Alternating Signal Temporal Logic

IFIP WG1.3

ifip

An IFIP TC1 Working Group

Holger Schlingloff

Institut für Informatik der Humboldt Universität

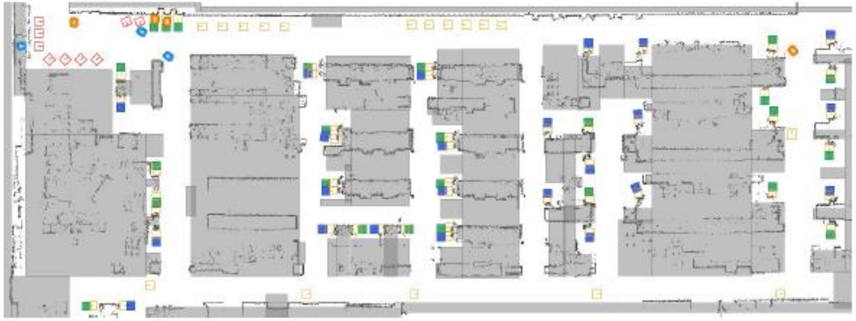
System Quality Center, Fraunhofer FOKUS

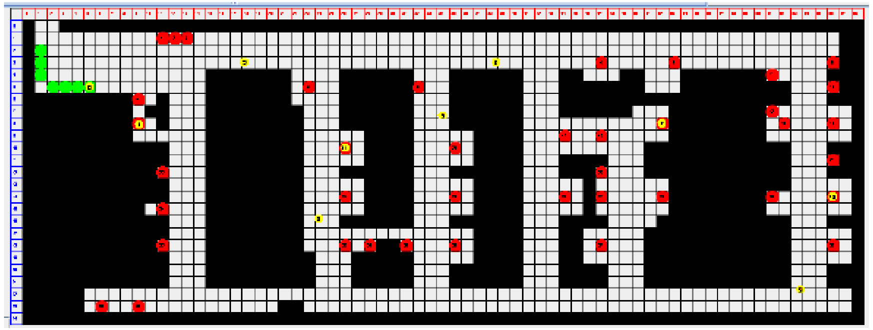
IFIP WG 1.3, Zoom, 19.1.2022

Motivation

Foundations of Systems Specification

- Formal methods in real applications
- Scheduling of transport robots
- Controller design
- Specification and verification of industrial systems





Structure of this talk

- More motivation
- Concurrent game structures and strategic logics
- Hybrid automata and signal temporal logics
- Combining strategic and continuous systems and logics
- Model checking results
- Perspectives and outlook

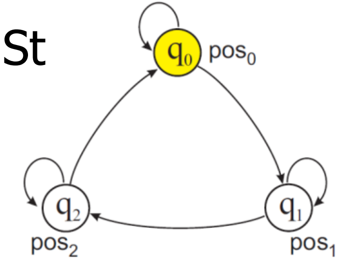
Industrial Multi-Agent Systems

- Real-time / hybrid
  - Continuous sensor inputs, discrete control, continuous motor outputs
- Inherently distributed, space might be important
  - Unreliable communication, intrusion
- "Intelligent" and autonomous
  - Beliefs, intentions, desires; fuzzy goals

- Research questions
  - How to model such systems?
  - How to specify properties?
  - How to synthesize winning strategies?

