Compositional Verification

VINO 2024 Seminar

Vincent Trélat

Université de Lorraine, CNRS, Inria, Loria, Nancy, France

November 3, 2025

Outline

Motivation

Until now:

(decision-making part of an) autonomous system $\approx$ a single agent

Motivation

Until now:

(decision-making part of an) autonomous system $\approx$ a single agent

Idea: generalize the approach to consider the whole system, including the environment.

Motivation

Until now:

(decision-making part of an) autonomous system $\approx$ a single agent

Idea: generalize the approach to consider the whole system, including the environment.

Motivation

A system may be **composed** of multiple parts interacting with each other:

autonomous system $\approx$ multiple agents

Motivation

A system may be **composed** of multiple parts interacting with each other:

autonomous system $\approx$ multiple agents

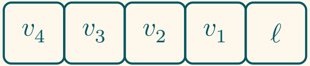
We want to verify each part **separately** and then **compose** the results.

Motivation
-Two guiding examples

1. Vehicle platooning system

Motivation
-Two guiding examples

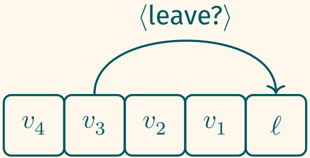
1. Vehicle platooning system



Motivation

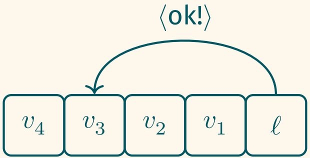
-Two guiding examples

1. Vehicle platooning system



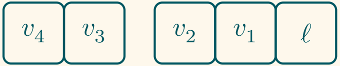
Motivation
-Two guiding examples

1. Vehicle platooning system



Motivation
-Two guiding examples

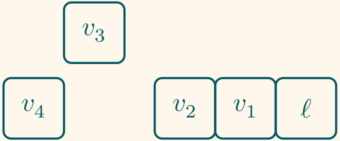
1. Vehicle platooning system



Motivation

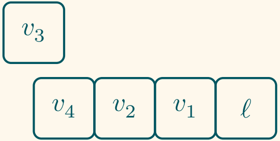
-Two guiding examples

1. Vehicle platooning system



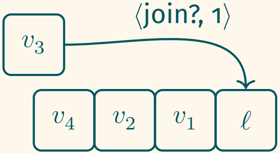
Motivation
-Two guiding examples

1. Vehicle platooning system



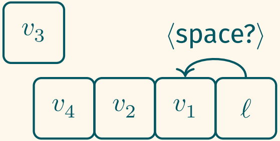
Motivation
-Two guiding examples

1. Vehicle platooning system



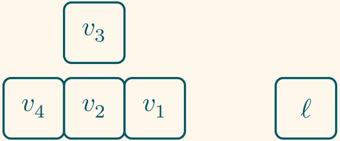
Motivation
-Two guiding examples

1. Vehicle platooning system



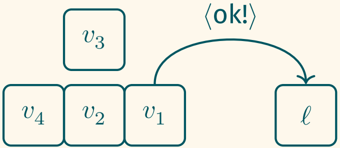
Motivation
-Two guiding examples

1. Vehicle platooning system



Motivation
-Two guiding examples

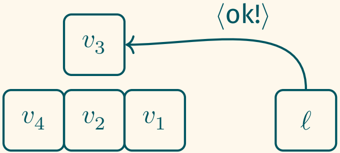
1. Vehicle platooning system



Motivation

-Two guiding examples

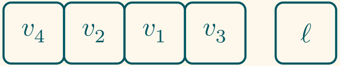
1. Vehicle platooning system



Motivation

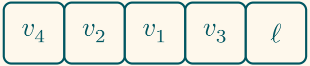
-Two guiding examples

1. Vehicle platooning system



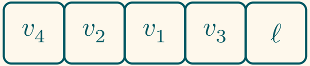
Motivation
-Two guiding examples

1. Vehicle platooning system



Motivation
-Two guiding examples

1. Vehicle platooning system



2. Autonomous search and rescue rover
   - rover
   - charger
   - routing planner
   - battery monitor

1 Motivation

2 How to compose?

- Vehicle platooning
  - Vehicle model
  - Verifying the platoon model
- Contract-based First Order Logic
- Autonomous search and rescue rover
  - Rover model
  - Verifying the rover model

How to compose?

