ESSAI 2024 course: Logic-based specification and verification of multi-agent systems

Lecture 4.2: Specification and Verification of Homogeneous Dynamic Multi-agent Systems

Valentin Goranko
Stockholm University

2nd European Summer School on Artificial Intelligence
ESSAI 2024
Athens, July 15-19, 2024

Based on:

Riccardo De Masellis and Valentin Goranko: **Logic-based Specification and Verification of Homogeneous Dynamic Multi-agent Systems**, J. of Autonomous Agents and Multiagent Systems, vol. 34 (2), 2020.

Introduction: Homogeneous Dynamic Multi-agent Systems (HDMAS)

We consider multi-agent systems which evolve by means of discrete transitions from state to state, and have two special features:

Homogeneous:

- all agents have the same type (protocol). In particular:
  – all agents have the same available actions at each system state;
  – the effect of any action does not depend on which agent performs it, but only on how many agents perform it.
- thus, transitions are completely determined by how many agents perform each possible action.

Dynamic:

- unbounded (but always finite) number of agents in the system.
- at every round, each agent may be 'active', by taking a 'real' action, or 'inactive' by performing action 'idle'.
  So, agents can 'join' and 'leave' the system dynamically.

Controllable and uncontrollable agents in HDMAS

All agents in a HDMAS are split into

– controllable (by the leader/supervisor), and

– uncontrollable (regarded as environment or adversaries).

All controllable and uncontrollable agents have the same type.

However, the controllable agents follow a prescribed strategy, whereas the uncontrollable ones are unconstrained.

Homogeneous Dynamic Multi-agent Systems: some examples

- sensor/computer/social networks;
- election systems and voting procedures;
- more abstractly, systems of *strategic dynamic resource allocation*, such as Colonel Blotto games.

HDMAS: a typical property to specify and verify

Assuming perfect coordination between the controllable agents, an HDMAS can be regarded as a concurrent game between two 'super-players': Proponent and Opponent.

Proponent has a (temporalised) qualitative objective, for instance:

– to eventually reach a desired goal state in the system, or
– to keep the system in a safe region, until a goal state is reached.

Opponent tries to prevent the achievement of that objective.

A typical property to specify and verify in an HDMAS:

Proponent has a joint strategy for a coalition of M controllable agents acting against (Opponent represented by) at most N non-controllable agents, to ensure that Proponent's objective is satisfied on every play enabled by that strategy.

HDMAS: technical preliminaries

- $Ag = \{ag_1, ag_2, \ldots\}$: unbounded universe of **agents**.
- $Act = \{act_1, \ldots, act_n\}$: a finite set of **(names for) actions**.
  $Act^+ = Act \cup \{\varepsilon\}$, where ε is a specific **'idle' action**.
- $X = \{x_1, \ldots, x_n\}$ and $X^+ = X \cup \{x_\varepsilon\}$: a fixed set of variables, called **action counters**, respectively associated with the actions in Act^+.
- **action profile**: a tuple of actions in Act^+, one for each agent in Ag.
- An **action distribution** for an action profile σ is a function
  $\mathbf{act}_\sigma : X \to \mathbb{N}$, where, for $i = 1, ..., n$,
  $\mathbf{act}_\sigma(x_i)$ is the number of agents taking action act_i in σ.
  (The idle action ε is not counted.)
- A **(transition) guard** is an open formula of Presburger arithmetic (PrA) over the variables in X.
- **Satisfaction of a transition guard g** by an action distribution **act**, denoted **act** $\models g$, is defined by the standard semantics of PrA.

HDMAS models: technical definition

An **HDMAS structure** for a set of agents Ag and a set of actions Act^+:

$$\mathcal{S} = \langle Ag, Act^+, S, \delta \rangle$$

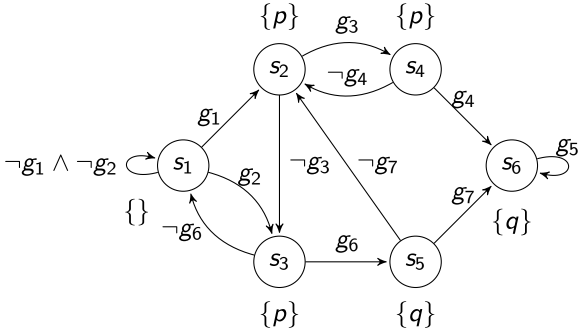
- S is a (usually finite) set of **states**.
- $\delta : S \times S \to G$ is a **transition guards function**.

