

Äquivalente Formeln der Aussagenlogik in einem Programmierbeispiel

Formeln der Aussagenlogik gibt es zum Beispiel beim Programmieren, etwa in den Bedingungen von if-Anweisungen. Abhängig davon, ob die Ausdrücke wahr oder falsch sind, werden die zugehörigen Programmteile ausgeführt. Wir betrachten dazu ein kleines C-Programm.

Bei dem Beispiel – eine Übungsaufgabe aus dem Buch „Programmieren in C“ von Kernighan und Ritchie – hat man sich zunächst eine Lösung mit zwei if-Anweisungen ausgedacht, sucht aber nach besseren Varianten. Mit Hilfe der Aussagenlogik formt man um und bekommt ein Ergebnis, das nicht nur kurz sondern auch gut lesbar und gut zu verstehen ist.

Man kann natürlich sagen „Auf das Endergebnis komme ich auch direkt.“, aber es ist doch die Frage, ob man so schnell mehrere Möglichkeiten finden würde und auch noch sicher wäre, daß diese korrekt sind. Die Aussagenlogik mit ihren Umformungsregeln bietet uns hier ein praktisches und mächtiges Werkzeug.

```
#include <stdio.h>

/* Folge von Leerzeichen in der Eingabe durch ein
   einzelnes Leerzeichen in der Ausgabe ersetzen */
main()
{
    int c, prev_c;

    prev_c = 'a'; /* previous character != ' ' */
    while ((c = getchar()) != EOF) {
        if (c != ' ')
            printf("%c", c);
        if (c == ' ' && prev_c != ' ')
            printf("%c", c);
        prev_c = c;
    }
}
```

Zunächst fassen wir die beiden if-Anweisungen zusammen. Man beachte, daß in dem obigen Programm bei jedem Durchlauf der while-Schleife *höchstens eine* der beiden if-Anweisungen ausgeführt wird. Deshalb funktioniert die folgende Zusammenfassung der Anweisungen. (Wir geben nur die while-Schleife an, der Rest bleibt ja unverändert.)

```

while ((c = getchar()) != EOF) {
    if (c != ' ' || (c == ' ' && prev_c != ' '))
        printf("%c", c);
    prev_c = c;
}

```

Mit Umformungsregeln der Aussagenlogik kommen wir zu der folgenden kürzeren Version. Allerdings erscheint uns die Verständlichkeit der Formel nicht gut zu sein; das ist aber natürlich auch Geschmackssache.

```

while ((c = getchar()) != EOF) {
    if (c != ' ' || prev_c != ' ')
        printf("%c", c);
    prev_c = c;
}

```

Eine weitere Umformung mit einer Regel der Aussagenlogik, und zwar mit der Regel von de Morgan, liefert schließlich die folgende Version der if-Anweisung. Sie ist kompakter als die Ausgangsversion, und wir halten sie auch für gut lesbar und verständlich.

```

while ((c = getchar()) != EOF) {
    if ( !(c == ' ' && prev_c == ' ') )
        printf("%c", c);
    prev_c = c;
}

```

Insgesamt haben wir damit als Endergebnis unserer Überlegungen die folgende Formulierung des Programms bekommen.

```

#include <stdio.h>

/* Folge von Leerzeichen in der Eingabe durch ein
   einzelnes Leerzeichen in der Ausgabe ersetzen */
main()
{
    int c, prev_c;

    prev_c = 'a'; /* previous character != ' ' */
    while ((c = getchar()) != EOF) {
        if ( !(c == ' ' && prev_c == ' ') )
            printf("%c", c);
        prev_c = c;
    }
}

```