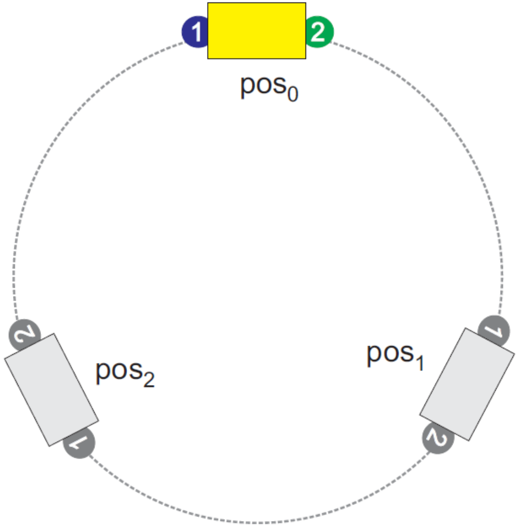
Concurrent Game Structures

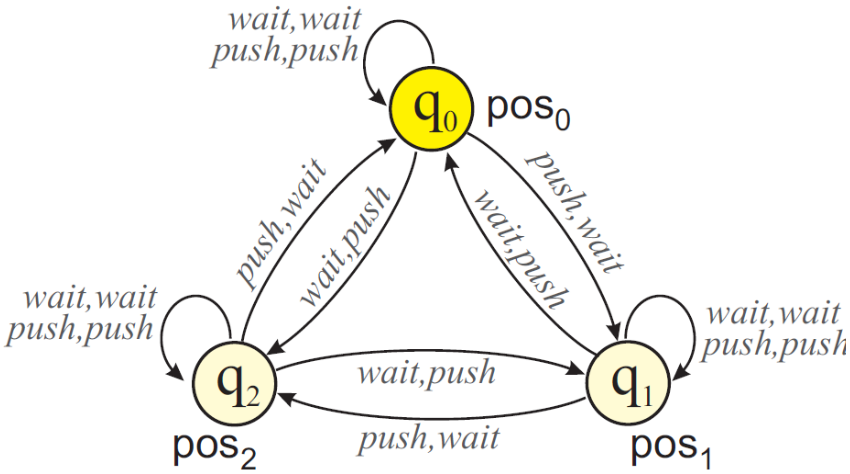
- Classical definitions
  - Transition system TS = (St, δ, s_0), $\delta \subseteq St \times St$, $s_0 \in St$
  - Labelled transition system LTS = (St, Act, δ, s_0), $\delta \subseteq St \times Act \times St$
  - Kripke structure KS = (St, Prop, δ, Int, s_0), $Int \subseteq St \times Prop$
  - Finite state machine, Büchi/Rabin/Muller automaton, …



- Several LTS's: ($LTS_1 \times \ldots \times LTS_n$)
  - Product transition system – synchronization via shared actions
  - Concurrent game structure CGS = (Agt, St, Act, δ, s_0), $\delta \subseteq St \times 2^{(Agt \times Act)} \times St$
    combined action $\alpha = \{(a_1, \alpha_{a1}), \ldots, (a_n, \alpha_{an})\}$, every agent at most one action
    ("agent a_1 chooses α_{a1} and … and agent a_n chooses α_{an}")

Example: Robots and Carriage





Thanks to Wojtek Jamroga and Wojtek Penczek

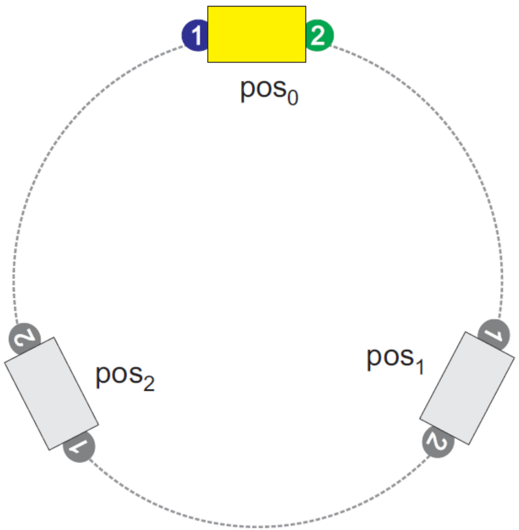
- Variants
  - Deterministic CGS with availability relation DCGS = (Agt, St, Act, *avail*, δ, s_0), *avail* $\subseteq$ Agt x St x Act, δ: St x $2^{(Agt \times Act)} \rightarrow$ St is a function, (s, α , s') $\in$ δ only if (a, s, α_a) $\in$ *avail*
  - Synchronous game structure: all agents have to choose an action in each state (that is, δ: St x $Act^n \rightarrow$ St)
  - Coalitional game structure: agents can form a coalition and choose their actions synchronously
  - Turn-based game structure: agents or coalitions take turn with their actions; TCDCGS = (Agt, St, Act, *avail*, δ, *turn*, s_0), *turn*: St $\rightarrow$ Agt; if *turn* (s) = a_i, then *avail* (a_i, s) $\neq \emptyset$, and for all j≠i, *avail* (a_j, s) = $\emptyset$.