-Vehicle platooning

A system is represented as the (parallel) composition of multiple agents.

How to compose?

-Vehicle platooning

A system is represented as the (parallel) composition of multiple agents.

Definition (Vehicle)

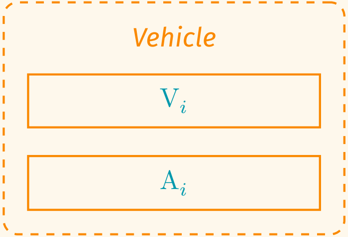
> A vehicle is represented as an **agent** V_i (TA abstracting vehicle control only, checked with UPPAAL) and an **agent** A_i (BDI automaton abstracting vehicle behavior only: checked with AJPF).

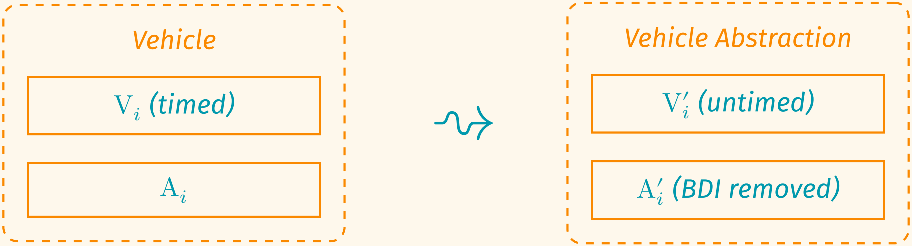
How to compose?
-Vehicle platooning

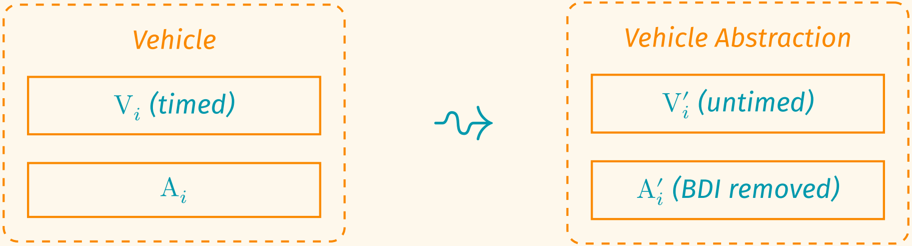
A system is represented as the (parallel) composition of multiple agents.

Definition (Vehicle)

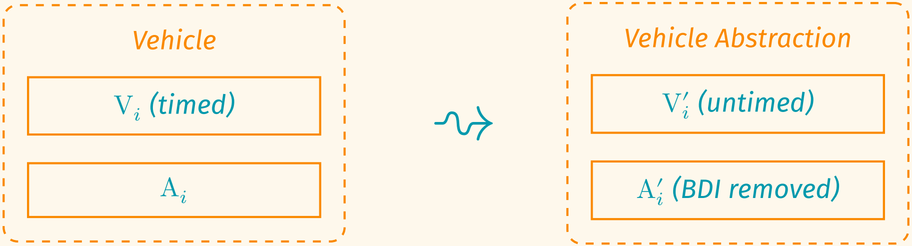
> A vehicle is represented as an **agent** V_i (TA abstracting vehicle control only, checked with UPPAAL) and an **agent** A_i (BDI automaton abstracting vehicle behavior only: checked with AJPF).




Vehicle
V_i (timed)
A_i
Vehicle Abstraction
V_i' (untimed)
A_i' (BDI removed)



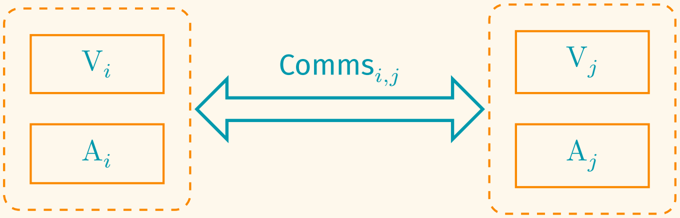
- $V_i \rightsquigarrow V_i'$: **untimed over-approximation** on the inputs (AJPF)
- $A_i \rightsquigarrow A_i'$: extracted model of the agent's behavior with **BDI removed** (UPPAAL)



