Computerlinguistische Algorithmen SS 00   3. Übung (4. Mai 2000)

1 Automatenoperationen (8 Punkte)

In dieser Aufgabe geht es darum, die drei Automatenoperationen

- `(concat-automata r-tabelle s-tabelle)`
- `(join-automata r-tabelle s-tabelle)`
- `(kleene-star r-tabelle)`

zu implementieren, die jeweils eine neue Tabelle zurückliefern. Gehe dazu folgendermassen vor:

a) Implementiere eine Funktion `verschiebe-automat`, die zwei Übergangstabellen (für nicht-deterministische Automaten mit Epsilon-Übergängen) und einen (positiven, ganzzahligen) Offset als Argumente erhält. Diese Funktion soll die erste Übergangstabelle so in die zweite übertragen, dass der ehemalige Startzustand an der Position Offset im neuen Feld liegt. Die Übergänge müssen natürlich entsprechend angepaßt werden.

b) Mit Hilfe der Funktion aus Teil a) können die drei Operationen realisiert werden. Erzeuge eine neue, hinreichend große Übergangstabelle, bewege den oder die Originalautomaten in die neue Tabelle und erzeuge die neuen Übergänge. Achte dabei auf eventuell neu zu erzeugende Zustände.

2 Wortautomaten (6 Punkte)

Vorsicht! Aufgabe vielleicht zuerst ganz lesen.

a) Schreibe eine einstellige Funktion `stringlist2table`, die als Eingabeparameter eine Liste von Strings erhält und die Tabelle für einen Automaten liefert, der genau diese Strings erkennt.

b) Erweitere `stringlist2table()` so, daß es auch eine Liste von `cons`-Zellen nimmt, deren `first` ein String ist und der `rest` ein beliebiges Lisp-Objekt. Das Lisp-Objekt soll so in die Tabelle eingetragen werden, daß es als Wert von `traverse-string` zurückgegeben wird, wenn der String erkannt wurde.

Teste beide Funktionen entsprechend mit den Listen aus

`http://www.dfki.de/~becker/lispkurs/vorlesungen.ss00/wortlisten.lsp`

3 Erkennen von Patterns (6 Punkte)

a) Schreibe analog zur vorherigen Aufgabe eine einstellige Funktion `patterns2table`, die als Eingabeparameter eine Liste von Patterns erhält und die Tabelle für einen Automaten liefert, der diese Patterns erkennt.

b) Gleichzeitig muss auch eine zweite Token $\longrightarrow$ Code Tabelle erzeugt werden, die die Lisp-Symbole auf Zahlen abbildet. Schreibe eine Funktion `category2code`, die eine `eq` - Hashtabelle zurückgibt, die das leistet (siehe `http://www.dfki.de/~becker/lispkurs/clm/node154.html`)

Teste die Funktionen mit den Patterns aus

`http://www.dfki.de/~becker/lispkurs/vorlesungen.ss00/patterns.lsp`

und den Testsätzen aus

`http://www.dfki.de/~becker/lispkurs/vorlesungen.ss00/saetze.lsp`

4 Schmökaufgabe (3 Punkte)

Modifiziere `traverse-list` aus der Vorlesung so, daß es die Übergangs- und Tokentabelle aus der vorherigen Aufgabe benutzt, um Patterns zu erkennen.

Abgabetermin ist Freitag, der 12. Mai 2000