Counter Abstraction for Open Multi Agent Systems

Alphy George, Shabana AT, S Sheerazuddin

NIT Calicut

Workshop on Research Highlights in Programming Languages

December 17, 2024

- **Multi-Agent System (MAS)**: Group of agents that interact with one another and their environment

- **Multi-Agent System (MAS)**: Group of agents that interact with one another and their environment
- Two types:
  - Swarms (Homogeneous)
  - Teams (Heterogeneous)

- **Multi-Agent System (MAS)**: Group of agents that interact with one another and their environment
- Two types:
  - Swarms (Homogeneous)
  - Teams (Heterogeneous)
- **Open MAS (OMAS)**: Countably many agents may leave or join the system at run-time

A semantics for reasoning about OMAS that is based on interpreted systems, the standard framework for modelling multi-agent systems .

An OIS consists of,

- **Agents**: Capture the behaviours of the individuals that are joining and leaving the system
- **Environment**: Capturing the rest of the state of the system

Agent Definition

Definition

Agent is defined by a tuple $A = \langle L, \iota, Act, P, t \rangle$

- L is set of local states
- $\iota \in L$ is a unique initial state
- Act is non empty set of actions
- $P : L \rightarrow \mathcal{P}(Act)$ is a protocol that selects which actions may be performed at a given state
- $t : L \times Act \times \mathcal{P}(Act) \times Act_E \rightarrow L$ is the transition function

Scenario:

- A number of trains wish to enter a tunnel
- A controller coordinates train entries to avoid collisions and maintain safe passage through the tunnel

Agent: Train, **Environment:** Controller

Scenario:

- A number of trains wish to enter a tunnel
- A controller coordinates train entries to avoid collisions and maintain safe passage through the tunnel

Agent: Train, **Environment:** Controller

Agent template

- Agent states:
  $L \triangleq \{outside, entering, tunnel, collision\}$
- Initial state: $outside$
- Actions: $Act \triangleq \{approach, enter, wait, exit\}$
- Protocol $P : L \to \mathcal{P}(Act)$ is
  - $P(entering) = \{enter\}$
  - $P(tunnel) = \{wait, exit\}$
  - $P(outside) = \{wait, approach\}$

Definition

Environment is defined by a tuple $E = \langle L_E, \iota_E, Act_E, P_E, t_E \rangle$

- L_E is a non-empty set of local states
- $\iota_E \in L$ is a unique initial state
- Act_E is a no-empty set of actions
- $P_E : L_E \rightarrow \mathcal{P}(Act_E)$ is a protocol that defines which actions are enabled at each local state
- $t_E : L_E \times Act_E \times \mathcal{P}(Act) \rightarrow L_E$

Environment template

- Environment states $L \triangleq \{green, red, alert\}$
- Initial state: $green$
- Actions: $Act_E \triangleq \{go, stop, alert\}$
- Protocol $P : L_E \rightarrow \mathcal{P}(Act_E)$ is
  - $P(green) = \{go\}$
  - $P(red) = \{stop, alert\}$

Definition

OIS is a tuple $\mathcal{O} = \langle A, E, V \rangle$ where $V : L \rightarrow \mathcal{P}(AP)$ is a labelling function for the agents local states.

Open Interpreted System (OIS)

Definition

OIS is a tuple $\mathcal{O} = \langle A, E, V \rangle$ where $V : L \rightarrow \mathcal{P}(AP)$ is a labelling function for the agents local states.

For train gate controller

$AP = \{safe\}$

$V(entering) = V(outside) = V(tunnel) = V(exited) = safe$

$V(collision) = \{\}$

Definition

OIS is a tuple $\mathcal{O} = \langle A, E, V \rangle$ where $V : L \rightarrow \mathcal{P}(AP)$ is a labelling function for the agents local states.

For train gate controller

$AP = \{safe\}$

$V(entering) = V(outside) = V(tunnel) = V(exited) = safe$

$V(collision) = \{\}$

Behaviour of an OIS over time

- A global state G_n is $(L \times \mathbb{Z}^+) \times \cdots \times (L \times \mathbb{Z}^+) \times L_E$
- Initial global state $(g_0) = \langle \iota_E \rangle$
- $G \triangleq \bigcup_{n \in \mathbb{N}} G_n$ for the set of all global states of any size

Transitions in OMAS

Agent Joining Transition:

- New agent dynamically joins the system to initial state
- Global state expands to include the new agent
- A unique, freshly generated identity is assigned

g = <outside, 1> <green>

↓ Agent joining

g' = <outside, 1> <outside, 2> <green>

Transitions in OMAS

Agent Joining Transition:

- New agent dynamically joins the system to initial state
- Global state expands to include the new agent
- A unique, freshly generated identity is assigned

g = <outside, 1> <green>

↓ Agent joining

g' = <outside, 1> <outside, 2> <green>

Agent Leaving Transition:

- An existing agent exits the system
- Global state is updated by removing the departing agent's information

g = <outside, 1> <tunnel, 2> <red>

↓ Agent leaving

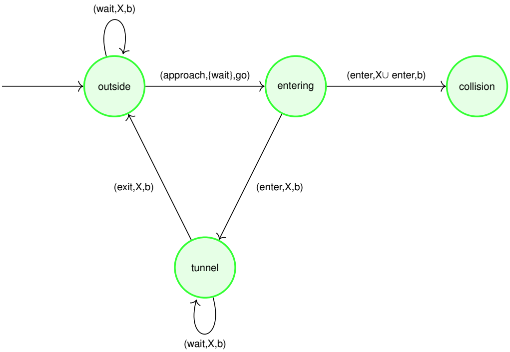
g' = <outside, 1> <red>

- **Joint Action Transition:**
  - Represents a system evolution where all agents and the environment perform actions simultaneously
  - Global state evolves based on the transition functions of agents and the environment, maintaining agent identities

Transition Function: Agent

- $t : L \times Act \times \mathcal{P}(Act) \times Act_E \rightarrow L$ is given by:

$$(l, a, X, a_E) \longmapsto \begin{cases} tunnel & \text{if } a = enter \\ outside & \text{if } a = exit \\ entering & \text{if } a = approach, a_E = go \text{ and } X = \{wait\} \\ collision & \text{if } a = enter \text{ and } enter \in X \\ l & \text{otherwise} \end{cases}$$



- $t_E : L_E \times Act_E \times \mathcal{P}(Act) \rightarrow L_E$ is given by

$$(l_E, a_E, X) \longmapsto \begin{cases} red & \text{if } enter \in X \\ green & \text{if } exit \in X \\ alert & \text{if } enter \in X \\ l_E & \text{otherwise} \end{cases}$$

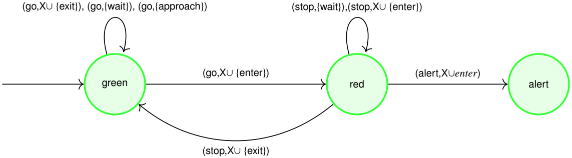


Figure: Environment

Definition (OMCP)

Given an OIS $\mathcal{O}$ and an $IACTL \setminus X$ formula φ, determine whether $\mathcal{O} \models \varphi$.

- OMCP is undecidable[a]
- Identified a subclass of OMAS where safety checking is decidable

Syntax of $IACTL \setminus X$:

$$\phi ::= \forall v : \phi \mid \psi$$

$$\psi ::= (p, v) \mid \neg(p, v) \mid \psi \wedge \psi \mid \psi \vee \psi \mid A(\psi U \psi) \mid A(\psi R \psi)$$

[a] **kouvaros2019formal**.

- **Regular OMAS**: a subclass of OMAS
- Agents can leave the system from a fixed **leaving state**
- Modified agent template: $A = \langle L, \iota, Act, P, t, \mathbf{leave} \rangle$
- $P(leave) = \emptyset$

- **Regular OMAS**: a subclass of OMAS
- Agents can leave the system from a fixed **leaving state**
- Modified agent template: $A = \langle L, \iota, Act, P, t, \mathbf{leave} \rangle$
- $P(leave) = \emptyset$

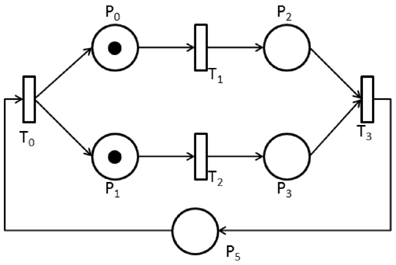
Modified agent template for Train gate controller

- $L \triangleq \{entering, tunnel, outside, collision, exited\}$
- leave state: $exited$

- Safety checking decidable through encoding Regular OMAS into Petri net
- Use the properties of Petri Nets to reason about safety
- Tools are available for Petri nets

A Petri net N is defined formally as a 5-tuple $N = (P, T, F, W, M_0)$ where

- P is a finite set of **places**
- T is a finite set of **transitions**
- F is a **flow relation** over places and transitions, $F \subseteq (P \times T) \cup (T \times P)$,
- W is the **weight function**, $W : F \longrightarrow \mathbb{N}$ and
- $M_0 : P \longrightarrow \mathbb{N}$ is the **initial marking**