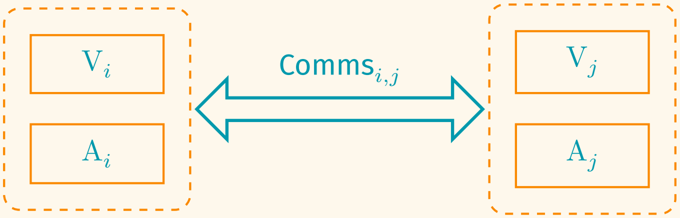
- $V_i \rightsquigarrow V'_i$: **untimed** **over-approximation** on the inputs (AJPF)
- $A_i \rightsquigarrow A'_i$: extracted model of the agent's behavior with **BDI removed** (UPPAAL)

We further assume that transitions are instantaneous in A_i.

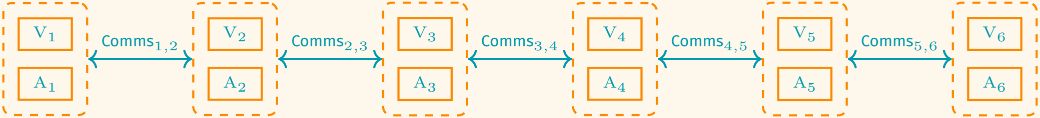
We also model communication between agents by a TA (checked with UPPAAL).



We also model communication between agents by a TA (checked with UPPAAL).



$\text{Comms}_{i,j}$ is also abstracted into an untimed automaton $\text{Comms}'_{i,j}$.


V_1
A_1
$\text{Comms}_{1,2}$
V_2
A_2
$\text{Comms}_{2,3}$
V_3
A_3
$\text{Comms}_{3,4}$
V_4
A_4
$\text{Comms}_{4,5}$
V_5
A_5
$\text{Comms}_{5,6}$
V_6
A_6

1. Motivation

2. How to compose?
   - Vehicle platooning
     - Vehicle model
     - Verifying the platoon model
   - Contract-based First Order Logic
   - Autonomous search and rescue rover
     - Rover model
     - Verifying the rover model

Definition (System properties)

For a system S composed of agents $A_1, \ldots, A_n$ over which we want to check φ, we separate φ into:

Definition (System properties)

For a system S composed of agents $A_1, \dots, A_n$ over which we want to check φ, we separate φ into:

- the agent properties φ_a

Definition (System properties)

For a system S composed of agents $A_1, \ldots, A_n$ over which we want to check φ, we separate φ into:

- the agent properties φ_a
- the timing properties φ_t

Definition (System properties)

For a system S composed of agents $A_1, \dots, A_n$ over which we want to check φ, we separate φ into:

- the agent properties φ_a
- the timing properties φ_t

Definition (Platoon Model)

A platoon model S is the parallel composition of the agents V_i and A_i and the communication automata $\text{Comms}_{i,j}$.

Definition (System properties)

For a system S composed of agents $A_1, \ldots, A_n$ over which we want to check φ, we separate φ into:

- the agent properties φ_a
- the timing properties φ_t

Definition (Platoon Model)

$$S := \mathrm{V}_1 \mid\mid \mathrm{A}_1 \mid\mid \textsf{Comms}_{1,2} \mid\mid \ldots \mid\mid \textsf{Comms}_{n-1,n} \mid\mid \mathrm{V}_n \mid\mid \mathrm{A}_n$$

Definition (System properties)

Checking a property φ over a platoon model S, i.e. $S \models \varphi$ is reduced to the following:

Definition (System properties)

Checking a property φ over a platoon model S, i.e. $S \models \varphi$ is reduced to the following:

- for each agent A_i, verify the agent properties φ_a on the agent within an over-approximating untimed environment

Definition (System properties)

Checking a property φ over a platoon model S, i.e. $S \models \varphi$ is reduced to the following:

- for each agent A_i, verify the agent properties φ_a on the agent within an over-approximating untimed environment
- verify the timing properties φ_t on the whole system where the agent program is replaced by an untimed automaton describing solely its input-output behaviour abstracting from internal BDI reasoning

Definition (System properties)

Checking a property φ over a platoon model S, i.e. $S \vDash \varphi$ is reduced to the following:

