FMCS – 2019
Project Statement

mehrnoosh.askarpour@polimi.it

You need to create a model of a robotic system involving a mobile robot and an operator, working in a workcell as the one shown in Figure 1. The model needs to replicate the movements and positioning of human and robot, and should verify if any collision between them occurs at some point.

The model should be built via TRIO temporal language, and verified via Zot. This document explains the specification of the system and provides instructions to use Zot.

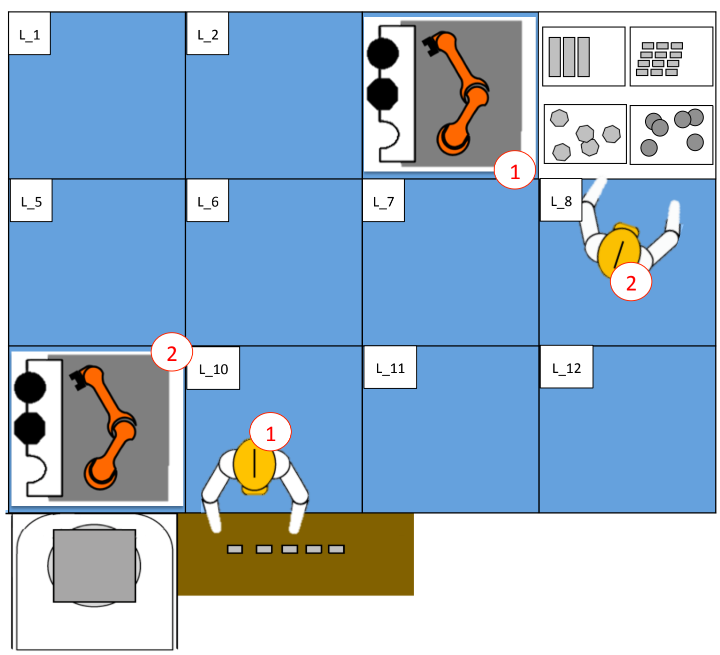

Figure 1: An abstract and discretized representation of the workcell. The operator and robot frequently move between L_10 and L_8, and L_3 to L_9 respectively.

Modeling Positioning

In order to model the positioning and movements in the workcell, we discretize it into 12 areas, labeled in Figure 1. At each time instant, the robot and human occupy exactly one of these areas. Assume that it is also physically feasible for them to co-exist in a single area. Use predicates in order to express where the robot or operator currently are. For example, you could define 12 predicates in form "robot_in_L_x" and 12 others as "operator_in_L_x". If at time t, robot is in L_1, then "robot_in_L_1" is *True*, while all the other 11 predicates are *False*.

Note that if for the same area x, both "robot_in_L_x" and "operator_in_L_x" are true, it means a collision between human and robot has just happened.

NB: We assume that there are precise sensors mounted in the workcell that function properly and give us the correct position of each element at each t (meaning that the positions are known at each t).

NB: Do not use any coordinate system for representing the layout and stick with the introduced areas.

Modeling Movements

Moving means changing the occupying area. For example if at time t, the operator is in L_11 and at t+1 is in L_10, it means that the operator has moved.

However, there should be constraints that render the movements realistic. For example, if an entity is in L_1 at t, it can not be in L_12 at t+1. Thus, the model needs formulae that express which moves are possible from each area. For example, L_2, L_5 and L_6 are reachable from L_1.

The robot should avoid any physical obstacle, including human, by stopping or changing its trajectory. In case of stopping, it can resume its move as soon as the obstacle has moved away. The choice between turning around and stopping depend on the current positions, the moving state of human, and proximity to the destination. For example, if the robot has just arrived in L_5 and human is in L_9 and not moving, it is better for the robot to stop and wait for the operator to move, rather than moving around L_9 non-stop (getting into a useless loop).

Both robot and human cannot enter L_4 where the bin of work-pieces is installed (only their arms can enter the area in order to grab workpieces which is not required to be stated in your model).

NB: Do not go into the details of directions for modeling the movements.

The System Task

The robot iteratively grabs three work-pieces and places them in the local-bin mounted on its cart, moves to the pallet in the bottom left corner of the workcell, manipulates them in certain way and then places them on the brown conveyor belt. Then operator inspects the shape of the work-pieces on the belt and take them back to the bin and places them in the bin of processed pieces (the one on top right where robot arm doesn't reach).

The initial configuration at the first iteration is operator being in L_10 and robot is in L_3.

In order to create an exhaustive model, you do not have to enforce a certain trajectory to the robot or operator. It means that, the robot can take any possible trajectory from L_3 to L_9 including {L_3, L_2,L_1,L_5,L_9} or {L_3, L_7, L_6, L_10, L_9}, always avoiding obstacles. The same hold for human as well.

NB: The only required part of the task to model is the movements. The "bin picking" and "piece manipulation" is not required.

Defining the Property to verify

The model (M) needs to be verified against the following property(P): "Human and robot are never in the same place while the robot is moving". The model ($M \wedge \neg$ P) should be unsatisfiable.

Using Zot

The Zot tool is a bounded model checker that works on TRIO models. The tool is implemented via Lisp, which means that your models should be consistent with the parenthesized prefix notion of lisp language. Thus, after building your model you need to slightly reshape the formulae and then feed them to the tool. For example, $Since(a, b)$ is acceptable for zot as ($Since\ a, b$). Also a predicate or variable named x is used as (-P- x) or (-V- x). Please find examples and guides here.

In order to install and run the tool follow the instructions below:

- Install docker on your pc (https://docs.docker.com/install/).
- Clone https://github.com/Askarpour/FM2019.
- Create a private git repository and add me as collaborator.
- Add all members of your group to the repository.
- Change the remote url of the cloned directory to your own repo:

```
git remote set-url origin <your-repo-url>
```

- You can create your own files (.lisp) and run them as the following:

```
./run.sh zot filename.lisp
```

  or analyze your outputs by

```
./run.sh z3 output.smt.txt
```

- Commit all your changes to your repo before the deadline.

What to turn in

No emails are accepted. Every group should commit their files on a git repository (see above).

- A PDF file containing the whole TRIO specification with a minimum of explanation + safety Property.
- lisp files executable for Zot.

Deadline

Friday, June 14th.