Goals and Strategies

- In the simple case, a goal is a designated set of states
  - more advanced goals can be described by logic, cf. later
- Strategy $\mathbf{s}_i$ for a_i in a DCGS $\mathbf{s}_i$: St → Act s.t. $(a_i, s, \mathbf{s}_i(s)) \in avail$
  - in case $\{\alpha_a \mid (a_i, s, \alpha_a) \in avail\} = \emptyset$, there is no strategy for a_i
- Combined action α is consistent with the strategies $\{\mathbf{s}_1 \ldots \mathbf{s}_k\}$ for agents $\{a_1 \ldots a_k\}$ in state s if $(a_i, \mathbf{s}_i(s)) \in \alpha$ for all i
- An execution σ following strategies $\{\mathbf{s}_1, \ldots, \mathbf{s}_k\}$ for agents $\{a_1, \ldots, a_k\}$ is an execution $(s_0, \alpha_0, s_1, \alpha_1, \ldots)$, where every α_i is consistent with the strategies in s_i
  - Given strategies for all agents in a DCGS, there is only one possible execution following all these strategies (usually called the outcome)

Example continued

- In the initial state, does Robot_1 have a strategy to bring the carriage into position pos_1?
- Do Robot_1 and Robot_2 have a combined strategy to reach any desired position?
- Does Robot_1 have a strategy to <u>avoid</u> pos_1?



Alternating Temporal Logic

- Syntax: $\langle\langle A \rangle\rangle \varphi$
  where A is a set of agents and φ is an LTL formula
- Semantics: $\mathcal{M} \models \langle\langle A \rangle\rangle \varphi$ iff for each a_i in A there is a strategy $\mathbf{s_i}$ such that in each execution σ of **M** which follows all these $\mathbf{s_i}$, it holds that $\sigma \models \varphi$
  - Note that $\langle\langle a_1, a_2 \rangle\rangle\ \varphi$ is not the same as $\langle\langle a_1 \rangle\rangle\ \langle\langle a_2 \rangle\rangle\ \varphi$!

- Examples
  - $\langle\langle Robot_1 \rangle\rangle\, \mathbf{F}\, pos_1$ – $Robot_1$ has a strategy to get to pos_1
  - $\langle\langle \{Robot_1, Robot_2\} \rangle\rangle\, \mathbf{F}\, pos_1$ – A combined strategy to reach pos_1
  - $\langle\langle Robot_1 \rangle\rangle\, \mathbf{G}\, \neg pos_1$ – A strategy to avoid pos_1

ATL Complexity and Algorithm

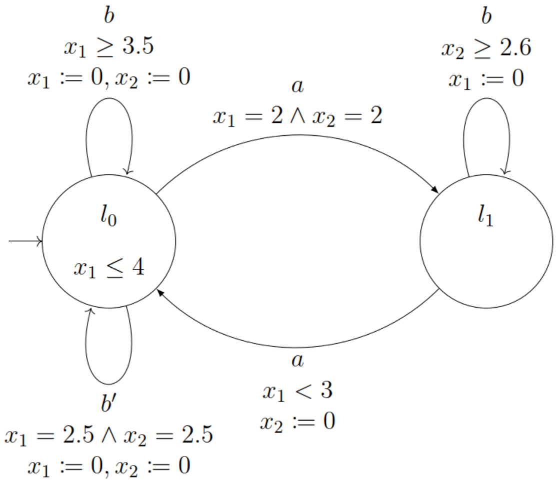
- Variants
  - ATL: every temporal operator preceded by exactly one cooperation modality
  - ATL*: no syntactic restriction
- Complexity results
  - ATL on CGS is in **P** (fixpoint unwinding)
  - ATL on CEGS with memory is undecidable