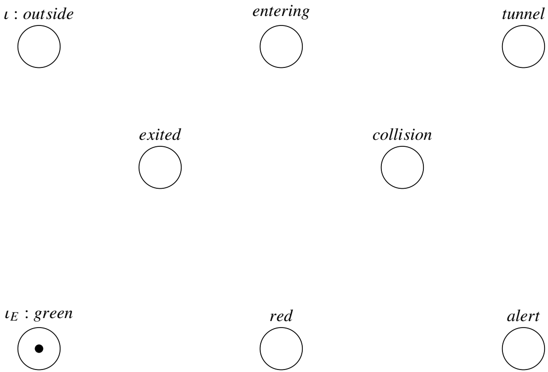
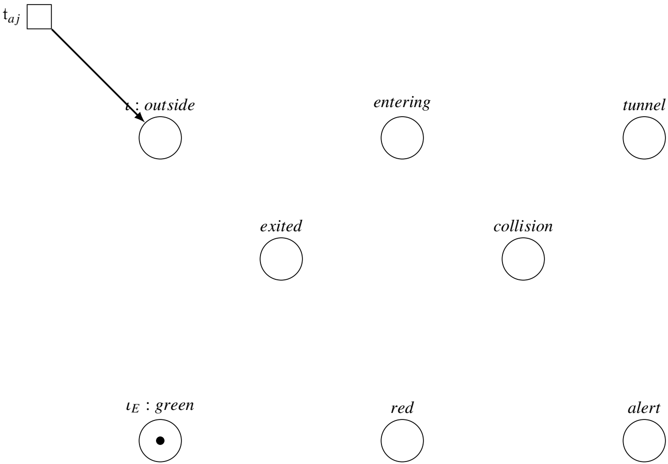
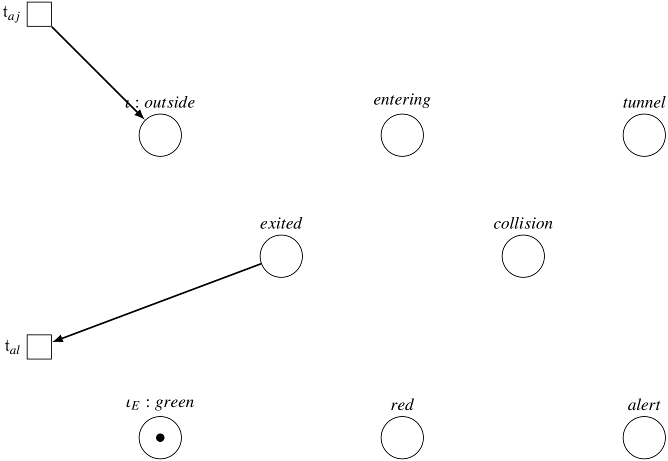
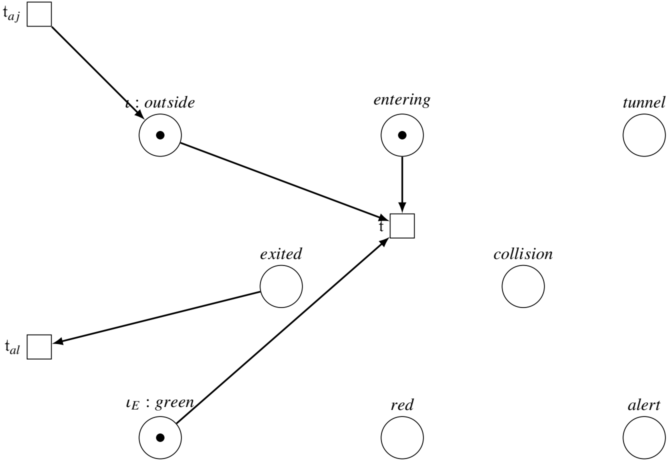


Figure: Places in Petri net encoding

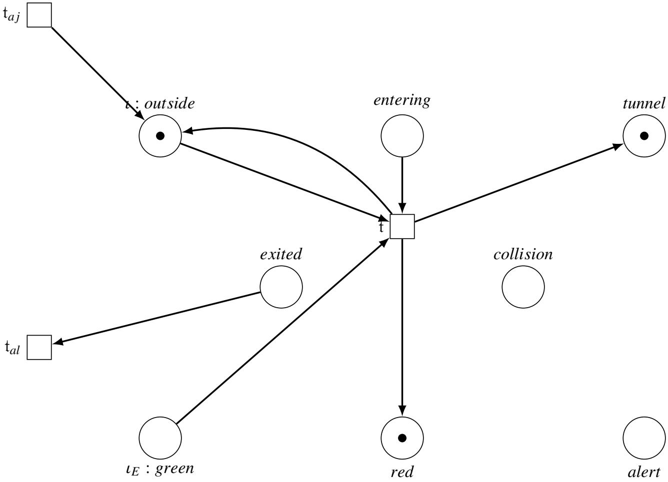

t_{aj}
ι : outside
entering
tunnel
exited
collision
ι_E : green
red
alert

Agent Leaving Tansition




t_{aj}
ι : outside
entering
tunnel
t
exited
collision
t_{al}
ι_E : green
red
alert

