Knowledge Representation & Symbolic Reasoning Assignment 2: First Order Logic and Prolog

Henk Jekel (5609593)

June 27, 2024

---

1 Modelling and reasoning in propositional logic

1.1 First order logic

1. "Our robot has grabbed a package of hagelslag from a table. There is also a can of coke on the table. Both a can of coke and a hagelslag package are items."

   - $grabbed(robot, hagelslag) \land at\_location(hagelslag, table)$
   - $at\_location(coke, table)$
   - $item(hagelslag) \land item(coke)$

2. "All items that are full should be stored in the shelf. All items that are empty should be discarded in the bin. An item that is heavy is full."

   - $\forall x, (item(x) \land is\_full(x)) \rightarrow should\_be\_stored\_in\_shelf(x)$
   - $\forall x, (item(x) \land is\_empty(x)) \rightarrow shoud\_be\_discarded\_in\_bin(x)$
   - $\forall x, (item(x) \land is\_heavy(x)) \rightarrow is\_full(x))$

3. "If there is room in the bin and the robot has grabbed an item, then the robot can discard it. If there is room in the shelf and the robot has grabbed an item, then the robot can store it."

   - $\forall x, (has\_room(bin) \land grabbed(robot, x) \land item(x)) \rightarrow can\_be\_discarded(x)$
   - $\forall x, (has\_room(shelf) \land grabbed(robot, x) \land item(x)) \rightarrow can\_be\_stored(x)$

1.2

To represent the fact that there are multiple shelves where items can be stored, we can modify the predicate should_be_stored_in_shelf/1 to a new predicate should_be_stored_in/2, where the first argument represents the item and the second argument represents the shelf where the item should be stored. This new predicate should replace the previous predicate should_be_stored_in_shelf/1.

The arity of the new predicate should_be_stored_in/2 is 2, where the first argument is an item and the second argument is a shelf.

The modified rule using the new predicate should_be_stored_in/2 would be:

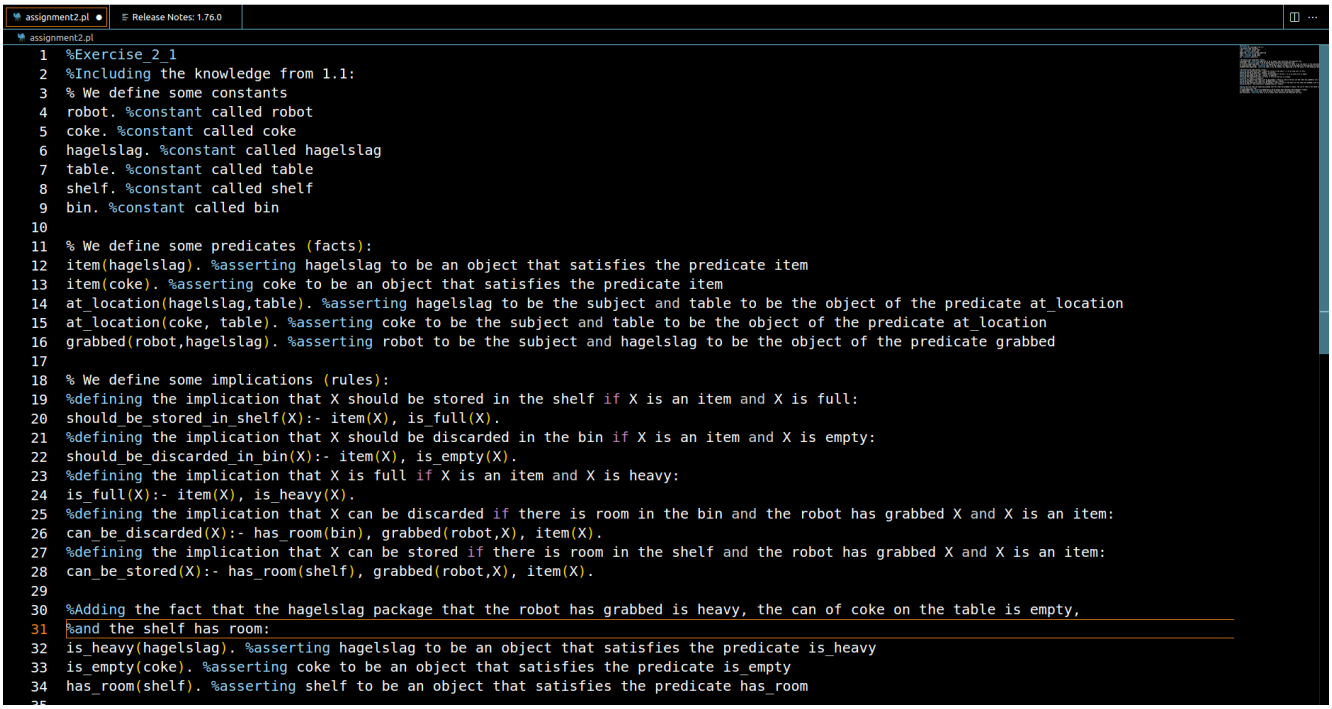
- $\forall x, \forall shelf, (has\_room(shelf) \land grabbed(robot, x) \land item(x)) \rightarrow can\_be\_stored(x, shelf)$