- for each agent A_i, verify the agent properties φ_a on the agent within an over-approximating untimed environment
- verify the timing properties φ_t on the whole system where the agent program is replaced by an untimed automaton describing solely its input-output behaviour abstracting from internal BDI reasoning

Theorem (Compositionality)

If:

- $\forall A_i \in S \setminus \{A_n\} \cdot \mathrm{V}'_i \mid\mid \mathrm{A}_i \mid\mid \mathit{Comms}'_{i,i+1} \vDash \varphi_a$
- $\mathrm{V}_1 \mid\mid \mathrm{A}'_1 \mid\mid \mathit{Comms}_{1,2} \mid\mid \ldots \mid\mid \mathit{Comms}_{n-1,n} \mid\mid \mathrm{V}_n \mid\mid \mathrm{A}'_n \vDash \varphi_t$

Then, $S \vDash \varphi$.

Definition (System properties)

Checking a property φ over a platoon model S, i.e. $S \models \varphi$ is reduced to the following:

- $\forall A_i \in S \setminus \{A_n\} \cdot \mathrm{V}'_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}'_{i,i+1} \models \varphi_a$
- verify the timing properties φ_t on the whole system where the agent program is replaced by an untimed automaton describing solely its input-output behaviour abstracting from internal BDI reasoning

Theorem (Compositionality)

If:

- $\forall A_i \in S \setminus \{A_n\} \cdot \mathrm{V}'_i \mid\mid \mathrm{A}_i \mid\mid \mathit{Comms}'_{i,i+1} \models \varphi_a$
- $\mathrm{V}_1 \mid\mid \mathrm{A}'_1 \mid\mid \mathit{Comms}_{1,2} \mid\mid \ldots \mid\mid \mathit{Comms}_{n-1,n} \mid\mid \mathrm{V}_n \mid\mid \mathrm{A}'_n \models \varphi_t$

Then, $S \models \varphi$.

Definition (System properties)

Checking a property φ over a platoon model S, i.e. $S \models \varphi$ is reduced to the following:

- $\forall A_i \in S \setminus \{A_n\} \cdot \mathrm{V}'_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}'_{i,i+1} \models \varphi_a$
- $\mathrm{V}_1 \mid\mid \mathrm{A}'_1 \mid\mid \mathsf{Comms}_{1,2} \mid\mid \dots \mid\mid \mathsf{Comms}_{n-1,n} \mid\mid \mathrm{V}_n \mid\mid \mathrm{A}'_n \models \varphi_t$

Theorem (Compositionality)

If:

- $\forall A_i \in S \setminus \{A_n\} \cdot \mathrm{V}'_i \mid\mid \mathrm{A}_i \mid\mid \mathit{Comms}'_{i,i+1} \models \varphi_a$
- $\mathrm{V}_1 \mid\mid \mathrm{A}'_1 \mid\mid \mathit{Comms}_{1,2} \mid\mid \dots \mid\mid \mathit{Comms}_{n-1,n} \mid\mid \mathrm{V}_n \mid\mid \mathrm{A}'_n \models \varphi_t$

Then, $S \models \varphi$.

Proof of compositionality

Hypotheses:

$$\forall A_i \in S \setminus \{A_n\} \cdot \mathrm{V}'_i \parallel \mathrm{A}_i \parallel \mathsf{Comms}'_{i,i+1} \models \varphi_a \quad (1)$$

$$\mathrm{V}_1 \parallel \mathrm{A}'_1 \parallel \mathsf{Comms}_{1,2} \parallel \ldots \parallel \mathsf{Comms}_{n-1,n} \parallel \mathrm{V}_n \parallel \mathrm{A}'_n \models \varphi_t \quad (2)$$

Proof of compositionality

Hypotheses:

$$\forall A_i \in S \setminus \{A_n\} \cdot \mathrm{V}'_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}'_{i,i+1} \vDash \varphi_a \quad (1)$$

$$\mathrm{V}_1 \mid\mid \mathrm{A}'_1 \mid\mid \mathsf{Comms}_{1,2} \mid\mid \ldots \mid\mid \mathsf{Comms}_{n-1,n} \mid\mid \mathrm{V}_n \mid\mid \mathrm{A}'_n \vDash \varphi_t \quad (2)$$

By definition, $\mathrm{V}'_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}'_{i,i+1}$ over-approximates $\mathrm{V}_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}_{i,i+1}$, thus (1) implies

$$\mathrm{V}_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}_{i,i+1} \vDash \varphi_a$$

Therefore, $S \vDash \varphi_a$.

