MCS Project: Parts 2 and 3
Warehouse Robots

Lennart de Graef, r0297214
Jonas Schouterden, r0260385

December 20, 2014

Design decisions

Part 2: Modelling the dynamic system in IDP

We started from the code of the first part of the project, which we adapted to the linear time calculus by making the predicates hold for every moment in time.

We have introduced one new function symbol: *heightOf.* This function maps each pallet onto a height:

- *heightOf* is defined inductively. We state that a pallet on the floor (which has no pallet below it and is not being carried,) has a height of 1. If a pallet is on top of another pallet, it's height is one higher.

Furthermore, we introduced to new predicates which describe the location in front and behind the robot at each moment in time:

- *locationInFront(time, location)* is true if and only if *location* is in front of the robot at time point *time*;
- *locationBehind(time, location)* is true if and only if *location* is in behind of the robot at time point *time*.

We also defined new dynamic predicates. The first group of predicates is about the direction the robot faces:

- *C_Facing(time, dir)* is true if and only if there is an action at time point *time* which causes the robot to face the direction *dir* in the next time point;
- *C_NFacing(time, dir)* is true if and only if there is an action at time point *time* which causes the robot **not** to face the direction *dir* in the next time point;
- *I_Facing(dir)* is true if and only if the robot is facing direction *dir* at the initial time point.

The second group of dynamic predicates has to do with the position of the robot:

- *C_Robotposition(time, location)* is true if and only if there is an action at time point *time* which causes the robot to have position *location* in the next time point.
- *C_NRobotposition(time, location)* is true if and only if there is an action at time point *time* which causes the robot **not** to have position *location* in the next time point.
- *I_Robotposition(location)* is true if and only if the robot has position *location* at the initial time point.

The third group of dynamic predicates is about the robot carrying a pallet:

- *C_Carried(time, pallet)* is true if and only if there is an action at time point *time* which causes the pallet *pallet* to be carried by the robot in the next time point;
- *C_NCarried(time,pallet)* is true if and only if there is an action at time point *time* which causes the pallet *pallet* **not** to be carried by the robot in the next time point;
- *I_Carried(pallet)* is true if and only if the robot carries pallet *pallet* at the initial time point.

Another group of predicates describes the causes for a pallet to be be on top op of another pallet:

- *C_On(time, pallet1, pallet2)* is true if and only if there is an action at time point *time* which causes the pallet *pallet1* to be positioned on top of pallet *pallet2* in the next time point.
- *C_NOn(time, pallet1, pallet2)* is true if and only if there is an action at time point *time* which causes the pallet *pallet1* **not** to be positioned on top of pallet *pallet2* in the next time point.
- *I_On(pallet1, pallet1)* is true if and only if pallet *pallet1* is positioned on top of pallet *pallet2* at the initial time point.

The last group of dynamic predicates has to do with the location of pallets:

- *C_Position(time, pallet, location)* is true if and only if there is an action at time point *time* which causes the pallet *pallet1* to be positioned at position *location* in the next time point.
- *C_NPosition(time, pallet, location)* is true if and only if there is an action at time point *time* which causes the pallet *pallet1* **not** to be positioned at position *location* in the next time point.
- *I_Position(pallet, location)* is true if and only if pallet *pallet* is positioned at position *location* at the initial time point.

Part 2: Modelling the dynamic system in NuSMV

For the NuSMV part, we started from the given skeleton. We didn't add a new module, so we have three modules defined in our solution: *main*, *robot* and *pallet.*

The module main

The module main defines three variables:

- *robot*: this is our robot. We define at as *robot(1,3, pallet1, pallet2).* this way, it knows about the two pallets.
- *pallet1* and *pallet2*: respectively. We give them an initial position, the name of the robot, a boolean which is true if the pallet is top of a stack of pallets and a boolean which is true when the pallet is being carried.

Furthermore, *main* contains a couple of invariants and transition descriptions.

The module robot

A robot has five variables:

- *x*: the x-coordinate of the robot
- *y*: the y-coorcinate of the robot
- *orientation*: the orientation of the robot
- *action*: the current action of the robot. This action will have taken effect in the state (time step)
- *carriedPallet*: this variable can have one of three values and describes the pallet that the robot carries. The value can be *p1*, emphp2 or emphnone, for pallet1, pallet2 or no pallet respectively.

The robot also has a couple of invariants on the actions it can execute.

In the *assignment*- block, *next*-functions for four of the five variables of robot are defined. Only for *action*, there is no *next*-function defined. The necessary constraints on the a *action*-variable are enforced by the invariants.

In the *DEFINE*-block, a some auxiliary variables are defined.

- The variables *x_in_front*, *y_in_front*, *x_behind* and *y_behind* describe the locations in front of and behind the robot.
- *allowedToMoveForward* and *allowedToMoveBackward* are boolean variables which values describe if a robot can move forward or backward.
- The variable *palletInFront* has as value the pallet in front of the robot ($p1$, $p2$ or *none*).

The module pallet

A pallet has four variables:

- *x*: the x-coordinate of the pallet
- *y*: the y-coorcinate of the pallet

- *onTopOfStack*: a boolean variable which is true if the pallet is on top of a stack of pallets.
- *beingCarried*: a boolean variable which is true if the pallet being carried by the robot.

In the *assignment*- block, *next*-functions for the four variables of pallet are defined.
In the *DEFINE*-block, one auxiliary variable is defined. The boolean variable *thisPalletInFrontOfRobot* is true if and only if the pallet is located on the position in front of the robot.

Time spent on the project

Our time spend on this part of this project is thirty-three hours each. Most of the time spent was on fixing bugs.