  δ labels all possible transitions between states
  with guards that determine, for every possible action distribution,
  a unique transition to a successor state.

NB: the transition caused by any action profile only depends on its action distribution.
Hereafter, we use action distributions instead of action profiles.

HDMAS-based model: HDMAS structure +
labelling λ of states with sets of atomic propositions in a fixed set Φ.

An example of HDMAS



The model involves 6 states and 2 atomic propositions (p and q). Besides ε, there are 3 'real' actions, with respective counters x_1, x_2, x_3. The guards:

$g_1 := (x_1 \geq 2x_2) \wedge (x_3 \leq 3)$

$g_2 := (x_1 + x_2 + x_3 \leq 10) \wedge (x_3 > 3)$

$g_3 := (x_1 > 5) \wedge (x_3 > x_1)$

$g_4 := x_1 > 5 \wedge (3x_2 < x_1 + 2x_3)$

$g_5 := x_1 = x_1$

$g_6 := x_1 + 2x_2 \geq x_3$

$g_7 := x_2 = x_3$

The logic $\mathcal{L}_{\mathsf{HDMAS}}$

$\mathcal{L}_{\mathsf{HDMAS}}$ is a logic (extending ATL) for specifying strategic properties in HDMAS, such as existence of a strategy for the controllable agents that guarantees satisfaction of Proponent's objective against any behavour of the uncontrollable agents.

The logic $\mathcal{L}_{\mathsf{HDMAS}}$: terms

The vocabulary of $\mathcal{L}_{\mathsf{HDMAS}}$ contains:

- a fixed set of atomic propositions $\Phi = \{p_1, p_2, ...\}$,
- a set of two special variables $Y = \{y_1, y_2\}$ ranging over $\mathbb{N}$, called **agent counters**, representing respectively the current numbers of controllable and uncontrollable agents,
- a set of auxiliary **agent-counting parameters** $Z = \{z_1, z_2, \ldots\}$.

The set of **terms**: $T = \mathbb{N} \cup Y \cup Z$.

(Natural numbers will be identified with their numerals.)

The logic $\mathcal{L}_{\mathsf{HDMAS}}$: formulae

Two sorts of formulae:

Path formulae: $\chi ::= \mathsf{X}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi$,
where φ are state formulae.

State formulae:
$\varphi ::= \top \mid p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \forall y\varphi \mid \exists y\varphi \mid \langle\!\langle t_1, t_2 \rangle\!\rangle\, \chi$
where $p \in \Phi$, $y \in Y$, $t_1 \in T \setminus \{y_2\}$, $t_2 \in T \setminus \{y_1\}$, and χ is a path formula.

In the formula $\langle\!\langle t_1, t_2 \rangle\!\rangle\, \chi$, t_1 denotes the number of controllable agents, and t_2 – the number of uncontrollable agents.

Positive polarity constraint for the formulae $\forall y\varphi$ and $\exists y\varphi$:
all free occurrences of y in φ must have a positive polarity
(i.e., must be in the scope of an even number of negations).

Formulae of $\mathcal{L}_{\mathsf{HDMAS}}$: some examples

- $\langle\langle 7, 5 \rangle\rangle\, \mathsf{X}\, p$:
  "7 controllable agents have a joint action ensuring against 5 uncontrollable agents that any outcome state satisfies p".
- $\forall y_2 \langle\langle 7, y_2 \rangle\rangle\, \mathsf{G}\, \neg p$:
  "for any number y_2, 7 controllable agents have a joint strategy to ensure against y_2 uncontrollable agents that any outcome play never reaches a p-state."
- $\forall y_2 \exists y_1 \langle\langle y_1, y_2 \rangle\rangle\, q\, \mathsf{U}\, \neg p$:
  "For any number (y_2) of uncontrollable agents there is a number (y_1) of controllable agents who have a joint strategy to ensure that any outcome play will stay within a q-region until it eventually reaches a non-p-state."