Proof of compositionality

Hypotheses:

$$\forall A_i \in S \setminus \{A_n\} \cdot \mathrm{V}'_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}'_{i,i+1} \vDash \varphi_a \tag{1}$$

$$\mathrm{V}_1 \mid\mid \mathrm{A}'_1 \mid\mid \mathsf{Comms}_{1,2} \mid\mid \ldots \mid\mid \mathsf{Comms}_{n-1,n} \mid\mid \mathrm{V}_n \mid\mid \mathrm{A}'_n \vDash \varphi_t \tag{2}$$

By definition, $\mathrm{V}'_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}'_{i,i+1}$ over-approximates $\mathrm{V}_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}_{i,i+1}$, thus (1) implies

$$\mathrm{V}_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}_{i,i+1} \vDash \varphi_a$$

Therefore, $S \vDash \varphi_a$.

By assumption, A'_i and A_i are time-equivalent, thus (2) can be rewritten as

$$\mathrm{V}_1 \mid\mid \mathrm{A}_1 \mid\mid \mathsf{Comms}_{1,2} \mid\mid \ldots \mid\mid \mathsf{Comms}_{n-1,n} \mid\mid \mathrm{V}_n \mid\mid \mathrm{A}_n \vDash \varphi_t$$

Proof of compositionality

Hypotheses:

$$\forall A_i \in S \setminus \{A_n\} \cdot \mathrm{V}'_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}'_{i,i+1} \models \varphi_a \quad (1)$$

$$\mathrm{V}_1 \mid\mid \mathrm{A}'_1 \mid\mid \mathsf{Comms}_{1,2} \mid\mid \ldots \mid\mid \mathsf{Comms}_{n-1,n} \mid\mid \mathrm{V}_n \mid\mid \mathrm{A}'_n \models \varphi_t \quad (2)$$

By definition, $\mathrm{V}'_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}'_{i,i+1}$ over-approximates $\mathrm{V}_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}_{i,i+1}$, thus (1) implies

$$\mathrm{V}_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}_{i,i+1} \models \varphi_a$$

Therefore, $S \models \varphi_a$.

By assumption, A'_i and A_i are time-equivalent, thus (2) can be rewritten as

$$S \models \varphi_t$$

Proof of compositionality

Hypotheses:

$$\forall A_i \in S \setminus \{A_n\} \cdot \mathrm{V}'_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}'_{i,i+1} \models \varphi_a \tag{1}$$

$$\mathrm{V}_1 \mid\mid \mathrm{A}'_1 \mid\mid \mathsf{Comms}_{1,2} \mid\mid \dots \mid\mid \mathsf{Comms}_{n-1,n} \mid\mid \mathrm{V}_n \mid\mid \mathrm{A}'_n \models \varphi_t \tag{2}$$

Finally,

$$S \models \varphi_a \wedge S \models \varphi_t$$

Proof of compositionality

Hypotheses:

$$\forall A_i \in S \setminus \{A_n\} \cdot \mathrm{V}'_i \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}'_{i,i+1} \models \varphi_a \tag{1}$$

$$\mathrm{V}_1 \mid\mid \mathrm{A}'_1 \mid\mid \mathsf{Comms}_{1,2} \mid\mid \ldots \mid\mid \mathsf{Comms}_{n-1,n} \mid\mid \mathrm{V}_n \mid\mid \mathrm{A}'_n \models \varphi_t \tag{2}$$

Finally,

$$S \models \varphi_a \wedge \varphi_t$$

Proof of compositionality

Hypotheses:

$$\forall A_i \in S \setminus \{A_n\} \cdot \mathrm{V}_i' \mid\mid \mathrm{A}_i \mid\mid \mathsf{Comms}'_{i,i+1} \models \varphi_a \tag{1}$$

$$\mathrm{V}_1 \mid\mid \mathrm{A}_1' \mid\mid \mathsf{Comms}_{1,2} \mid\mid \ldots \mid\mid \mathsf{Comms}_{n-1,n} \mid\mid \mathrm{V}_n \mid\mid \mathrm{A}_n' \models \varphi_t \tag{2}$$

