Piecewise Constant Navigation Benchmark

Alessandro Cimatti, Alberto Griggio, Sergio Mover, Stefano Tonetta

December 18, 2016

1 Introduction

The benchmarks is inspired by the classic navigation benchmark from [?], but the goals are quite different.

The proposed benchmark differs from the one in [?] for the kind of dynamics (piecewise constants and not affine) used to describe the motion of the vehicles in each cell and also the coordinate systems used is different.

The benchmark is significant for the following reasons.

1. It contains non-deterministic piece-wise constants dynamics, that change in each discrete location. The class of PWC hybrid automata allow to model non-deterministic dynamics that have several applications in the modeling of real-time systems and as a high level abstraction of hybrid systems.

   The proposed model differs from the other PWC ARCH benchmark [?], which only contains clock variables.

2. Scalable: the complexity of the model can be increased by increasing the number of cells. A scalable model allow us to compare the performance of different verification engines.

We provide the benchmarks in SpaceEx and hydi format in a public github repository: `https://github.com/smover/arch_comp_hydi`

2 Benchmark description

The model describes an object that moves on the 2d plane. The plane is divided into cells by a grid with the same number of rows and columns, as shown in Figure ??. The continuous variables x and y represent the position of the object on the x and y axes.

TODO: add an image of the grid, with the directions

In each cell the object moves following a specified dynamic that constraints the velocity in the x and y directions.

The dynamic can be one among the following:

```
           |     |     |          |    |
      SE   | SE  | SE  |          | SEE| NWW
           |     |     |          |    |
 ----------+-----+-----+--      --+----+-----------------
      NEE  | NEE | NEE |  other   | NEE| NE
           |     |     |  columns |    |
-----------+-----+-----|--      --+----+-----------------
      NEE  | NEE | NEE |          | NEE| NE
           |     |     |          |    |
-----------+-----+-----+--      --+----------------------

                  ... other rows...

           |     |     |          |    |
-----------+-----+-----+--      --+----------------------
      NEE  | NEE | NEE |          | NEE| NE
           |     |     |          |
```

Figure 1: nxn version of the navigation benchmark

- N (North): $\dot{x} = 0, \dot{y} = 1.4$
- NE (Nort-East): $\dot{x} = 1.4, \dot{y} = 1.4$
- E (East): $\dot{x} = 1.4, \dot{y} = 0$
- SE (South-East): $\dot{x} = 1.4, \dot{y} = -1.4$
- S (South): $\dot{x} = 0, \dot{y} = -1.4$
- SW (South-West): $\dot{x} = -1.4, \dot{y} = -1.4$
- W (West): $\dot{x} = -1.4, \dot{y} = 0$
- NW (North-West): $\dot{x} = -1.4, \dot{y} = 1.4$
- NE-E (North-East East), the object moves non-deterministally in the range NE - E: $\dot{x} = 1.4, 0 <= \dot{y}, \dot{y} <= 1.4$
- SE-S (South-East South), the object moves non-deterministally in the range SE - S: $0 <= \dot{x}, \dot{x} <= 1.4, \dot{y} = -1.4$
- NW-W (North-West West), the object moves non-deterministally in the range NW - W: $\dot{x} = -1.4, 0 <= \dot{y}, \dot{y} <= 1.4$
- FREE, the object moves non-deterministically in all the possible direction, with a maximum rate of 1: $-1 <= \dot{x}, \dot{x} <= 1, -1 <= \dot{y}, \dot{y} <= 1$

The objects starts in $x = 0$ and $y = 0$ coordinates. The directions used in the cells are such that the object will eventually reach the 4 top right cells. Then, the object is forced to move inside these cells.

The benchmark is scaled by increasing the size of the grid.

3 Verification problem

- Safety property: the safety property states that after a specific amount of time the object must stay inside the 4 top-right cells.