Universität des Saarlandes
Fachrichtung 6.2 – Informatik
J. Siekmann, S. Autexier, K. Fischer, E. Melis, C. Zinn
M. Schiller, E. Schulz, R.Vollmann

6th Theoretical Assignment in Artificial Intelligence (WS 2006/2007)

Issued: November, 30 2006

Exercise 6.1 Consider the following axioms in first-order logic (involving the natural number 0, the successor function s that assigns each number its successor, addition $+$ and the property that a number is *even*).

- $x + 0 = x$
- $x + s(y) = s(x + y)$
- $even(0)$
- $\forall x(even(x) \Rightarrow even(s(s(x)))$
- $2 = s(s(0))$

In the following, you will show that $\forall x(even(x) \Rightarrow even(x + 2))$.

1. Transform the problem to clause normal form (CNF), such that you can apply resolution to it!
2. Use resolution with paramodulation to show that $\forall x(even(x) \Rightarrow even(x + 2))$! Indicate each step.

Exercise 6.2
In Russel/Norvig (pages 298-300) a resolution proof for the query *Did Curiosity kill the cat?* is given. In the following, the clauses are given in a slightly different manner.

1. (a) $Dog(D)$
   (b) $Own(Jack, D)$
2. $\neg Dog(y) \vee \neg Own(x, y) \vee AnimalLover(x)$
3. $\neg AnimalLover(x) \vee \neg Animal(y) \vee \neg Kills(x, y)$
4. $Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna)$
5. $Cat(Tuna)$
6. $\neg Cat(x) \vee Animal(x)$
7. $\neg Cat(x) \vee \neg Kills(Curiosity, x)$

For both resolution strategies below, give a resolution proof.

- Unit preference strategy
- Set of support strategy (Choose the clause 7 as Set of Support)

Exercise 6.3 In this exercise, we are concerned with the encoding of rules in a production system. The domain we chose are horses and variants. For these we can define the following, although very simplified, circumstances:

> *If a car is faster than a Porsche then it is a sportscar. It is also a sportscar if it is faster than another sportscar. If X is faster than Y and Y is faster than Z, then X is faster than Z. Furthermore, a car that is faster than a Porsche and a Ferrari is a Formula-1-car.*

(a) List predicates that are required to encode the above facts (e.g., $Faster(X, Y)$)

(b) Encode the above facts as production rules.

Exercise 6.4

In this exercise we prepare the construction of a RETE-network. To this end we must identify subformulas that occur frequently in the preconditions of the production rules. Then we introduce new predicates to abbreviate these subformulas. For instance, if $H(X) \wedge P(X, Y)$ is a subformula that occurs in different production rules, then we define a predicate $B(X, Y)$ by setting $B(X, Y) := H(X) \wedge P(X, Y)$.

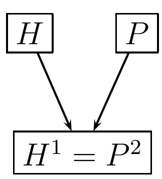
(a) There is a multiply occurring subformula of the production rules from the previous exercise. Identify this subformula.

(b) Introduce a new predicate for the identified subformula and reformulate the production rules using this new predicate.

In the next exercise we will see that using this predicate simplifies the construction of a RETE network.

Exercise 6.5 Construct a RETE-network using the simplified production rules you obtained in the previous exercise. Note that in a RETE-network only predicates occur, but no literals. To represent a subformula $H(X) \wedge P(Y, X)$, where the variable X occurs in two different literals, we use a notation which expresses that the respective arguments of H and P must be equal. For instance we represent the above formula by $H^1 = P^2$. Assume, the literals $H(X)$ and $P(Y, Z)$ are already represented as nodes in the network:

Then the node for the formula $H(X) \wedge P(Y, X)$ is represented by



Construct a RETE-network using that notation for the production rules you obtained in the previous exercise.