

Übungsblatt 8: Übersetzer und sprachverarbeitende Werkzeuge (SS 2011)

Hand Out: 8. Juni 2011
Hand In: 22. Juni 2011

Aufgabe 1 Prüfungsfragen zu den Vorlesungen

Denken Sie bitte daran sich Prüfungsfragen zu überlegen. (<http://bit.ly/kJORSi>)

Aufgabe 2 Attributgrammatiken

Betrachten Sie folgende Sprache zur Beschreibung von Texten, gegeben in Katja-Notation.

```
Document ( Comp comp )  
Comp = Composite  
      | Textbox  
Composite * Comp  
Textbox (Integer nlines)
```

```
Layout * Page  
Page * PagePart ( Textbox box, Integer position )
```

Ein Text ist repräsentiert durch ein Dokument, das aus einer Komponente (Comp) besteht. Komponenten können entweder Textboxen (Textbox) oder zusammengesetzte Komponenten (Composite) sein. Textboxen bestehen aus einer gegebenen Anzahl an Zeilen (nlines). Ihre Aufgabe besteht nun darin einen Algorithmus zu entwerfen, der für ein Dokument ein *Layout* berechnet. Ein *Layout* (Layout) ist eine Positionierung von Textboxen auf Seiten (Page). Positionen sind die jeweilige Zeile auf der Seite bei der die Textbox beginnt. Ihr Algorithmus soll für eine gegebene Seitengröße (angegeben in Anzahl von Zeilen) ein Layout mit einer minimalen Anzahl von Seiten finden. Die Reihenfolge der Textboxen soll dabei erhalten bleiben. Textboxen müssen als ganzes auf einer Seite erscheinen (sie dürfen nicht umgebrochen werden). Zwischen zwei Textboxen auf einer Seite muss genau eine Zeile frei bleiben. Textboxen von zusammengesetzten Komponenten können auf verschiedenen Seiten erscheinen und müssen nicht zusammen auf einer Seite sein. Sie können davon ausgehen, dass Textboxen nie größer als Seiten sind.

Realisieren Sie den Algorithmus indem Sie entsprechende Attribute auf dem AST des Dokuments definieren. Die Eingabe des Algorithmus ist die Seitengröße, die Ausgabe ein Layout. Schreiben Sie ihren Algorithmus in Pseudocode auf oder verwenden Sie Attributierungsdiagramme.

Optionale Zusatzaufgabe: Verändern Sie Ihren Algorithmus, sodass immer die volle Seitengröße ausgenutzt wird. Fügen Sie dazu zusätzliche Leerzeilen zwischen Textboxen der selben Seite ein. Bei Seiten mit nur einer Textbox soll diese so gut wie möglich zentriert sein.

Aufgabe 3 Manuelle Übersetzung zu MIPS-Assembler

- a) Übersetzen Sie folgendes C-ähnliches Pseudocodeprogramm von Hand in MIPS-Assembler. Optimieren Sie das Programm dabei *nicht*. Dokumentieren Sie wo die auftretenden Variablen des Programms in der MIPS-Maschine gespeichert werden.

```
int fibonacci[10];  
int i;  
int result;
```

```
void main() {
    i = 2;
    result = 1;

    fibonacci[0] = 1;
    fibonacci[1] = 1;

    while (i < 10) {
        fibonacci[i] = fibonacci[i-1] + fibonacci[i-2];
        i = i+1;
    }
    result = fibonacci[9];
}
```

Testen Sie Ihr Programm mit Mars¹ und geben Sie es in digitaler Form ab.

b) Betrachten Sie folgende Deklaration eines dreidimensionalen Feldes in Pascal:

```
var a:array[-3..3][-3..3][-3..3] of integer;
```

Erstellen Sie von Hand dokumentierten MIPS-Assembler Code für den Feldzugriff $a[-1, 0, 1]$. Vergessen Sie nicht Ihren Lösungsweg anzugeben.

¹<http://courses.missouristate.edu/KenVollmar/MARS/>