INFOMLSAI Logics for Safe AI
Coursework 3

Coursework released:	22 May 2023, on Blackboard
Coursework due:	23:59 2 June 2023, on Blackboard
Submission format:	a folder containing a pdf and an .ispl file, one per group

Please do the coursework in groups of 2-3 people. Submit a single zipped folder on Blackboard for your group. The folder should contain the pdf file and the ispl file. Please name the folder group-X, where X is the number of your group, and state in the pdf file and comments in the ispl file the names of the members of the group.

Tasks that can be done in Week 5 (w/c 22 May)

The following task can be done after the lectures on Coalition Logic and ATL

W5-1 Represent two agents playing Odds and Evens `https://en.wikipedia.org/wiki/Odds_and_evens_(hand_game)` as a Concurrent Game Structure M_{oe}. In each round of the game, each agent has a choice of two actions: one or two. If the sum of actions is odd (one of the players choses one and another two), player O wins, otherwise (two ones or two twos), player E wins.

Assume that atomic propositions are $\mathsf{win_O}$ and $\mathsf{win_E}$. Look at the encoding of the Prisoner's Dilemma in Slides 5/1 or in the reader. Note that you do not need propositions for utilities, just for wins. Instead of making the game single shot as in the reader (so that after the first move, the game stays in the same state whatever the actions), allow for iterated playing. (1 mark)

W5-2 Express in Coalition Logic: the coalition of agents $\{O, E\}$ can enforce $\mathsf{win_O}$, and it can also enforce $\neg\mathsf{win_O}$. Is this formula true in the initial state q_0 of CGS M_{oe}? Justify your answer with the reference to the truth definition for Coalition Logic. (1 mark)

W5-3 Assume that player O is bored and always alternates between one and two. Change CGS M_{oe} into M_{foe} (for the Fixed strategy of O) to represent this fixed strategy of the Odds player. The Evens player should be unchanged and have a free choice of actions in every state. *Hint: you can encode in the state which action O has just played or is about to play.*

In this structure, player E should have a strategy to always win regardless where the play starts and where player O is in its cycle of one and two. (In a real game situation, player E would need to know this to chose the first correct action. However, so far we are in a perfect information setting, so it is enough that player E *has* a correct action, and we are not worrying about how it knows what this action is.) (1 mark)

W5-4 Express in ATL: player E has a strategy to always win. Note that in the current state win_E does not have to hold yet, so 'always' here should mean from the next state onwards. Define a memoryless strategy for player E that makes this formula true in any state in M_{foe}. (1 mark)

W5-5 In ATL*, it makes a difference whether strategies are memoryless or perfect recall. Perfect recall strategies are unrealistic in many circumstances. Usually, there is a bound on the number of previous states the agent can remember, so the strategies of agents can only take a bounded number of preceding states into account, such as at most 10 or 100000 (but not an arbitrary number).[1]

In both counterexamples to the equivalence of perfect recall and memoryless strategies in ATL* (in the lecture and in the reader), one only needs a bounded strategy to enforce the property (action choice is made knowing the current and the preceding state, which is a history of length at most 2). Give an example of an ATL* property and a CGS, where the bound 2 is not sufficient for a strategy to enforce the property. (1 mark)

Tasks that can be done during Week 6 (w/c 30 May)

The following tasks can be done after the lecture on model checking ATL.

W6-1 Implement M_{foe} in ISPL. Check that the formula from W5-4 is true there and provide a witness strategy (if the formula is the Nth in the list of formulas, look in the formulaN.dot file for the actions).

Hint: when you are encoding a pre-defined CGS in ISPL, it is often the easiest to define a variable `state: {q0, q1, q2}` *(where the values are all the states in the CGS) in the Environment agent, and copy the transition function into the Environment's Evolution function, as in*

```
state = q1 if state = q0 and Agent1.Action = alpha
```

Agents should have state in their Lobsvars because their Protocol depends on states. (1 mark)

Suggestion for testing: if the groups are defined as follows:

```
Groups
        gO = {PlayerO};
```

[1] For a proper definition and discussion of bounded strategies, see [1], although note that this paper applies to imperfect information setting.

```
        gE = {PlayerE};
        g_all = {PlayerO,PlayerE};
end Groups
```

then some formulas that should be true are:

```
        <gE> X winE;
        <gE> F winE;
        !<gO> X winO;
        <g_all> F winO;
```

W6-2 Implement the following scenario in ISPL. A museum consists of three rooms along a corridor: *room0*, *room1*, *room2*. The entrance is *room0*, the exit is *room2*. A valuable painting is in *room2*. A guard patrols the three rooms in the following fixed pattern: *room0, room1, room2, room1, room0, room1, room2*, . . . The guard has an action to move *left* and *right* to the next room; the encoding should make sure that the guard only moves all the way to the right and all the way to the left along the corridor. The thief has the same actions *left, right* but also *wait, steal* and *exit*. The thief can steal the painting if he is alone in *room2* (the guard is in another room). The thief is caught if he is in the same room with the guard after the painting is stolen. The goal of the thief is to have the painting stolen and exit without being caught (to exit, the thief needs to be in *room2*). Express in ATL and check in your encoding that the thief can steal the painting and escape while not being caught. Hint: encode in the `Protocol` which actions are available in which state; the guard's direction of patrolling can be encoded as part of the guard's state. Another hint: the states are quite large, so it is handy to specify evolution of each variable separately. To make sure they update in sync, precede the file with a statement `Semantics=SingleAssignment`. This makes sure that all applicable clauses are applied simultaneously. This requires that only one clause specifying how a variable is updated is enabled in each state.

Suggestion for testing: if `gT` is the thief, and *escaped* becomes true after executing an exit action, then some formulas that should be true are:

```
<gT> G !caught;
<gT> F (stolen and escaped);
!<gT> X stolen;
!<gT> G escaped;
!<gT> X caught;
```

Note that a conjunction of the first two formulas is *not* the correct answer. It is possible to have a strategy to never be caught (e.g. by not stealing) and a different strategy to steal and escape, but the requirement is to have one single strategy to achieve both. Hint: use Until to express that the thief can steal the painting and escape while not being caught.

(2 marks)

W6-3 Modify the guard's patrol behaviour to allow the guard to *wait* in each room, and to perform both the *left* and *right* actions in *room1*, i.e., the guard's patrol behaviour is unrestricted.

- Express in ATL and check in your encoding that the guard can always prevent the picture from being stolen.
- Express in ATL and check that a coalition consisting of the thief and the guard can ensure that the painting is stolen and the thief exits without being caught.

(1 mark)

W6-4 Concurrent game structures are represented in ISPL in a compact form (see reader sections 5.3.1 and 5.3.2), and sometimes are exponentially smaller than the corresponding CGS. Is it always possible, given a CGS, to represent it as an exponentially smaller ISPL encoding? Explain your answer (how to do it or why this is not possible). (1 mark)

References

[1] Steen Vester. Alternating-time temporal logic with finite-memory strategies. In Gabriele Puppis and Tiziano Villa, editors, *Proceedings Fourth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2013*, volume 119 of *EPTCS*, pages 194–207, 2013.