

Aufgabe 10: Syntaxfehlerbehandlung mit allgemeinen Fangsymbolen

Schreiben Sie für folgende Grammatik Parsermethoden und implementieren Sie dabei eine Fehlerbehandlung mit allgemeinen Fangsymbolen (Abschnitt 3.4.2). Sie können die Mengenoperationen im Pseudocode formulieren.

$A = \{ b C \mid d \}$.
 $C = [c] e$.

A ist das Startsymbol, dessen Nachfolger eof ist.

Lösung

Der Nachfolger von A ist eof, die möglichen Nachfolger von C sind b und d (da eine weitere Iteration folgen kann) sowie eof (falls die Iteration zu Ende ist). Die Parsermethode von A wird mit $A(\{eof\})$ aufgerufen.

```
static void A (BitSet sux) {
    for (;;) {
        if (sym == b) {                // first alternative => stay in loop
            scan();
            C({b, d} ∪ sux);
        } else if (sym == d) {        // second alternative => stay in loop
            scan();
        } else if (sym ∈ sux) {       // leave loop
            break;
        } else {                      // error => stay in loop
            error("b or d expected", {b, d} ∪ sux);
            // recover with b or d or sux
        }
    }
}

static void C (BitSet sux) {
    if (sym != c && sym != e) {
        error("c or e expected", {c, e} ∪ sux);
        // recover with c or e or sux
    }
    if (sym == c) scan();
    check(e, sux);
}
```

Die Methode error gibt eine Fehlermeldung aus und überliest dann Symbole, bis eines aus der mitgegebenen Nachfolgermenge sux auftritt (siehe Abschnitt 3.4.2).

```
static void error (String msg, BitSet sux) {
    if (errDist >= 3) {
        System.out.println("line " + la.line + ", col " + la.col + ": " + msg);
        errors++;
    }
    while (!sux.get(sym)) scan();
    errDist = 0; // restart counting
}
```