safeAI | checking logical models

ANNELINE DAGGELINCKX, MATTHIJS KEMP, and OTTO MÄTTAS, Utrecht University, The Netherlands

1 WEEK 8 ASSIGNMENTS

1.1 Defining Concurrent Epistemic Game Structures

Consider a CEGS $M_{chicken}$, where $(\mathbb{A}gt, St, Act, d, out)$ is a concurrent game structure, and $\sim_a$ are indistinguishability relations over St, one per agent a in $\mathbb{A}gt$. We can now define the CEGS as a tuple

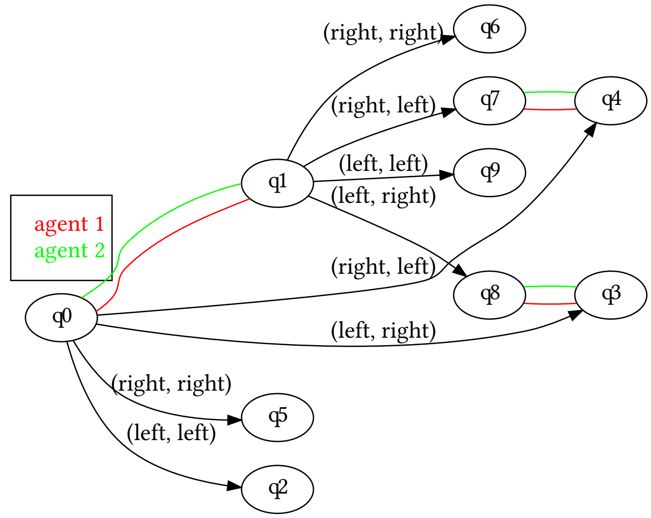
$$M_{chicken} = (\mathbb{A}gt, St, \sim_a |a \in \mathbb{A}gt, Act, d, out) \tag{1}$$

, where

- $\mathbb{A}gt = \{a_1, a_2\}$
- $St =$
  - q_0 = left-hand traffic
  - q_1 = right-hand traffic
  - q_2 = left-hand traffic; a_1 drives left; a_2 drives left
  - q_3 = left-hand traffic; a_1 drives left; a_2 drives right
  - q_4 = left-hand traffic; a_1 drives right; a_2 drives left
  - q_5 = left-hand traffic; a_1 drives right; a_2 drives right
  - q_6 = right-hand traffic; a_1 drives right; a_2 drives right
  - q_7 = right-hand traffic; a_1 drives right; a_2 drives left
  - q_8 = right-hand traffic; a_1 drives left; a_2 drives right
  - q_9 = right-hand traffic; a_1 drives left; a_2 drives left
- $\sim_a = \{q_0, q_1\}, \{q_3, q_8\}, \{q_4, q_7\}, \{q_2\}, \{q_5\}, \{q_6\}, \{q_9\}$
- $Act = \{drive\_left, drive\_right\}$
- $d(\mathbb{A}gt, q_i) = \{drive\_left, drive\_right\}, \forall i \in \{0, ..., 9\}$
- $out =$
  - $out(q_0, drive\_left, drive\_left) = q_2$
  - $out(q_0, drive\_left, drive\_right) = q_3$
  - $out(q_0, drive\_right, drive\_left) = q_4$
  - $out(q_0, drive\_right, drive\_right) = q_5$
  - $out(q_1, drive\_left, drive\_left) = q_9$
  - $out(q_1, drive\_left, drive\_right) = q_8$
  - $out(q_1, drive\_right, drive\_left) = q_7$
  - $out(q_1, drive\_right, drive\_right) = q_6$
- let the evaluations be V, where
  - $V(crash) = \{q_3, q_4, q_5, q_7, q_8, q_9\}$
  - $V(lft) = \{q_0, q_2, q_3, q_4, q_5\}$

As there is no propositional argument matching the action $drive\_right$, we are evaluating q_6 implicitly.

