

Lecture

Topics: Timed Games and UPPAAL-TIGA

We shall have a (brief) look at Timed Games and, in particular, UPPAAL-TIGA. The aim to synthesize control systems on the basis of winning strategies.

Readings

Read the manual for UPPAAL-TIGA. Available from DTU Learn.

Exercise 1: Install and experiment with UPPAAL-TIGA

1. Install UPPAAL-TIGA from <https://uppaal.org/>
2. Open the example from the lecture: `concur05.xml` (available from FileSharing and the UPPAAL-TIGA distribution) and experiment with the system to get at feeling for Timed Games.
3. Use the command: `verifytga -t0 concur05.xml` to synthesize a winning strategy and get a grasp of that strategy.
4. Use of the command: `verifytga -t0 concur05.xml 1> tga.out`, on my Windows system will direct the output from above command to the file `tga.out`.

Exercise 2: Goat/Wolf/Cabbage Puzzle

You shall now have a new look at the puzzle from the first week of the course. The aim is to let UPPAAL-TIGA generate a control strategy for the man, that will bring the three passengers safely from one shore to the other (according to the rules of the puzzle).

1. Model the system as a Timed Game. The starting point could be a solution to the second exercise from Week 1 (for example, one that is available from DTU Learn). Eliminate any intelligence of the man from that solution, that is, the man can make bad as well as good choices.

Extend that model with states, where one of the passengers wins (that is, the man will lose when those states are reached). For example, the wolf wins when it eats the goat.

2. Specify the winning condition of the man as a Pure Reachability condition, and use the verifier to check whether there is a winning strategy for the man.
3. If a winning strategy for the man exists, then use command `verifytga` to generate this strategy.