BRSU

# Homework for Artificial Intelligence for Robotics
# Assignment 11

Bastian Lang

June 22, 2015

## 1 Task 1

*What are the performance differences between Backward and Forward Chaining?*
Forward Chaining is data-driven and Backward Chaining is goal-driven. The complexity of Backward Chaining can be much less than linear in the size of KB.

## 2 Task 2

*Can the following sentence be expressed in propositional logic, considering all real numbers?*

- *Negative numbers don't have square roots.*
- *Every positive number has exactly two square roots.*

To make a statement for every positive or negative number in propositional logic one would in theory have to enumerate every possible number and make a statement. Because the real numbers are non-denumerable it is not possible to express the above sentences using propositional logic.

## 3 Task 3

*Represent the following sentences in first-order logic, using a consistent vocabulary (which you must define):*

- *(a) A person born in the UK, each of whose parents is a UK citizen or a UK resident, is a UK citizen by birth.*
- *(b) A person born outside the UK, one of whose parents is a UK citizen by birth, is a UK citizen by descent.*
- *(c) Politicians can fool some of the people all of the time, and they can fool all of the people some of the time, but they can not fool all of the people all of the time.*
- *(d) There is a barber who shaves all men in the town who do not shave themselves.*
- *(e) The best score in Greek is always higher than the best score in French.*

## 3.1 (A)

Let $P_i$ denote a person, $born\_in\_uk(P_i)$ evaluates true iff person $P_i$ is born in the UK, $citizen\_of\_uk(P_i)$ iff a person is citizen of the UK, $resident\_of\_uk(P_i)$ iff a person is a resident of the UK and $parent(P_i, P_j)$ iff $P_i$ is parent of $P_j$.
Then (a) can be expressed as:
$\forall P_1, P_2, P_3 :$
$(born\_in\_uk(P_3) \land parent(P_1, P_3) \land parent(P_2, P_3)$
$\land (citizen\_of\_uk(P1) \lor resident\_of\_uk(P1))$
$\land (citizen\_of\_uk(P_2) \lor resident\_of\_uk(P_2))$
$\rightarrow citizen\_of\_uk\_by\_birth(P_3))$

## 3.2 (B)

$\forall P_1, P_2, P_3 :$
$(\neg born\_in\_uk(P_3) \land parent(P_1, P_3) \land parent(P_2, P_3)$
$\land (citizen\_of\_uk\_by\_birth(P_1) \lor citizen\_of\_uk\_by\_birth(P_2)))$
$\rightarrow citizen\_of\_uk\_by\_descent(P_3))$

## 3.3 (C)

## 3.4 (D)

Let Barber(x) be true iff x is a barber, inTown(x) if x lives in town, man(x) iff x is a man and shaves(x,y) iff x shaves y. Then:

$\exists x : barber(x) \land inTown(x) \land \forall y : man(y) \land inTown(y) \land ((shaves(x, y) \land \neg shaves(y, y)) \lor shaves(y, y))$

### 3.5 (E)

## 4 Task 4

*PROLOG is a programming language commonly used for logic. It uses first-order logic. Learn some basic PROLOG 1 and write axioms describing the following relationships: Child, Grandchild, GreatGrandparent, Brother, Sister, Daughter, Son, Aunt, Uncle*

```
parent(john, mary).
parent(mary, william).
parent(mary, emma).
parent(william, sophia).
parent(william, ryan).
parent(william, jacob).
parent(emma, joseph).
parent(emma, olivia).
parent(emma, emily).
female(mary).
female(sophia).
female(emma).
female(olivia).
female(emily).
male(john).
male(william).
male(ryan).
male(jacob).
male(joseph).
child(X,Y) :- parent(Y,X).
grandparent(X,Y) :- parent(X,Z), parent(Z,Y).
brother(X,Y) :- parent(Z,X), parent(Z,Y), male(X), X \=Y.
sister(X,Y) :- parent(Z,X), parent(Z,Y), female(X), X \= Y.
uncle(X,Y) :- brother(X,Z), child(Y,Z).
aunt(X,Y) :- sister(X,Z), child(Y,Z).
daughter(X,Y) :- child(X,Y), female(Y).
son(X,Y) :- child(X,Y), male(X).
grandchild(X,Y) :- child(X,Z), child(Z,Y).
greatgrandparent(X,Y) :- parent(X,Z), parent(Z,A), parent(A,Y).
```