Semantics of $\mathcal{L}_{\mathsf{HDMAS}}$

Assignment: a function $\theta : T \to \mathbb{N}$, where $\theta(i) = i$ for $i \in \mathbb{N}$.

Let $\mathcal{M}$ be a HDMAS, s be a state and θ an assignment in it.
The **satisfaction relation** $\models$ is inductively defined as follows:

- $\mathcal{M}, s, \theta \models p$ iff $p \in \lambda(s)$,
- The semantics of $\top$, $\wedge$, $\vee$, $\neg$, $\forall$, $\exists$: as in classical logic.
- $\mathcal{M}, s, \theta \models \langle\!\langle t_1, t_2 \rangle\!\rangle \, \chi$ *iff*
  there exists a joint strategy σ for a coalition of $\theta(t_1)$ controllable agents such that every play enabled by σ, against (at most) $\theta(t_2)$ uncontrollable agents, satisfies the temporal objective χ.

Monotonicity properties and quantifier elimination equivalences

$\langle\langle t_1, t_2 \rangle\rangle\, \chi$ is monotone with respect to t_1
and anti-monotone with respect to t_2.

That means:

- If $\mathcal{M}, s, \theta \models \langle\langle t_1, t_2 \rangle\rangle\, \chi$ and $C > t_1$ then $\mathcal{M}, s, \theta \models \langle\langle C, t_2 \rangle\rangle\, \chi$;
- If $\mathcal{M}, s, \theta \models \langle\langle t_1, t_2 \rangle\rangle\, \chi$ and $N < t_2$ then $\mathcal{M}, s, \theta \models \langle\langle t_2, N \rangle\rangle\, \chi$.

Based on these, some quantifier elimination equivalences are valid, e.g.:

- $\forall y_1 \langle\langle y_1, t \rangle\rangle\, \chi \equiv \langle\langle 0, t \rangle\rangle\, \chi[0/y_1]$,
- $\exists y_2 \langle\langle t, y_2 \rangle\rangle\, \chi \equiv \langle\langle t, 0 \rangle\rangle\, \chi[0/y_2]$,
- $\forall y_2 \forall y_1 \langle\langle y_1, y_2 \rangle\rangle\, \chi \equiv \forall y_2 \langle\langle 0, y_2 \rangle\rangle\, \chi[0/y_1]$,
- $\exists y_1 \exists y_2 \langle\langle y_1, y_2 \rangle\rangle\, \chi \equiv \exists y_1 \langle\langle y_1, 0 \rangle\rangle\, \chi[0/y_2]$, etc.

Normal form of $\mathcal{L}_{\mathsf{HDMAS}}$-formulae

The full language $\mathcal{L}_{\mathsf{HDMAS}}$ is not suitable for algorithmic model checking, because of the unconstrained nesting of strategic operators and quantification over agent counters.

For model checking we transform $\mathcal{L}_{\mathsf{HDMAS}}$-formulae to **normal form**, by imposing syntactic restrictions on the patterns of quantification.

Informally, the formulae in $\mathcal{L}^{\mathsf{NF}}_{\mathsf{HDMAS}}$ are defined by modifying the recursive definition of state formulae of $\mathcal{L}_{\mathsf{HDMAS}}$, where the clauses $\forall y\varphi$ and $\exists y\varphi$ are replaced with the following, where χ is a temporal objective:

$$\exists y_1 \langle\langle y_1, t_2\rangle\rangle\, \chi \mid \forall y_2 \exists y_1 \langle\langle y_1, y_2\rangle\rangle\, \chi \mid \forall y_2 \langle\langle t_1, y_2\rangle\rangle\, \chi \mid \exists y_1 \forall y_2 \langle\langle y_1, y_2\rangle\rangle\, \chi$$

$\mathcal{L}^{\mathsf{NF}}_{\mathsf{HDMAS}}$: the fragment of $\mathcal{L}_{\mathsf{HDMAS}}$-formulae in normal form.

Transformation to normal forms

Normal forms restrict the language syntactically, but not its expressiveness.

