Michael Arndt
Fritz Hamm
Peter Schroeder-Heister

Sommersemester 2003
Übungsblatt 10
9.7.2003

Übungen zur Vorlesung *Logikprogrammierung*

Aufgabe 1 Sequenzen (3 Punkte)

Wenn $X_1, \ldots, X_m$ und $Y_1, \ldots, Y_n$ aussagenlogische Formeln sind, dann ist $X_1, \ldots, X_m \Rightarrow Y_1, \ldots, Y_n$ eine aussagenlogische *Sequenz*.

Erweitern Sie die Prädikate `lexan/2` und `parse/2`, so daß aussagenlogische Sequenzen lexikalisch analysiert und geparst werden können. Gestalten Sie dabei das Prädikat `parse/2` derart, daß das äußerste Klammerpaar einer komplexen Formel nicht unbedingt vorhanden sein muß.

Hinweise: Der Zeichenkombination “`=>`” kann z.B. das Token `seqT` und dem Zeichen “`,`” das Token `komT` zugewiesen werden. Das Ergebnis des Parsings sollen zwei Listen funktionaler Terme sein, eventuell durch ein Funktionszeichen in einen Term zusammengefaßt.

```
?- lexan("A, B => A&B",L).
L = [aT, komT, bT, seqT, aT, konT, bT]
Yes
?- parse([aT, komT, bT, seqT, aT, konT, bT],S).
S = [a,b]/[kon(a,b)]
Yes
```

Aufgabe 2 Primzahlen (3 Punkte)

Definieren Sie ein Prädikat `prim(N,L)`, das in `L` die Liste der ersten `N` Primzahlen konstruiert.