Übung 2 Künstliche Intelligenz

Julian Dobmann, Kai Kruschel

Aufgabe 1

a)

```
% If you work hard then you get lucky
fof(a0,axiom,
    ( ! [X]:
        (works_hard(X)
        => gets_lucky(X)))).

% Either you get lucky or you work hard, or both
fof(a1,axiom,
    ( ! [X]:
        (gets_lucky(X) | works_hard(X)))).

% If you get lucky then, either you are not a rogue or you work hard (but
not both)
fof(a2, axiom,
    ( ! [X]:
        gets_lucky(X) =>
            (~rogue(X) <~> works_hard(X)))).

% You are a rogue
fof(a3, axiom,
    (rogue(me))).

% Conjecture: You work hard
fof(c0, conjecture,
    (works_hard(me))).
```

b)

```
% Garfield is a cat
fof(a0,axiom,
    (cat(garfield))).

% Odie is a dog
fof(a1,axiom,
    (dog(odie))).

% Cats and dogs are animals
```

```
fof(a2,axiom,
    ( ! [X]:
        cat(X) => animal(X))).
fof(a3,axiom,
    ( ! [X]:
        dog(X) => animal(X))).

% Jon is a human
fof(a4,axiom,
    (human(jon))).

% Every animal has a human owner
fof(a5,axiom,
    ( ! [X]:
        animal(X) => ? [Y]:
            is_owner_of(Y,X))).

% Jon is the owner of Garfield and Odie
fof(a6,axiom,
    (is_owner_of(jon,garfield))).
fof(a7,axiom,
    (is_owner_of(jon,odie))).

% Garfield and Odie are the only animals that Jon owns
fof(a8,axiom,
    ( ! [X]:
        (is_owner_of(jon,X)
            => ( X = odie
               | X = garfield)))).

% If a cat is chased by a dog, then the owner of the cat hates the owner of
the dog
fof(a9,axiom,
    ( ! [X,Y]:
        (cat(X) & dog(Y) & chased(Y,X))
            => (! [Q,P]:
                (is_owner_of(Q,X) & is_owner_of(P,Y))
                    => hates(Q,P)))).

% Odie has chased Garfield
fof(a10,axiom,
    (chased(odie,garfield))).

% Conjecture:Jon hates himself
fof(c0, conjecture,
    (hates(jon,jon))).
```

Aufgabe 2

a)

```
myLast([Item], Item).
myLast([ _ | Rest], LastItem) :-
  myLast(Rest, LastItem).
```

b)

```
myMax([], _).
myMax([Item | Rest], MaxItem) :-
  Item =< MaxItem,
  myMax(Rest, MaxItem).
```

c)

```
mySum([X], Y):-
  X is Y.
mySum([Head | Tail], X):-
  mySum(Tail, (X - Head)).
```

d)

```
# myOrder ufert ein bisschen aus...
myOrder([X, Y]):-
  X >= Y.

myOrder([X, Y]):-
  X =< Y.

myOrder([X, Y | Rest]):-
  X >= Y,
  myOrderGreater(Y | Rest).

myOrder([X, Y | Rest]):-
  X =< Y,
  myOrderLesser([Y | Rest]).

myOrderGreater([X | [Y]]):-
  X >= Y.

#Aus irgendeinem Grund wird das immer zu false ausgewertet
myOrderGreater([X, Y | Rest]):-
  X >= Y,
  myOrderGreater([Y | Rest]).

myOrderLesser([X | [Y]]):-
  X =< Y.

myOrderLesser([X, Y | Rest]):-
  X =< Y,
  myOrderLesser([Y | Rest]).
```