A key technical result: a recursive procedure $\textsc{nf}$, converting every $\mathcal{L}_{\mathsf{HDMAS}}$-formula φ into $\textsc{nf}(\varphi) \in \mathcal{L}^{\mathsf{NF}}_{\mathsf{HDMAS}}$, such that:

1. $\textsc{nf}(\varphi) \equiv_{\mathsf{fin}} \varphi$. ($\equiv_{\mathsf{fin}}$ is equivalence on all finite HDMAS models)
2. If $\varphi \in \mathcal{L}^{\mathsf{NF}}_{\mathsf{HDMAS}}$ then $\textsc{nf}(\varphi) = \varphi$.
3. $\textsc{nf}(\varphi)$ can be computed effectively and has length linearly bounded above by the length of φ.

Thus, normal forms restrict the language syntactically,
but do not reduce its expressiveness over finite models.

Fixpoint equivalences for formulae in normal form

The strategic operators for formulae in $\mathcal{L}^{\mathsf{NF}}_{\mathsf{HDMAS}}$ satisfy fixpoint equivalences over finite models, listed in the theorem below.

Theorem. For every terms t, t', t'' the following equivalences hold, where the formulae on the left are in $\mathcal{L}^{\mathsf{NF}}_{\mathsf{HDMAS}}$.

- $\langle\!\langle t', t'' \rangle\!\rangle\, \mathsf{G}\, \varphi \equiv \varphi \wedge \langle\!\langle t', t'' \rangle\!\rangle\, \mathsf{X}\, \langle\!\langle t', t'' \rangle\!\rangle\, \mathsf{G}\, \varphi$
- $\langle\!\langle t', t'' \rangle\!\rangle\, \psi\, \mathsf{U}\, \varphi \equiv \varphi \vee (\psi \wedge \langle\!\langle t', t'' \rangle\!\rangle\, \mathsf{X}\, \langle\!\langle t', t'' \rangle\!\rangle\, \psi\, \mathsf{U}\, \varphi)$
- $\exists y_1 \langle\!\langle y_1, t \rangle\!\rangle\, \mathsf{G}\, \varphi \equiv_{\mathsf{fin}} \varphi \wedge \exists y_1 \langle\!\langle y_1, t \rangle\!\rangle\, \mathsf{X}\, \exists y_1 \langle\!\langle y_1, t \rangle\!\rangle\, \mathsf{G}\, \varphi$
- $\forall y_2 \langle\!\langle t, y_2 \rangle\!\rangle\, \mathsf{G}\, \varphi \equiv_{\mathsf{fin}} \varphi \wedge \forall y_2 \langle\!\langle t, y_2 \rangle\!\rangle\, \mathsf{X}\, \forall y_2 \langle\!\langle t, y_2 \rangle\!\rangle\, \mathsf{G}\, \varphi$
- $\exists y_1 \langle\!\langle y_1, t \rangle\!\rangle\, \psi\, \mathsf{U}\, \varphi \equiv_{\mathsf{fin}} \varphi \vee (\psi \wedge \exists y_1 \langle\!\langle y_1, t \rangle\!\rangle\, \mathsf{X}\, \exists y_1 \langle\!\langle y_1, t \rangle\!\rangle\, \psi\, \mathsf{U}\, \varphi)$
- $\forall y_2 \langle\!\langle t, y_2 \rangle\!\rangle\, \psi\, \mathsf{U}\, \varphi \equiv_{\mathsf{fin}} \varphi \vee (\psi \wedge \forall y_2 \langle\!\langle t, y_2 \rangle\!\rangle\, \mathsf{X}\, \forall y_2 \langle\!\langle t, y_2 \rangle\!\rangle\, \psi\, \mathsf{U}\, \varphi)$
- $\forall y_2 \exists y_1 \langle\!\langle y_1, y_2 \rangle\!\rangle\, \mathsf{G}\, \varphi \equiv_{\mathsf{fin}} \varphi \wedge \forall y_2 \exists y_1 \langle\!\langle y_1, y_2 \rangle\!\rangle\, \mathsf{X}\, \forall y_2 \exists y_1 \langle\!\langle y_1, y_2 \rangle\!\rangle\, \mathsf{G}\, \varphi.$
- $\exists y_1 \forall y_2 \langle\!\langle y_1, y_2 \rangle\!\rangle\, \mathsf{G}\, \varphi \equiv_{\mathsf{fin}} \varphi \wedge \exists y_1 \forall y_2 \langle\!\langle y_1, y_2 \rangle\!\rangle\, \mathsf{X}\, \exists y_1 \forall y_2 \langle\!\langle y_1, y_2 \rangle\!\rangle\, \mathsf{G}\, \varphi.$
- $\forall y_2 \exists y_1 \langle\!\langle y_1, y_2 \rangle\!\rangle\, \psi\, \mathsf{U}\, \varphi \equiv_{\mathsf{fin}} \varphi \vee (\psi \wedge \forall y_2 \exists y_1 \langle\!\langle y_1, y_2 \rangle\!\rangle\, \mathsf{X}\, \forall y_2 \exists y_1 \langle\!\langle y_1, y_2 \rangle\!\rangle\, \psi\, \mathsf{U}\, \varphi).$
- $\exists y_1 \forall y_2 \langle\!\langle y_1, y_2 \rangle\!\rangle\, \psi\, \mathsf{U}\, \varphi \equiv_{\mathsf{fin}} \varphi \vee (\psi \wedge \exists y_1 \forall y_2 \langle\!\langle y_1, y_2 \rangle\!\rangle\, \mathsf{X}\, \exists y_1 \forall y_2 \langle\!\langle y_1, y_2 \rangle\!\rangle\, \psi\, \mathsf{U}\, \varphi).$