	Ir	IR	ir	iR
Simple $\mathscr{L}_{CL}$	Σ_2^P	Σ_2^P	Σ_2^P	Σ_2^P
$\mathscr{L}_{CL}$	Δ_3^P	Δ_3^P	Δ_3^P	Δ_3^P
$\mathscr{L}_{ATL}$	Δ_3^P	Δ_3^P	Δ_3^P	Undecidable†
$\mathscr{L}_{ATL^+}$	Δ_3^P	*PSPACE*	Δ_3^P	Undecidable†
$\mathscr{L}_{ATL^*}$	*PSPACE*	*2EXPTIME*	*PSPACE*	Undecidable†

- Algorithms
  - Calta, Schlingloff (2010): $O(l*n^2*3^{(2*n*a^k/3)})$-algorithm for CEGSs (l strategic modalities, n states, k agents, a actions)
  - Lomuscio et al (2015): MCMAS tool for ATLK on CGS (ISPL)

Timed and Hybrid Automata

- Basically, a HA is an LTS with additional real-valued variables
  - Finite set of *locations* - a *state* is a location plus a variable valuation
  - ➔ infinitely many states!
  - A *region* is a (convex) set of states
  - Locations and/or transitions may be constrained by equations using variables
  - Variables and their derivatives may be assigned values



- Timed automata (TA)
  - A clock is a variable where the derivative is always 1
  - All clocks always advance with the same speed, no stopwatches

Signal Temporal Logic

- Syntax and semantics borrowed from interval temporal logic
- Basic propositions: (s ~ c)
  - s is a real-valued signal (i.e., variable or clock), c is a constant
- Boolean junctors, interval temporal until:
  - $(S,t) \models (\varphi_1 \, \mathbf{U}_I \, \varphi_2) \longleftrightarrow \exists t_1 \in t + I \; (S,t_1) \models \varphi_2 \text{ and } \forall t_2 \in [t,t_1) \; (S,t_2) \models \varphi_1$
- Examples
  - $\mathbf{G}_{(0,\infty)}(sensor.USfront \leq 5)$ – (no crash)
  - $\mathbf{F}_{(0,30]}((clock \geq 300) \vee (collect = true)))$ – (finding items in time)
  - $\mathbf{G}_{[0,\infty)}((sensor.r \geq 200) \rightarrow ((wheels = -5) \, \mathbf{U}_{(2,4)} \, (sensor.r \leq 200)))$ – (how to drive)

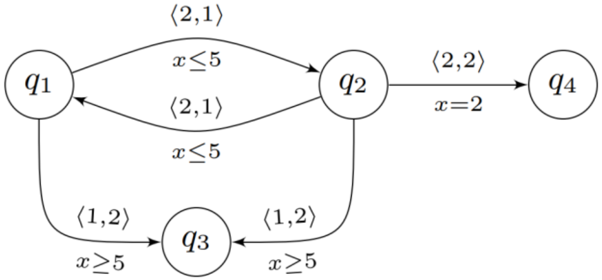
Model Checking STL on HA

- Reachability for hybrid automata is undecidable
  - but can be easily expressed in the considered logics
  - ➔ no hope to come up with a terminating model checking algorithm
- Reachability for timed automata is decidable (PSPACE-complete)
  - model checking for STL on TA can be reduced to this problem
  - Region-Graph and Difference-Bound Matrix construction of [Yovine 97, Alur 98]
- “Rectangular” hybrid automata
  - first derivative is bounded by constants
  - LTL model checking on RHA is in PSPACE

