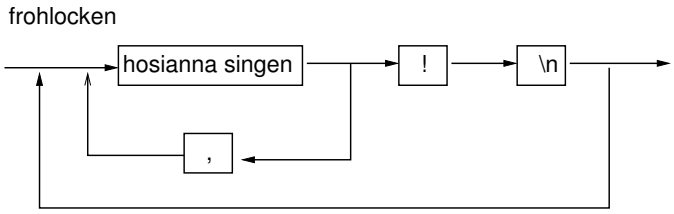
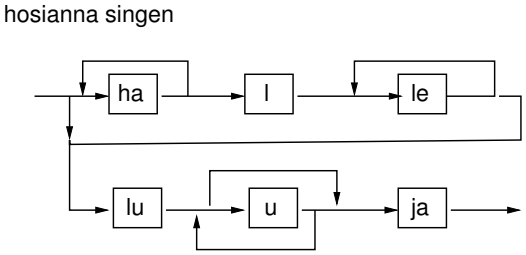
Übungen zur Stringverarbeitung: Blatt 1

Reguläre Ausdrücke (regular expressions)

1. (POSIX-Stil): Originalaufgabe aus dem Bundeswettbewerb Informatik 1990
   Als der Münchner Dienstmann Alois in den Himmel kam, überreichte ihm Petrus eine Harfe und wies ihn in die himmlische Zungensprache Frohlocken ein. Ihre Syntax wird entsprechend dem folgenden Diagramm gebildet





   a) Schreiben Sie ein Programm, das alle Zeilen einer Textdatei überprüft, ob sie gültige Sätze in Frohlocken darstellen. Das Programm soll die richtigen Sätze auf den Bildschirm ausgeben — 5 Punkte

   b) Testen Sie Ihr Programm an der Beispieldatei „froh.txt". Berücksichtigen Sie, dass zusätzlich nach dem Komma beliebig viele Leerzeichen erlaubt sind. Welche Zeilen stellen keine Sätze der angegebenen Grammatik dar und warum? — 2 Punkte

   c) Schreiben Sie ein Programm, das alle gültigen Hosianna-Wörter aus der Beispieldatei heraussucht. Die ganzen Sätze dürfen dabei ungültig sein. — 2 Punkte

   d) Schreiben Sie ein Programm, das alle ungültigen Sätze aus „froh.txt" heraussucht. — 1 Punkte

   Anmerkungen

   a) Als ich die Aufgabe 1990 mit elementarer Programmierung in COMAL (einem BASIC-Dialekt) gelöst habe (Teilaufgabe 1a), hatte meine Lösung ca. 1000 Zeilen, und die Programmierung hat einige Wochen gedauert. Die Musterlösung mit regulären Ausdrücken habe ich in 5 Minuten erstellt, und sie besteht aus 67 Zeichen.

   b) Für Teilaufgabe 1c und 1d können Sie das Verhalten von `egrep` mit Optionen verändern. Der Befehl `egrep -help` gibt Ihnen mögliche Optionen aus.

2. (POSIX-Stil): Sie sind ein böser Spammer und möchten daher so viele E-Mailadressen wie möglich sammeln. Glücklicherweise gibt es das Usenet, wo viele nette Menschen ihre E-Mailadresse zum Antworten im Klartext hinterlassen. Sie haben nun bereits ein

Programm geschrieben, das alle aktuellen Newsgroup-Artikel von einem Archiv-Server aus dem Internet herunterlädt. (Die ersten 100 kB dieses Archivs liegen in der Datei „postings.txt" vor.) Natürlich wollen Sie die Adressen aus dieser Datei nicht von Hand heraussuchen. Schreiben Sie ein Programm, das aus der Datei „postings.txt" alle E-Mailadressen heraussucht. 10 Punkte

Hinweise:

a) Das Problem lässt sich mit einem einzigen regulären Ausdruck im POSIX-Stil lösen. Nutzen Sie das Programm `egrep` mit der Option `-o`, um alle Strings zu suchen, die auf Ihren Ausdruck passen:
`egrep -o 'Ihr RegExp' postings.txt` Die Datei „fiesemail.txt" enthält noch ein paar Beispiele für Adressen nach dem angegebenen Schema.

b) Eine gültige E-Mailadresse besteht aus einem Benutzernamen, gefolgt von '`@`', gefolgt vom Servernamen. *Beispiel:* Die E-Mailadresse `Christian.Gollwitzer@stud.uni-bayreuth.de` hat als Benutzernamen `Christian.Gollwitzer` und als Servernamen `stud.uni-bayreuth.de`

Der Benutzername darf dabei aus Ziffern, großen und kleinen Buchstaben des lateinischen Alphabets und den folgenden Sonderzeichen bestehen: `._-`. Er muss mit einem Buchstaben oder einer Ziffer beginnen und darf maximal 30 Zeichen lang sein und muss mindestens 3 Zeichen lang sein.

Der Servername setzt sich zusammen aus den Subdomain-Namen und der Top-Level-Domain (TLD), die durch Punkte getrennt sind. Im Beispiel ist die Top-Level-Domain `de`, die Second-Level-Domain `uni-bayreuth` und die Third-Level-Subdomain `stud`. Sie wollen englischen Spam verschicken; daher ist für die TLD nur einer der folgenden Namen zugelassen: `com, net, gov, uk, de, cn, tw, eu`. Sie dürfen der Einfachheit halber annehmen, dass die TLD kleingeschrieben ist. Die Subdomain-Namen müssen mindestens 3 Zeichen lang sein und dürfen nach dem ersten Zeichen auch Ziffern und die Sonderzeichen `_-` enthalten. Insgesamt soll der Servername aus nicht mehr als 4 solchen Teilen bestehen. Beispiele für gültige Servernamen:
`spam.newsserv.uni-bayreuth.de, uni-erlangen.de, tla.com`
Beispiele für ungültige Servernamen:

i. `spam1.spam.newsserv.uni-bayreuth.de`: 5 Teile
ii. `neuserver.mödlareuth.com`: Umlaute in der 2LD
iii. `christian.bt.de`: 2LD zu kurz

3. (Shell-Stil): In einem Verzeichnis befinden sich Dateien, die mit `up1.txt` bis `up299.txt` benannt sind (ohne führende Nullen). Zusätzlich gibt es Dateien mit den Nummern 2000 bis 2009.

a) Schreiben Sie ein Shell-Skript, das die Namen aller Dateien mit Nummern größer als 2000 ausgibt (2000 ausgenommen) 1 Punkte

b) Schreiben Sie ein Shell-Skript, das Dateien mit Nummern kleiner gleich 250 ausgibt. *Hinweis:* Sie dürfen mehrere Muster verwenden 4 Punkte

c) Löschen Sie alle Dateien mit Nummern $n \geq 250$ 5 Punkte

Abgabe bis 3. Mai 2006 per E-Mail an `Christian.Gollwitzer@uni-bayreuth.de`