Model checking of formulae in $\mathcal{L}^{\mathsf{NF}}_{\mathsf{HDMAS}}$

Given a state formula φ of $\mathcal{L}_{\mathsf{HDMAS}}$, a HDMAS model $\mathcal{M}$, a state $s \in \mathcal{M}$, and an assignment θ in $\mathcal{M}$:

- the **local model checking problem for HDMAS** is the problem of deciding whether $\mathcal{M}, s, \theta \models \varphi$,
- the **global model checking problem** is the problem of computing the **state extension of φ in $\mathcal{M}$ for θ**, formally defined as:

$$[\![\varphi]\!]^{\theta}_{\mathcal{M}} = \{s \in S \mid \mathcal{M}, s, \theta \models \varphi\}.$$

Recall: the transitions in HDMAS models are represented symbolically, in terms of the guards.
So, an explicit representation of the transition graph is generally infinite.

That is why, the model checking of HDMAS models is done symbolically, by reduction to Presburger arithmetic (PrA).

Algorithm for global model checking in $\mathcal{L}^{\mathrm{NF}}_{\mathrm{HDMAS}}$: the core sub-procedure PREIMG

For a set of states $Q \subseteq S$ and integers $C, N \in \mathbb{N}$, the **(C, N)-controllable pre-image of Q** is the set of states from which C controllable agents have a joint action, which, when played against *any* joint action of N uncontrollable agents produces an outcome state in Q.

The procedure PREIMG returns the (C, N)-controllable pre-image of Q.

PREIMG is extended to compute, for any terms t_1, t_2,
the **(t_1, t_2)-controllable pre-image of Q**
by means of a PrA-formula with t_1, t_2 as parameters.

Algorithm for global model checking in $\mathcal{L}^{\mathsf{NF}}_{\mathsf{HDMAS}}$: informal description

1. For $\varphi = \langle\langle t_1, t_2\rangle\rangle \mathsf{X}\, \psi$, $\textsc{preImg}$ applied to $Q = [\![\psi]\!]^{\theta}_{\mathcal{M}}$, computes the state extension of φ, as a PrA-formula parameterised with t_1, t_2.

