

## Aufgabe 1: Verstehen attributierter Grammatiken

Gegeben sei folgende attributierte Grammatik:

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

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

- Was macht diese Grammatik, d.h., was liefert N als Ausgangsattribut?
- Übersetzen Sie die beiden Produktionen in Parsermethoden, die auch die Attribute und semantischen Aktionen enthalten. Die Terminalsymbole "0" und "1" können als zero und one angesprochen werden.

### Lösung

- Die Grammatik berechnet den Wert einer Binärzahl, die als Folge von Binärziffern gegeben ist. Das Nonterminalsymbol D liefert den Wert einer einzelnen Binärziffer und das Nonterminalsymbol N den Wert der gesamten Binärzahl als Ausgangsattribut.
- Übersetzung in Parsermethoden (Attribute und semantische Aktionen sind grau dargestellt):

```
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;
}
```