Seminar “Implementierung von Stringalgorithmen”

Leiß/Goller, CIS

7.5.2004

3 Graphische Darstellung von Suffixtries

Leider hatte ich beim letzten Mal nicht mitbekommen, daß ich Installationsanweisungen für ein Programm zur Darstellung von Graphen auf die Kursseite legen sollte. Deswegen wird das hier nachgetragen und die Aufgabe nochmal gestellt, da sie von den meisten meines Wissens nicht bearbeitet wurde..

Wir wollten das Graphenvisualisierungsprogramm `graphviz` benutzen, das man unter

`http://www.research.att.com/sw/tools/graphviz/download.html`

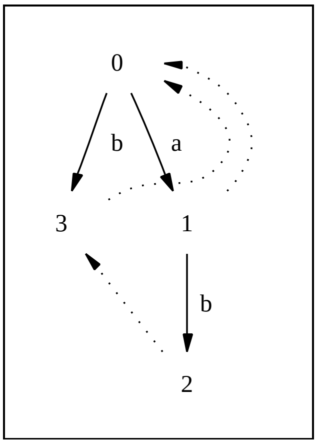
findet. In diesem Paket gibt es neben einem Editor für Graphen u.a. ein Programm `dot`, das aus Beschreibungen von Graphen in einem einfachen Format eine graphische Darstellung berechnet und in z.B. Postscript format ausgibt. Wir brauchen nur dieses Programm `dot`. (Es sollte auf den Rechnern der Informatik installiert sein.) Die Aufgabe war:

Aufgabe 2.3 (= 3.1) Eine Funktion soll implementiert werden, die den Trie traversiert und einen Ausdruck auf STDOUT erzeugt, der als Input für das Programm Dot geeignet ist. Auf diese Weise können wir sehr einfach zu Visualisierungen unserer Baumstruktur kommen. Installationsanweisungen und Angaben zum Inputformat finden sich in Kürze auf der Kursseite.

Für die genauen Optionen von `dot`, die die Darstellung von Ecken, Kanten und Marken im Graphen steuern, beachte die Handbuchseiten `man dot`.

Beispiel: die Eingabe von `dot` ist eine Datei `<Graphbeschreibung>.dot`, in der ein oder mehrere Graphen beschrieben sind. Der Suffixtrie von `aba` hat folgende `dot`-Beschreibung und folgendes `.ps`-Aussehen:

```
digraph suffixTrie_aba {center=true;
size="7.5,14";
ratio=compress;
node [shape=plaintext];
0 -> 3 [label = b] ;
3 -> 0 [style = dotted, constraint = false] ;
0 -> 1 [label = a] ;
1 -> 0 [style = dotted, constraint = false] ;
1 -> 2 [label = b] ;
2 -> 3 [style = dotted, constraint = false] ;
}
```



Aus der Beschreibungsdatei wurde eine Postscriptausgabe durch

```
dot -Tps <Graphbeschreibung>.dot -o <Graphausgabe>.ps
```

erzeugt; andere Graphikformate können auch ausgegeben werden. Die Aufgabe ist also, für einen beliebigen Trie eine `dot`-Beschreibung zu erzeugen. (Damit man den Trie-Datentyp der früheren Aufgabe benutzen kann, kann man statt der Darstellung mit Suffixzeigern auch die Darstellung, bei der die wesentlichen Knoten durch Kreise markiert sind, erzeugen.)

Aufgabe 3.2 Um die Konstruktion des Tries zu veranschaulichen, verwende man die Darstellung mit Suffixzeigern. Man schreibe ein Programm, das zu einem Eingabestring $w = a_1 \cdots a_n$ die bei der Konstruktion nach Ukkonen gebildeten Suffixtries der Präfixe $\epsilon,\ a_1,\ a_1a_2,\ \ldots,\ a_1 \cdots a_n$ erzeugt und ihre `dot`-Beschreibung in eine Datei schreibt.

Die Konstruktion des Tries $T(w)$ kann man dann als „Film“ der aufeinanderfolgenden $T(a_1 \cdots a_i)$ verfolgen, indem man die entsprechende `.ps`-Datei mit `ghostview` oder `gv` anzeigt.

Hinweis: Da der Trie sich während der Konstruktion ändert, muß man zwischendurch die einzelnen Stadien $T(a_1 \cdots a_i)$ in die `.dot`-Datei schreiben. Man muß (glaube ich) die Graphen verschieden benennen, da sonst nur der letzte Graph gezeigt wird.