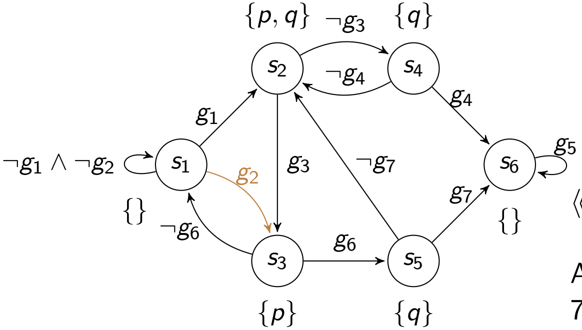
2. Then the procedure is readily extended to all quantified extensions of $\langle\langle t_1, t_2\rangle\rangle \mathsf{X}\, \psi$, by adding the respective quantification to the result.

3. Lastly, for the long-term temporal objectives model checking is done by fixpoint unfolding iterations (like in model checking of CTL or ATL).
Every iteration stage produces again a PrA-formula.
Stabilisation and reaching the fixpoint is detected by checking equivalence of the PrA-formulae produced at the successive iterations.

So, in all cases, computing the state extension of a $\mathcal{L}^{\mathsf{NF}}_{\mathsf{HDMAS}}$-formula is reduced to computing a PrA-formula.

Thus, the local model checking problem for $\mathcal{L}^{\mathsf{NF}}_{\mathsf{HDMAS}}$ is reduced to checking the truth of PrA-formulae.

Local model checking in $\mathcal{L}^{\mathrm{NF}}_{\mathrm{HDMAS}}$: example 1



$\langle\langle 7, 5 \rangle\rangle \, \mathrm{X} \, p$ is true in s_1.

An action profile for the 7 controllable agents :

1 agent performs act_1,

4 agents perform act_3,

2 agents perform ϵ

$g_1 := (x_1 \geq 2x_2) \wedge (x_3 \leq 3)$

$g_2 := (x_1 + x_2 + x_3 \leq 10) \wedge (x_3 > 3)$

$g_3 := (x_1 > 5) \wedge (x_3 > x_1)$

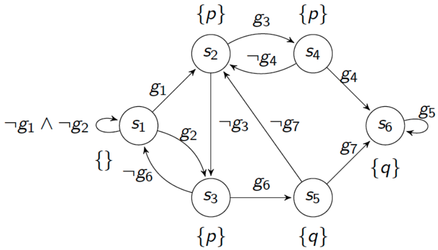
$g_4 := x_1 > 5 \wedge (3x_2 - 2x_3 < x_1)$

$g_5 := \ldots$

$\ldots$

Global model checking problem for $\mathcal{L}^{\mathsf{NF}}_{\mathsf{HDMAS}}$: example 2

Computing $[\![\varphi]\!]_{\mathcal{M}}$ for $\varphi = \exists y_1 \forall y_2 \langle\!\langle y_1, y_2 \rangle\!\rangle \mathrm{X}\,(p \vee q)$ in the given model $\mathcal{M}$.



$$g_1 := (x_1 \geq 2x_2) \wedge (x_3 \leq 3)$$
$$g_2 := (x_1 + x_2 + x_3 \leq 10) \wedge (x_3 > 3)$$
$$g_3 := (x_1 > 5) \wedge (x_3 > x_1)$$
$$g_4 := x_1 > 5 \wedge (3x_2 < x_1 + 2x_3)$$
$$g_5 := x_1 = x_1$$
$$g_6 := x_1 + 2x_2 \geq x_3$$
$$g_7 := x_2 = x_3$$

1. Compute $[\![p \vee q]\!]_{\mathcal{M}} = \{s_2, s_3, s_4, s_5, s_6\}$.

2. For each $s \in \mathcal{M}$ check the truth of $\exists y_1 \forall y_2 \mathrm{PrF}(\mathcal{M}, s, y_1, y_2, [\![p \vee q]\!]_{\mathcal{M}})$.

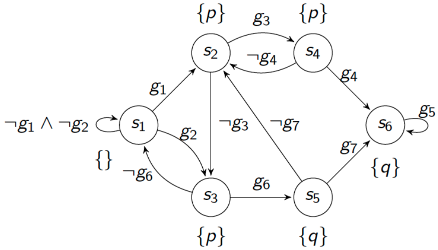
– 11 uncontrollable agents can keep the system in s_1 by all performing act_3; so, $\exists y_1 \forall y_2\, \mathrm{PrF}(\mathcal{M}, s_1, y_1, y_2, [\![p \vee q]\!]_{\mathcal{M}})$ is false, hence s_1 is not in the $\exists y_1 \forall y_2 (y_1, y_2)$- controllable pre-image of $[\![p \vee q]\!]_{\mathcal{M}}$.

