Katholieke
Universiteit
Leuven

Department of
Computer Science

MCS PROJECT PART 1:SAVING ARKHAM

Alexander Tang - s0189509
Xueying Deng - r0601458

Contents

1 LTCvocabulary extension

Note that the explicit use and context is further elaborated in the commentary provided with each constraint and verification procedure.

- **Connected(location, location):** Extends **Has_Connection(location, location)** relation to represent the symmetrical aspect. **Connected(a,b) = Connected(b,a)**
- **CT_Win(Time):** True if and only if the win condition has been triggered; will cause game to win next turn.
- **CT_Lose(Time):** True if and only if the lose condition has been triggered; will cause game to lose next turn.
- **Can_Move_To(Time,location):** True if and only if the investigator is able to move to the location at that time.
- **Stuck(Time):** True if and only if there are no more available moves for the investigator.
- **Monster_Move(Time, location, district) : monsternb:** Equals the amount of monsters moving from location to district at a given time.
- **Reachable(location, location):** True if and only if the given location is reachable from the other given location. Is used in **verification1proc** only.

2 Design decisions

The theory in the *solution.idp* is also split up in sections 2.1 to 2.6 to easily identify the structure with the assignment.

2.1 Game turns

- CF_Win and CF_Lose don't need to be defined since game states that are won or lost cannot change state anymore.
- $\forall$t : CT_Win(t) $\leftarrow$ #{b[building]: Closed_Gate(t,b)} $\geqslant$ Closed_To_Win.
  Since a player can only close one gate at a time '=' might have been sufficient, but $\geqslant$ is used for completeness (in case the game rules change later).

2.2 The investigator

- $\forall$t[Time] : GameState(t) = player_turn $\Rightarrow$ $\exists$l : Move_To(t,l) $\wedge$ Connected(Location(t), l) $\wedge$ Monsters_In(Next(t), l) $<$ 3.
  This constraint takes care of the following conditions:
  - must move every player turn
  - move to location with less than 3 monsters only
  - move to connected location

2.3 Monsters

- For the monsters turn, we provide two constraints for the monster movement:
  - $\forall$t[Time] : $\forall$l[location] : GameState(t) = monster_turn $\Rightarrow$ sum{d[district] n[monsternb]:n=Monster_Move(t,l,d) $\wedge$ Connected(l,d):n} = Monsters_In(t,l).
    The sum of all the monsters moving from a location l to other (connected) districts must be equal to the amount of monsters currently in l. Not less, since all monsters must move and not more, since you cannot move out more monsters than those available.

– $\forall$t[Time] : $\forall$d[district] : GameState(t) = monster_turn $\Rightarrow$ Monsters_In(Next(t),d) = sum{l[location] n[monsternb]:n=Monster_Move(t,l,d) $\wedge$ Connected(l,d):n}
  The sum of all monsters moving from other (connected) locations to a district d will be equal to the amount of monsters in d next turn.

- Two additional constraints have been added to prevent **Move_To** from taking on random values in undefined situations. While the game worked fine without these constraints, they provided a lot more models through permutation with the values of **Move_To**. Read the commentary with the provided constraints in *solution.idp* for more information.

3 Verifications

The commentary provided with both the procedures and the theory of the verifications is enough to clarify the results, though a few things can be mentioned here.

- Verification 1 needed an extra predicate in order to clearly define the constraints. Adding this to the original vocabulary resulted in unwanted extra models, thus it has its own extension to the Arkham vocabulary. (See **LCTVocabulary v1**).
- For verification 3, I found a result which seems to be in conflict with the expected result. We have to check that for the structure **Unlosable** that it is impossible to lose, however it is possible. There are 20 turns in the game, the map is not too big and two gates open very early on. By letting the investigator wander around aimlessly and letting the monsters grow in numbers, it's not very hard to corner the investigator at the later stages of the game. This can be manually verified by running the game on **Unlosable** and following the steps on the graph.
- For verification 4, I use a term **Duration** in order to find the minimization to the amount of lost turns, through which we can easily find how long the investigator survived. Through *minimize* I was able to return a table with the results, in which we can see that the investigator survives for 8 turns (which equals 4 player turns) and loses at turn 8. From the help manual I found out it should have been able to return the value for the amount of lost turns, but I have not succeeded in returning this value. The solution would have been found by subtracting the amount of lost turns from **MAX[:Time]**.