Combining Continuity and Strategies

Timed and hybrid CGS

- A fusion of CGS and timed / hybrid automata
- Formally, an automaton with dedicated actions for each player
  - Transitions are labelled with concurrent actions and activation conditions on the clocks or continuous variables
- Additional requirements: determinism, non-Zeno-ness
- Given a choice of strategies for all players, there is at most one run of the automaton



http://www.lsv.fr/Projects/anr-dots/PUBLIS/BLMO-concur07.pdf

Alternating Signal Temporal Logic

- Distinguish between monitored and controlled variables (sensors and actuators)
- Strategic decisions concern controlled variables

$$
\begin{aligned}
(\mathcal{T}, Q_0) &\models \top && \longleftrightarrow \forall l \in L\; Q_0^l \models I(l) \\
(\mathcal{T}, Q_0) &\models x \sim c && \longleftrightarrow Q_0 \models x \sim c \text{ and } \forall l \in L\; Q_0^l \models I(l) \\
(\mathcal{T}, Q_0) &\models \neg\varphi && \longleftrightarrow (\mathcal{T}, Q_0) \not\models \varphi \text{ and } \forall l \in L\; Q_0^l \models I(l) \\
(\mathcal{T}, Q_0) &\models (\varphi_1 \vee \varphi_2) && \longleftrightarrow (\mathcal{T}, Q_0) \models \varphi_1 \text{ or } (\mathcal{T}, Q_0) \models \varphi_2 \\
(\mathcal{T}, Q_0) &\models \langle\!\langle A \rangle\!\rangle\, (\varphi_1 \,\mathbf{U}_I\, \varphi_2) && \longleftrightarrow \exists F_A\, \forall \zeta \in \mathrm{sim}_{\mathcal{T}}\, (Q_0, F_A)\, \exists i \in \mathbb{N}\, \exists j \in \mathbb{N} \\
& && \quad \exists t_1 \in I\; (\mathcal{T}, \{\zeta(t_1, i)\}) \models \varphi_2 \\
& && \quad \wedge \forall t_2 \in [\inf(I), t_1)\; (\mathcal{T}, \{\zeta(t_2, j)\}) \models \varphi_1
\end{aligned}
$$

from the BA thesis of Sami Kharma, Nov. 2021

Model Checking ASTL on Timed Games

- Region-equivalence construction can be lifted to timed CGS
  - model checking for TATL on timed games is exponential [BLMO 07]
  - our result: lifting results to ASTL
  - still unclear: the case of ASTL*
- ASTL on rectangular hybrid games is still open
  - but we believe that it could be done

	algo. compl. w.r.t. ϕ and $\mathcal{T}$	theoretical complexity
ATL*	$2^{2^{O(\vert\phi\vert)}} \cdot 2^{O(\vert\mathcal{T}\vert)}$	2EXPTIME-complete
TATL	$2^{O(\vert\phi\vert) \cdot O(\vert\mathcal{T}\vert)}$	EXPTIME-complete
TALTL	$2^{2^{O(\vert\phi\vert)}} \cdot 2^{O(\vert\mathcal{T}\vert)}$	2EXPTIME-complete

from Brihaye, Laroussinie, Markey, Oreiby: Timed Concurrent Game Structures

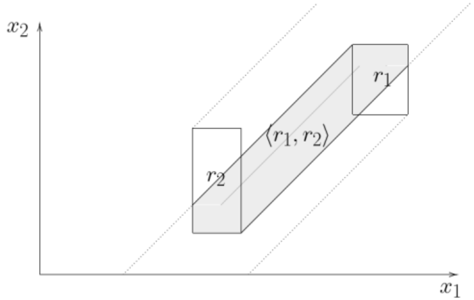
Algorithm

Algorithm 1 ASTL symbolic model-checking

Input: timed game $\mathcal{T}$, ASTL formula φ
Output: boolean *true* or *false*