Joint Action



Correctness

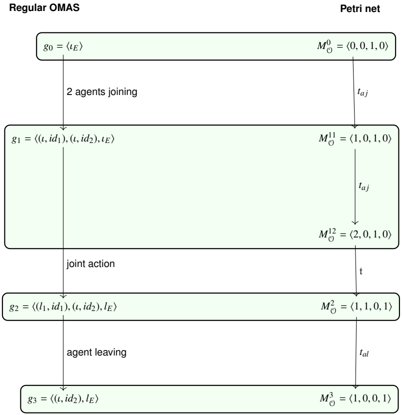


Figure: A run in Petri net from the run of regular OMAS

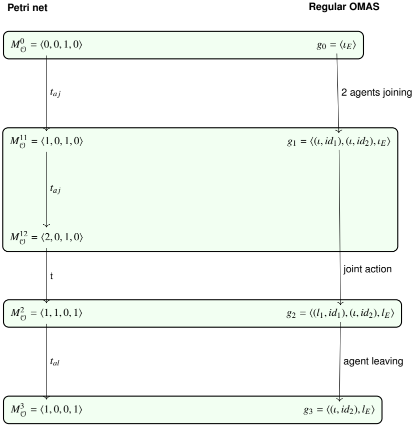
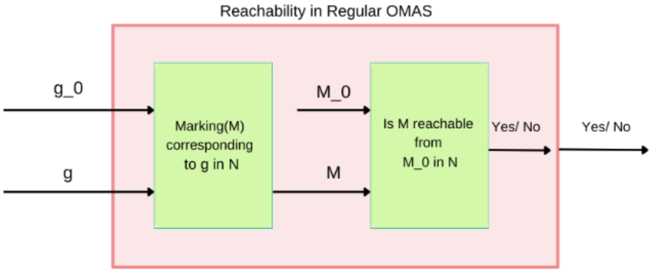


Figure: A run in regular OMAS from the run of Petri net


Reachability in Regular OMAS
g_0
g
Marking(M) corresponding to g in N
M_0
M
Is M reachable from M_0 in N
Yes/ No
Yes/ No

- A Regular OMAS $\mathcal{O}$ is safe if all reachable global states g from g_0 are safe.
- A global state $g = (l_1, id_1), (l_2, id_2), \ldots, l_e$ is safe if:

$$\forall l \in g \setminus l_e, V(l) = safe$$

Theorem

Safety checking in Regular OMAS is decidable.

Definition

A transition t in a Petri net (N, M_O) is said to be $L1 - live$ if t can be fired atleast once in some firing sequence in $\mathcal{L}(M_0)$[a],[b].

[a] **murata1989petri**.
[b] **livenessdecidability**.

Definition

A transition t in a Petri net (N, M_O) is said to be $L1 - live$ if t can be fired atleast once in some firing sequence in $\mathcal{L}(M_0)$[a],[b].

[a] **murata1989petri**.
[b] **livenessdecidability**.

1. A transition t in $T_{\mathcal{O}}$ is $\langle (l_1, a_1, A_1, l'_1), (l_2, a_2, A_2, l'_2), \ldots, (l_e, b, l'_e) \rangle$.
2. **Identify Unsafe Transitions:** Compute $\mathrm{t}_{\text{unsafe}}$:
   - Find the set of all the transitions, t_{unsafe} in $T_{\mathcal{O}}$ where $V(l') = \{\}$ in any of the tuple present in t.
3. **Check Liveness:** For each $\mathrm{t} \in \mathrm{t}_{\text{unsafe}}$, determine if it is $L1$-live:
   - If any t is $L1$-live $\rightarrow \mathcal{O}$ is unsafe.
   - Otherwise $\rightarrow \mathcal{O}$ is safe.

- Design and implement a tool for encoding Regular OMAS into Petri net.
- Explore the possibility of safety checking in OMAS
- **Explore Reverse Encoding:**
  - Investigate encoding Petri net into Regular OMAS.
  - Not possible? Identify **subclass of Petri net** that can be encoded into Regular OMAS.
- **Discover New Subclasses:**
  - Identify **interesting subclasses of OMAS**.
  - Establish corresponding **subclasses of Petri net**.


Thank you