
   return x;
}
```