

6. Übungsblatt

Ausgabe: 26.05.16

Abgabe: 06.06.16

6.1 Zustände

6 Punkte

Wir betrachten das Programm aus Abbildung 1. Geben Sie jeweils in der Notation aus der Vorlesung auf Folie 11 die Zustände an den Positionen 1, 2, 3 und 5 an.

6.2 Operationale Semantik

7 Punkte

Geben Sie die operationale des Teilprogramms zwischen Position 3 und 4 unter Verwendung des folgenden Zustands

buch.chapters[0].startpage → 1	buch.chapters[2].startpage → 5
buch.chapters[0].endpage → 2	buch.chapters[2].endpage → 6
buch.chapters[1].startpage → 3	buch.chapters[3].startpage → 7
buch.chapters[1].endpage → 4	buch.chapters[3].endpage → 8

6.3 Hoare-Kalkül Beweise

7 Punkte

1. Beweisen Sie die partielle Korrektheit im Hoare-Kalkül von

```
/** { buch.pages = 0 and
    buch.chapters[0].startpage = 1 and buch.chapters[0].endpage = 5 and
    buch.chapters[1].startpage = 6 and buch.chapters[1].endpage = 10 } */
int length = 0;
int i = 1;
length = length + (buch.chapters[i].endpage - buch.chapters[i].startpage) + 1;
i = 0;
length = length + (buch.chapters[i].endpage - buch.chapters[i].startpage) + 1;
buch.pages = length;
/** { buch.chapters[0].startpage = 1 and buch.chapters[0].endpage = 5 and
    buch.chapters[1].startpage = 6 and buch.chapters[1].endpage = 10 and
    buch.pages = 10 } */
```

Verwenden Sie dabei die in der zweiten Vorlesung eingeführt Notation vollständig annotierter Programme.

2. Wie sähe der Beweis aus für

```
/** { buch.pages = 0 and
    buch.chapters[0].startpage = n1 and buch.chapters[0].endpage = n2 and
    buch.chapters[1].startpage = n3 and buch.chapters[1].endpage = n4 } */
int length = 0;
int i = 1;
length = length + (buch.chapters[i].endpage - buch.chapters[i].startpage) + 1;
i = 0;
length = length + (buch.chapters[i].endpage - buch.chapters[i].startpage) + 1;
buch.pages = length;
/** { buch.chapters[0].startpage = n1 and buch.chapters[0].endpage = n2 and
    buch.chapters[1].startpage = n3 and buch.chapters[1].endpage = n4 and
    buch.pages = (n2-n1+1)+(n4-n3+1) } */
```

```

struct chapter {
    char title[20];
    int startpage;
    int endpage;
};

// Position 1

struct chapter chapter1 = { {'A', 'n', 'f', 'a', 'n', 'g', '\0'}, 1,9};
struct chapter chapter2 = { {'T', 'e', 'i', 'l', '\0', '\0', '\0'}, 10,19};
struct chapter chapter3 = { {'T', 'e', 'i', 'l', '\0', '\0', '\0'}, 20,29};
struct chapter chapter4 = { {'E', 'n', 'd', 'e', '\0'}, 30,35};

// Position 2

struct book {
    char title[20];
    int pages;
    struct chapter chapters[4];
} buch = { {'D', 'a', 's', '\0', '\0', '\0', '\0'},
          0,
          { chapter1, chapter2, chapter3, chapter4 } };

// Position 3

int length = 0;
int i = 4;
while (i) {
    i = i - 1;
    length = length + (buch.chapters[i].endpage - buch.chapters[i].startpage) +1;
};

// Position 4

buch.pages = length;

// Position 5

```

Abbildung 1: Beispielprogramm

Änderungen:

- 26.05.16 *Initiale Version.*
- 26.05.16 *Doppelte Deklaration von i in Aufgabe 6.3 (2.) entfernt.*
- 30.05.16 *Anpassung der Bedingungen an geänderte Repräsentation ohne expliziten Zustand.*