Authors' address: Anneline Daggelinckx, a.daggelinckx@students.uu.nl; Matthijs Kemp, m.g.r.kemp@students.uu.nl; Otto Mättas, o.mattas@students.uu.nl, Utrecht University, P.O. Box 80125, Utrecht, Utrecht, The Netherlands, 3508 TC.



1.2 Validating Concurrent Epistemic Game Structures through Memoryless Strategies

For the defined CEGS $M_{chicken}$ in 1.1, it is **untrue** that under the independent combination of ATL semantics with epistemic semantics (no uniform strategies requirement) agent a_1 has a memoryless strategy in q_0 to enforce $\neg crash$ in the next state ($\langle\langle 1\rangle\rangle X\neg crash$).

This is because in q_0, the only way not to crash is for both agents to take action $drive\_left$. Agent a_1 cannot force this protocol alone, agent a_2 needs to adhere to it as well.

1.3 Validating Concurrent Epistemic Game Structures through Indistinguishability

For the defined CEGS $M_{chicken}$ in 1.1, it is **untrue** that under ATL$_{ir}$, $M_{chicken}, q_0 \models_{ir} \langle\langle 1\rangle\rangle X\neg crash$ holds.

This is because agent a_1 does not know whether it is in q_0 or q_1. Therefore it does not know whether the action to take is $drive\_left$ or $drive\_right$. Even more, if agent a_1 would choose the correct action ($drive\_left$), agent 2 can still cause a crash by executing $drive\_right$.

1.4 Validating Concurrent Epistemic Game Structures through Memoryless Strategies

For the defined CEGS $M_{chicken}$ in 1.1, it is **true** that under the independent combination of ATL semantics with epistemic semantics (no uniform strategies requirement), both agents together have a memoryless strategy in q_0 to enforce $\neg crash$ in the next state.

This is because there is a strategy in q_0 which leads to a state with $\neg crash$ from both agents' perspective. The strategy is as follows $s_i(q_0) = drive\_left \ \ \forall i \in \{1, 2\}$; $s_i(q_1) = drive\_right \ \ \forall i \in \{1, 2\}$.

1.5 Validating Concurrent Epistemic Game Structures through Indistinguishability

For the defined CEGS $M_{chicken}$ in 1.1, it is **untrue** that under ATL$_{ir}$, $M_{chicken}, q_0 \models_{ir} \langle\langle 1,2 \rangle\rangle X \neg crash$ holds.

This is because agents do not know whether they are in q_0 or q_1. Therefore, they do not know whether the action to take is $drive\_left$ or $drive\_right$, not being able to create a uniform strategy.

1.6 Validating Concurrent Epistemic Game Structures through Knowledge

For the defined CEGS $M_{chicken}$ in 1.1, it is **true** that under the independent combination of ATL semantics with epistemic semantics (no uniform strategies requirement), both agents together know that they have a memoryless strategy in q_0 to enforce $\neg crash$ in the next state ($K_1\langle\langle 1,2 \rangle\rangle X \neg crash \wedge K_2\langle\langle 1,2 \rangle\rangle X \neg crash$).

For $q \in \{q_0\}$, it holds that $M_{chicken}, q \models \langle\langle 1,2 \rangle\rangle X \neg crash$, using the following strategy $s_i(q_0) = drive\_left \;\; \forall i \in \{1,2\}$; $s_i(q_1) = drive\_right \;\; \forall i \in \{1,2\}$.

1.7 Validating Concurrent Epistemic Game Structures through Indishtinguishable Knowledge

For the defined CEGS $M_{chicken}$ in 1.1, it is **untrue** that under ATL$_{ir}$, $M_{chicken}, q_0 \models_{ir} K_1\langle\langle 1,2 \rangle\rangle X \neg crash \wedge K_2\langle\langle 1,2 \rangle\rangle X \neg crash$ holds.