Finally,

$$S \models \varphi$$

■

Applying the result

Applying the result

AJPF *If a vehicle with a goal of joining the platoon never believes it has received confirmation from the leader, then it never initiates joining to the platoon.*

Applying the result

AJPF $\Box(\mathbf{G}_v\mathsf{platoonM}(v,\ell) \implies \neg\mathbf{A}_v\mathsf{perf}(\mathsf{changingLane}(1)) \mathrm{R}\, \mathbf{B}_v\mathsf{joinAgr}(v,\ell))$

Applying the result

AJPF $\square(\mathbf{G}_v \mathsf{platoonM}(v,\ell) \implies \neg \mathbf{A}_v \mathsf{perf}(\mathsf{changingLane}(1)) \,\mathrm{R}\, \mathbf{B}_v \mathsf{joinAgr}(v,\ell))$

UPPAAL *If an agent ever receives a joining agreement from the leader, then the preceding agent has increased its space to its front agent.*

Applying the result

AJPF $\Box(\mathbf{G}_v \mathsf{platoonM}(v,\ell) \implies \neg\mathbf{A}_v \mathsf{perf}(\mathsf{changingLane}(1))\ \mathrm{R}\ \mathbf{B}_v \mathsf{joinAgr}(v,\ell))$

UPPAAL $\mathbf{A}\Box((\mathrm{A}_i.\mathtt{rdyChLane} \wedge \ell.\mathtt{joinVeh.front} = j) \implies (\mathrm{A}_j.\mathtt{incrSpacing} \wedge \mathrm{A}_j.\mathtt{spacingDone}))$

Applying the result

AJPF $\square(\mathbf{G}_v \mathsf{platoonM}(v,\ell) \implies \neg\mathbf{A}_v \mathsf{perf}(\mathsf{changingLane}(1)) \mathrm{R}\, \mathbf{B}_v \mathsf{joinAgr}(v,\ell))$

UPPAAL $\mathbf{A}\square((\mathrm{A}_i.\mathtt{rdyChLane} \wedge \ell.\mathtt{joinVeh.front} = j) \implies (\mathrm{A}_j.\mathtt{incrSpacing} \wedge \mathrm{A}_j.\mathtt{spacingDone}))$

Combining both properties thanks to the composition theorem, we get:

Applying the result

AJPF $\Box(\mathbf{G}_v \mathsf{platoonM}(v,\ell) \implies \neg\mathbf{A}_v \mathsf{perf}(\mathsf{changingLane}(1)) \,\mathrm{R}\, \mathbf{B}_v \mathsf{joinAgr}(v,\ell))$

UPPAAL $\mathbf{A}\Box((\mathrm{A}_i.\mathtt{rdyChLane} \wedge \ell.\mathtt{joinVeh.front} = j) \implies (\mathrm{A}_j.\mathtt{incrSpacing} \wedge \mathrm{A}_j.\mathtt{spacingDone}))$

Combining both properties thanks to the composition theorem, we get:

If a vehicle never believes it has received confirmation from the leader, then it never initiates joining to the platoon. ***and*** *If an agent ever receives a joining agreement from the leader, then the preceding agent has increased its space to its front agent.*

Applying the result

AJPF $\Box(\mathbf{G}_v \mathsf{platoonM}(v,\ell) \implies \neg \mathbf{A}_v \mathsf{perf}(\mathsf{changingLane}(1))\ \mathrm{R}\ \mathbf{B}_v \mathsf{joinAgr}(v,\ell))$

UPPAAL $\mathbf{A}\Box((\mathrm{A}_i.\texttt{rdyChLane} \wedge \ell.\texttt{joinVeh.front} = j) \implies (\mathrm{A}_j.\texttt{incrSpacing} \wedge \mathrm{A}_j.\texttt{spacingDone}))$

Combining both properties thanks to the composition theorem, we get:

An agent never initiates joining the platoon unless the preceding agent has increased its space to its front agent.