for φ' **in** $\mathrm{Sub}(\varphi)$ **do**
    case $\varphi' = \top$
        $[\varphi'] \leftarrow Q_I$
    case $\varphi' = x \sim c$
        $[\varphi'] \leftarrow \mathrm{Reg}_{\mathcal{T}}(x \sim c)$
    case $\varphi' = \neg\theta$
        $[\varphi'] \leftarrow Q_I \setminus [\theta]$
    case $\varphi' = (\theta_1 \vee \theta_2)$
        $[\varphi'] \leftarrow [\theta_1] \cup [\theta_2]$
    case $\varphi' = \langle\langle A \rangle\rangle\, (\theta_1\, \mathbf{U}_I\, \theta_2)$
        $[\varphi'] \leftarrow \mathrm{Pre}^*_{\mathcal{T},I}(A, [\theta_2], [\theta_1])$
end for
return $q_0 \in [\varphi]$

- Pre-image calculation: for a timed game, set of players, interval, returnes the set of states from which a goal state can be reached with one decision, traversing only given intermediate states
- can be calculated by algebraic considerations

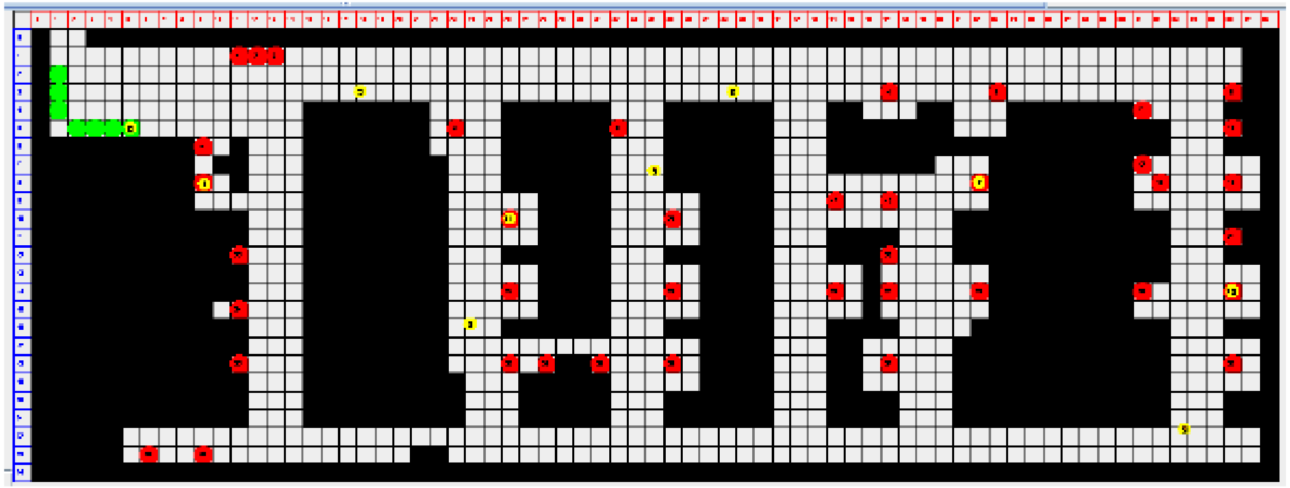


Experimental Results

- Yet unavailable

Back to the Transport Robot Example

- MCMAS model
- Robots have a joint strategy to accomplish all transport jobs in time: checked
- „Fuzzy" properties
  - Average and maximal waiting time should be as low as possible
  - Robots should keep their battery charged between 40% and 60%, if possible
  - Robots should provides approximately equal wear and tear within the fleet
  - Target properties are in Pareto equilibrium
  - "Reasoning about Quality and Fuzziness of Strategic Behaviours" [Bouyer et al. 2019]



Conclusion

- Presented a new combination of strategic and continuous logic
  - model checking algorithm
- Well-suited to specify and model certain properties
  - real time, interactivity
- Models for industrial control tasks are often much more complex
  - no firm goals, but approximate targets

- Vision: Verify control program with respect to the objectives
- Dream: Generate control programs automatically from the rules

Thank you for your attention!