Where $shelf$ now represents any of the shelves where an item can be stored.

2 Prolog

2.1 programming your knowledge base in Prolog

The prolog code is attached in brightspace but also displayed in figure 1.



```
%Exercise_2_1
%Including the knowledge from 1.1:
% We define some constants
robot. %constant called robot
coke. %constant called coke
hagelslag. %constant called hagelslag
table. %constant called table
shelf. %constant called shelf
bin. %constant called bin

% We define some predicates (facts):
item(hagelslag). %asserting hagelslag to be an object that satisfies the predicate item
item(coke). %asserting coke to be an object that satisfies the predicate item
at_location(hagelslag,table). %asserting hagelslag to be the subject and table to be the object of the predicate at_location
at_location(coke, table). %asserting coke to be the subject and table to be the object of the predicate at_location
grabbed(robot,hagelslag). %asserting robot to be the subject and hagelslag to be the object of the predicate grabbed

% We define some implications (rules):
%defining the implication that X should be stored in the shelf if X is an item and X is full:
should_be_stored_in_shelf(X):- item(X), is_full(X).
%defining the implication that X should be discarded in the bin if X is an item and X is empty:
should_be_discarded_in_bin(X):- item(X), is_empty(X).
%defining the implication that X is full if X is an item and X is heavy:
is_full(X):- item(X), is_heavy(X).
%defining the implication that X can be discarded if there is room in the bin and the robot has grabbed X and X is an item:
can_be_discarded(X):- has_room(bin), grabbed(robot,X), item(X).
%defining the implication that X can be stored if there is room in the shelf and the robot has grabbed X and X is an item:
can_be_stored(X):- has_room(shelf), grabbed(robot,X), item(X).

%Adding the fact that the hagelslag package that the robot has grabbed is heavy, the can of coke on the table is empty,
%and the shelf has room:
is_heavy(hagelslag). %asserting hagelslag to be an object that satisfies the predicate is_heavy
is_empty(coke). %asserting coke to be an object that satisfies the predicate is_empty
has_room(shelf). %asserting shelf to be an object that satisfies the predicate has_room
```

Figure 1: Prolog code.

2.1.1 Which item(s) can be stored in the shelf?

To answer the first question, we can use the following Prolog query:

```
?- should_be_stored_in_shelf(Item).
```

This query will find out which item(s) can be stored in the shelf. The output is:

```
Item = hagelslag.
```

The output tells us that only the hagelslag package can be stored in the shelf.

2.1.2 Which item(s) should be discarded in the bin and at which location(s) are they?

To answer the second question, we can use the following Prolog query:

```
?- should_be_discarded_in_bin(Item), at_location(Item, Location).
```

This query will find out which item(s) should be discarded in the bin and their locations. The output should be:

```
Item = coke,
Location = table.
```

The output tells us that the can of coke on the table should be discarded in the bin and its location is the table.