This is because there is a state q_1 indistinguishable from q_0 for both agents. For $q \in \{q_0, q_1\}$, there is no uniform strategy to satisfy the requirement ($M_{chicken}, q \models_{ir} \langle\langle 1,2 \rangle\rangle X \neg crash$ for $q \in \{q_0, q_1\}$).

1.8 Formalising ATL$_{ir}$

To say in ATL$_{ir}$ that agent a_1 can ensure that eventually it knows whether it is in a country that drives on the left or a country that drives on the right, we can specify a formula as follows

$$M_{chicken}, q_0 \models_{ir} \langle\langle 1 \rangle\rangle F(K_1 lft \vee K_1 \neg lft) \tag{2}$$

This formula is **untrue** in q_0. This is because agent a_1 can not come to the knowledge on it's own volition as agent a_2 has to also participate in order for agent a_1 to know this. If agent a_2 chooses another action than agent a_1 (so they do (left, right) or (right, left)), they certainly crash, independently of which country they were in. In that case agent 1 does not know whether they are in a left or right driving country.

We can also see that states q_7 and q_4 are indistinguishable, where in the first state of the two, they are in a right driving country, while in the second one they are in a left driving country. The same goes for q_8 and q_3.

1.9 Formalising ATL$_{ir}$

To say in ATL$_{ir}$ that it is inevitable that if in the next state there is no crash, then agent a_1 knows whether it is in a country that drives on the left or a country that drives on the right, we can specify a formula as follows

$$\langle\langle \emptyset \rangle\rangle X(\neg crash \rightarrow (K_1 lft \vee K_1 \neg lft)) \tag{3}$$

This formula is **true** in q_0. This is because there are two states (q_2 and q_6) without a crash which both are distinguishable from each other for both agents. Namely, q_2 can be reached by taking action $drive\_left$ and q_6 can be reached by taking action $drive\_right$.

1.10 Specifying model checking algorithms

To say in ATL$_{ir}$ that there is a strategy which in two steps guarantees enforcing of φ from any state indistinguishable from q, we can specify an algorithm as follows

(1) For each uniform strategy of agent a, generate a model M_i where agent a has only the action assigned by the strategy;
   - There are N different models with $N = |Act|^{|St|}$ ($|Act|$ is the number of actions of agent a and $|St|$ is the number of states.)

(2) For each state M_i, execute algorithm $mcheck_{ATL_{ir}}(M_i, \varphi_0)$. The algorithm has O complexity $O(|St| * |\varphi|)$;

$$\begin{aligned}
&\textbf{function } mcheck_{ATL_{ir}}(M, \varphi_0) \\
&\\
&\textbf{for } \varphi\prime \in Sub(\varphi_0) \textbf{ do} \\
&\\
&\textbf{case } \varphi\prime = p \\
&[\varphi\prime]_M \leftarrow V(p) \\
&\\
&\textbf{case } \varphi\prime = True \\
&[\varphi\prime]_M \leftarrow V(True) = St \\
&\\
&\textbf{case } \varphi\prime = False \\
&[\varphi\prime]_M \leftarrow V(False) = \emptyset \\
&\\
&\textbf{case } \varphi\prime = \langle\langle a \rangle\rangle X^2 \psi \\
&[\varphi\prime]_M \leftarrow pre(a, pre(a, [\psi]_M))
\end{aligned} \tag{4}$$

(3) Check whether $\exists i$ such that for all states $s \in \{q\prime | q\prime \sim_a q\}$: $s \in [\varphi]_{M_i}$. If there exists an i, φ is true in q, otherwise it is false;

(4) The complete algorithm has big O complexity $O(|St| * |Act|^{|St|} * |\varphi|)$.

The algorithm $mcheck_{ATL_{ir}}$ has complexity $O(|St| * |\varphi|)$ and is executed for each generated strategy model M_i. Since there are maximum $|Act|^{St}$ different models, the total complexity is $O(|St| * |\varphi| * |Act|^{St})$.