1 Motivation

2 How to compose?

- Vehicle platooning
  - Vehicle model
  - Verifying the platoon model
- Contract-based First Order Logic
- Autonomous search and rescue rover
  - Rover model
  - Verifying the rover model

Definition

Let $\mathcal{M}$ be a set of modules. Let $C \in \mathcal{M}$. A contract is a tuple $\langle \mathcal{I}_C, \mathcal{U}_C, \mathcal{A}_C, \mathcal{G}_C \rangle$ where:

- $\mathcal{I}_C$ is a set of input modules
- $\mathcal{U}_C$ is a set of updates
- $\mathcal{A}_C$ is an assumption predicate
- $\mathcal{G}_C$ is a guarantee predicate

Definition

Let $\mathcal{M}$ be a set of modules. Let $C \in \mathcal{M}$. A contract is a tuple $\langle \mathcal{I}_C, \mathcal{U}_C, \mathcal{A}_C, \mathcal{G}_C \rangle$ where:

- $\mathcal{I}_C$ is a set of input modules
- $\mathcal{U}_C$ is a set of updates
- $\mathcal{A}_C$ is an assumption predicate
- $\mathcal{G}_C$ is a guarantee predicate

C is assumed to obey the following:

$$\forall \varphi, \bar{x} \cdot \bar{x} \subseteq \Sigma \setminus \mathcal{U}_C \wedge \mathcal{A}_C \wedge C^{\downarrow} \wedge \varphi(\bar{x}) \implies \Diamond(\mathcal{G}_C \wedge C^{\uparrow} \wedge \varphi(\bar{x})) \quad \text{(module execution)}$$

$$C_i^{\uparrow} \wedge C_i \in \mathcal{I}_{C_j} \implies C_j^{\downarrow} \quad \text{(chain)}$$

Rover model

The rover is composed of:

- a goal-reasoning agent
- a planner module (not an agent but formally described)
- a plan execution agent
- an agent abstracting sensor data

1 Motivation

2 How to compose?

- Vehicle platooning
  - Vehicle model
  - Verifying the platoon model
- Contract-based First Order Logic
- Autonomous search and rescue rover
  - Rover model
  - Verifying the rover model

Goal Reasoning Agent Contract

- **Inputs**: vision sensor V, heat sensor H, execution agent E

Goal Reasoning Agent Contract

- **Inputs**: vision sensor V, heat sensor H, execution agent E
- **Updates**: target goal g

Goal Reasoning Agent Contract

- **Inputs**: vision sensor V, heat sensor H, execution agent E
- **Updates**: target goal g
- **Assumptions**: $\top$

Goal Reasoning Agent Contract

- **Inputs**: vision sensor V, heat sensor H, execution agent E
- **Updates**: target goal g
- **Assumptions**: $\top$
- **Guarantees**:

$$
\begin{aligned}
&(g \neq \texttt{chargePos} \implies \\
&\quad (\exists h \in \mathbb{N} \cdot (g, h) \in \texttt{GoalSet} \land (\forall p, h_1 \cdot (p, h_1) \in \texttt{GoalSet} \implies h \geq h_1))) \\
&\land (\texttt{recharge} \iff g = \texttt{chargePos})
\end{aligned}
$$

We define contracts for the **goal-reasoning agent**, the **planner module** and the **plan execution agent**.

We define contracts for the **goal-reasoning agent**, the **planner module** and the **plan execution agent**.

We check with AJPF that each agent meets its contract given its specification.

We define contracts for the **goal-reasoning agent**, the **planner module** and the **plan execution agent**.

We check with AJPF that each agent meets its contract given its specification.

Really close to Event-B!

We define contracts for the **goal-reasoning agent**, the **planner module** and the **plan execution agent**.

We check with AJPF that each agent meets its contract given its specification.

Really close to Event-B!

In the end, we are able to compositionally verify properties like

> *If at any point all plans sent to the plan execution agent by the planner module are longer than available battery power, then eventually the current plan will contain the charging position as the goal or there is no route to the charging position.*

Conclusion

Both approaches provide a way to compositionally verify complex systems

Conclusion

Both approaches provide a way to compositionally verify complex systems

- *individual agent* verification using AJPF and *combined timed behavior* verification using UppAal

Conclusion

Both approaches provide a way to compositionally verify complex systems

- *individual agent* verification using AJPF and *combined timed behavior* verification using UPPAAL
- *contract-based* unifying logic, contract-level verification using AJPF