Um die folgenden Aufgaben zu lösen, müssen Sie zunächst das Kapitel 1 - 3 im Prolog-kristina-Tutorial http://www.coli.uni-saarland.de/%7Ekris/learn-prolog-now/lpnpage.php?pageid=online durcharbeiten.

Bitte bringen Sie zum KI-Kurs am 28.06.2007 einen Laptop mit, um die Lösungen der folgenden Aufgaben mit SWI-Prolog vorzuführen.

Aufgabe 1 (exercise 2.3 from the kristina-tutorial)

Gegeben sind ein kleines Lexikon und eine Mini-Grammatik mit nur einer einzigen Regel. Die Regel definiert einen Satz mit fünf Wörtern: einem Artikel, einem Nomen (noun), einem Verb und wieder einem Artikel und einem Nomen.

```
word(article,a).
word(article,every).
word(noun,criminal).
word(noun,'big kahuna burger').
word(verb,eats).
word(verb,likes).

sentence(Word1,Word2,Word3,Word4,Word5) :-
   word(article,Word1),
   word(noun,Word2),
   word(verb,Word3),
   word(article,Word4),
   word(noun,Word5).
```

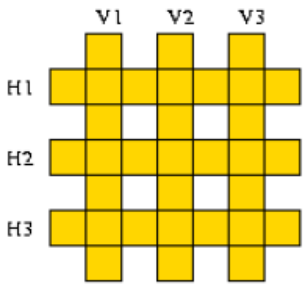
Welche Anfrage muss formuliert werden, um herauszufinden, welche Sätze die Grammatik generieren kann? Liste alle Sätze auf, welche die Grammatik generieren kann und zwar in der Reihenfolge, die durch die Prolog-Inferenz bestimmt wird.

Aufgabe 2 (exercise 2.4 from the kristina-tutorial)

Gegeben sind 6 englische Wörter:

`abalone, abandon, anagram, connect, elegant, enhance`

die in ein Kreuzwort-Puzzle eingetragen werden sollen.



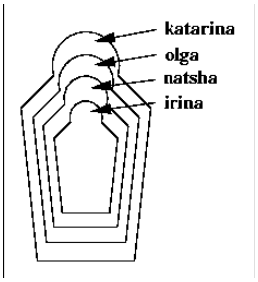
Die folgende Wissensbasis repräsentiert ein Lexikon, das diese Wörter enthält:

```
word(abalone,a,b,a,l,o,n,e).
word(abandon,a,b,a,n,d,o,n).
word(enhance,e,n,h,a,n,c,e).
word(anagram,a,n,a,g,r,a,m).
word(connect,c,o,n,n,e,c,t).
word(elegant,e,l,e,g,a,n,t).
```

Schreiben Sie ein Prädikat `crossword/6`, welches herausfindet, wie das Puzzle zu füllen ist. Die ersten drei Argumente von `crossword/6` sollten die horizontalen Wörter von oben nach unten und die nächsten drei Argumente die vertikalen Wörter von links nach rechts sein.

Aufgabe 3 **(** exercise 3.1 from the kristina-tutorial)

Schreiben Sie eine Wissensbasis für russische Puppen; verwenden Sie dazu das Prädikat `directlyIn/2`.



Definieren Sie ein rekursives Prädikat `in/2`, welches prüft, welche Puppe in welcher (direkt oder indirekt) enthalten ist. Beispiel: `in(katarina, natasha)` evaluiert zu true, `in(olga, katarina)` führt zum fail.

Aufgabe 4 **(** exercise 3.3 from the kristina-tutorial)

Schreiben Sie eine Wissensbasis für binäre Bäume. Blätter werden durch `leaf(Label)` repräsentiert, z.B. `leaf(3)` und `leaf(7)`. Ein Blatt repräsentiert auch den leeren Baum. Gegeben seinen zwei binäre Bäume `B1` und `B2`, die mit dem Prädikat `tree(B1, B2)` kombiniert werden. Wir können einen Baum mit `tree(leaf(1), leaf(2))` konstruieren. Aus den Bäumen `tree(leaf(1), leaf(2))` und leaf(4) wird mit `tree(tree(leaf(1), leaf(2)), leaf(4))` ein neuer Baum konstruiert.

Definiere ein Prädikat `swap/2`, welches einen binären Baum spiegelt, zum Beispiel:

```
?- swap(tree(tree(leaf(1), leaf(2)), leaf(4)),T).
T = tree(leaf(4), tree(leaf(2), leaf(1))).
yes
```

Aufgabe 5 **(** exercise 3.4 from the kristina-tutorial)

Das Prädikat `descend/2` (abstammen von) lautet:

```
descend(X,Y) :- child(X,Y).
descend(X,Y) :- child(X,Z), descend(Z,Y).
```

Kann das Prädikat auch wie folgt formuliert werden?

```
descend(X,Y) :- child(X,Y).
descend(X,Y) :- descend(X,Z), descend(Z,Y).
```

Fakten zum Prädikat `descend/2`:

```
child(martha,charlotte).
child(charlotte,caroline).
child(caroline,laura).
child(laura,rose).
```

Vergleichen Sie die deklarative und die prozedurale Bedeutung dieser Prädikat-Definition.

Hinweis: Was passiert bei der Anfrage `descend(rose, martha)`?

Aufgabe 6 **(** exercise 3.5 from the kristina-tutorial)

Folgende Wissensbasis ist gegeben:

```
directTrain(forbach,saarbruecken).
directTrain(freyming,forbach).
directTrain(fahlquemont,stAvold).
directTrain(stAvold,forbach).
directTrain(saarbruecken,dudweiler).
directTrain(metz,fahlquemont).
directTrain(nancy,metz).
```

Schreiben Sie ein rekursives Prädikat `travelBetween/2`, das feststellt, ob zwischen zwei Städten eine (direkte oder indirekte) Zugverbindung besteht. Die Anfrage

```
travelBetween(nancy, saarbruecken).
```

sollte `yes` liefern.

Erweitern Sie das Prädikat `travelBetween/2` so, dass wenn eine Verbindung von A nach B existiert auch eine Verbindung von B nach A existieren soll. Zum Beispiel soll die Anfrage

```
travelBetween(saarbruecken, nancy).
```

ebenfalls `yes` liefern.

Welche Probleme können dabei auftreten?