– All outgoing transitions from s_2 lead to states in $[\![p \vee q]\!]_{\mathcal{M}}$; hence $\exists y_1 \forall y_2\, \mathrm{PrF}(\mathcal{M}, s_2, y_1, y_2, [\![p \vee q]\!]_{\mathcal{M}})$ is true, so s_2 is in the $\exists y_1 \forall y_2 (y_1, y_2)$-controllable pre-image of $[\![p \vee q]\!]_{\mathcal{M}}$.

– Checking all other states likewise produces the final result: $[\![\varphi]\!]_{\mathcal{M}} = \{s_2, s_4, s_5, s_6\}$.

Global model checking problem for $\mathcal{L}_{\mathrm{HDMAS}}^{\mathrm{NF}}$: example 3

Computing $[\![\psi]\!]_{\mathcal{M}}$ for $\psi = \langle\!\langle 7,4 \rangle\!\rangle \mathsf{X} (\forall y_2 \exists y_1 \langle\!\langle y_1, y_2 \rangle\!\rangle \mathsf{G}\, p)$ in the model $\mathcal{M}$.



$g_1 := (x_1 \geq 2x_2) \wedge (x_3 \leq 3)$

$g_2 := (x_1 + x_2 + x_3 \leq 10) \wedge (x_3 > 3)$

$g_3 := (x_1 > 5) \wedge (x_3 > x_1)$

$g_4 := x_1 > 5 \wedge (3x_2 < x_1 + 2x_3)$

$g_5 := x_1 = x_1$

$g_6 := x_1 + 2x_2 \geq x_3$

$g_7 := x_2 = x_3$

1. Initialize $Z \leftarrow \{s_2, s_3, s_4\}$ and $W \leftarrow S = \{s_1, \ldots, s_6\}$.

A while-loop computing the fixpoint:

– $W \leftarrow \{s_2, s_3, s_4\}$;

– $\text{PREIMG}(\mathcal{M}, y_1, y_2, \{s_2, s_3, s_4\}, \theta, \forall y_2 \exists y_1) = \{s_2, s_4, s_5\}$;

– $Z \leftarrow \{s_2, s_4, s_5\} \cap \{s_2, s_3, s_4\} = \{s_2, s_4\}$.

Next round: $Z \leftarrow \ldots\{s_2, s_4\}$.

Now the fixpoint is reached.

So, $[\![\forall y_2 \exists y_1 \langle\!\langle y_1, y_2 \rangle\!\rangle \mathsf{G}\, p]\!]_{\mathcal{M}} = \{s_2, s_4\}$.

Lastly, for computing the outer Next-formula the algo calls the PREIMG procedure.

For each $s \in S$ the truth of formula $\mathrm{PrF}(\mathcal{M}, s, 7, 4, \{s_2, s_4\})$ is called.

The final result is $[\![\psi]\!]_{\mathcal{M}} = \{s_4, s_5\}$.

Complexity estimates

Complexity of model checking of $\mathcal{L}^{\mathsf{NF}}_{\mathsf{HDMAS}}$- formulae: by using results (by Hasse and others) on complexity of model checking of PrA-formulas.

- Ranges from Σ_3^{EXP} in the general case, to **NP**-complete when the number of controllable or uncontrollable agents is fixed or bounded.
- When the number of actions is bounded, too, it is **P**-complete.

Lecture 4.2: Closing remarks

Future works (in some possible futures) include:

- Allowing several types of agents.
- Allowing several coalitions of controlled agents.
- Extending the language, e.g. by relaxing some syntactic restrictions.
- Possible applications include:

  – solving games of the type of generalised Colonel Blotto games by model checking $\mathcal{L}^{\mathsf{NF}}_{\mathsf{HDMAS}}$-formulae.

  – design and verification of sensor networks and voting procedures.

  – etc.

END OF LECTURE 4.2