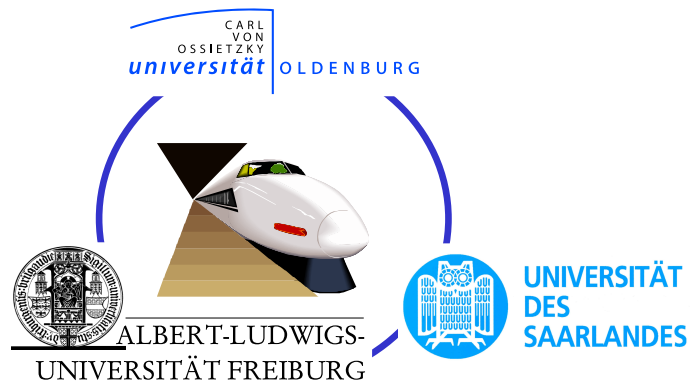


Abstract Interpretation of Graph Transformation

Jörg Bauer

Universität des Saarlandes

Dagstuhl Seminar 06161



A Sample Dynamic System: Platoons

- Cars on a highway build Platoons
- Platoons consist of
 1. One leader and a set of followers
 2. A single leader called free agent
- Star-like communication between a leader and its followers
- Inter-platoon communication between leaders only
- Platoons can merge and split
- Unbounded creation

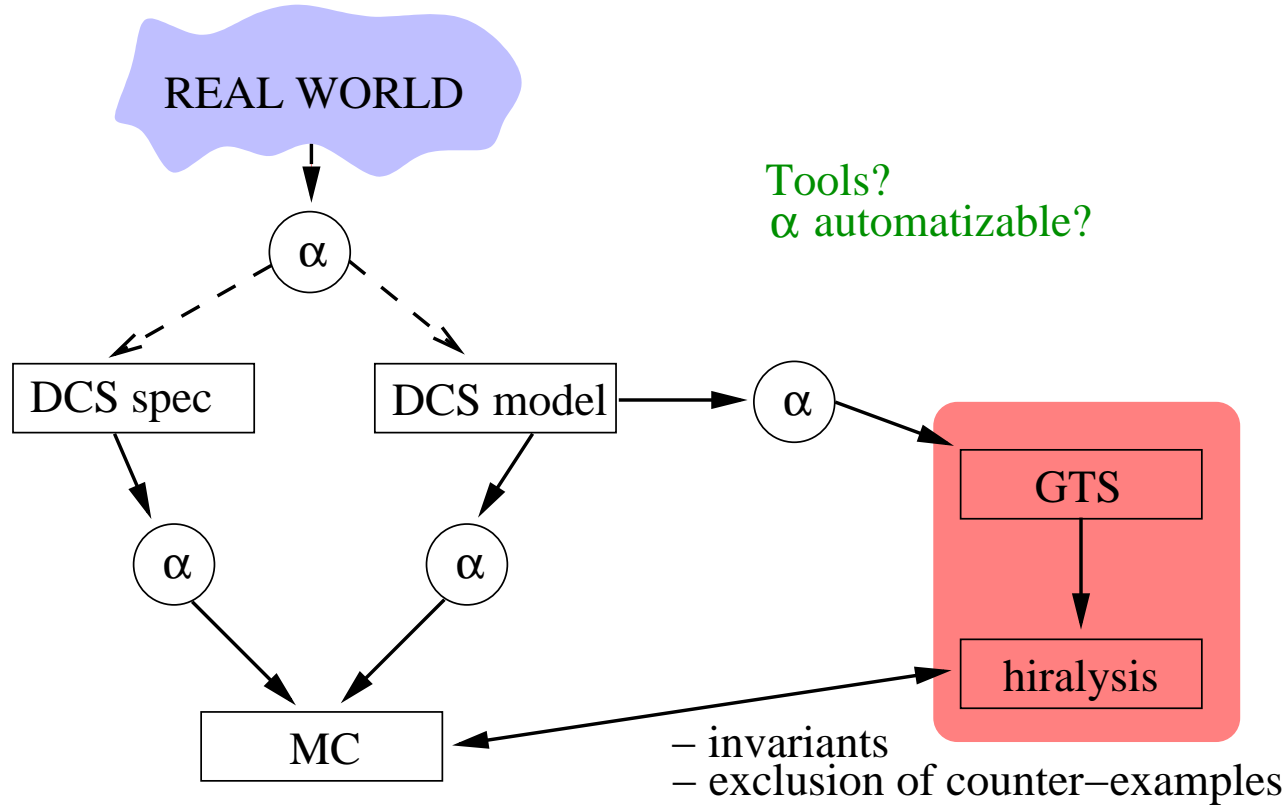
Layers of Abstraction

Reality	DCS	GTS
<ul style="list-style-type: none">● physical behavior● hybrid elements● faulty channels● distance sensors	<ul style="list-style-type: none">● queue based communication● non-determinism abstracts physical and hybrid behavior● reliable comm.	<ul style="list-style-type: none">● model only changes to communication topology● no queues● further non-determinism

DCS: Dynamic Communication Systems

GTS: Graph Transformation Systems

Big Picture



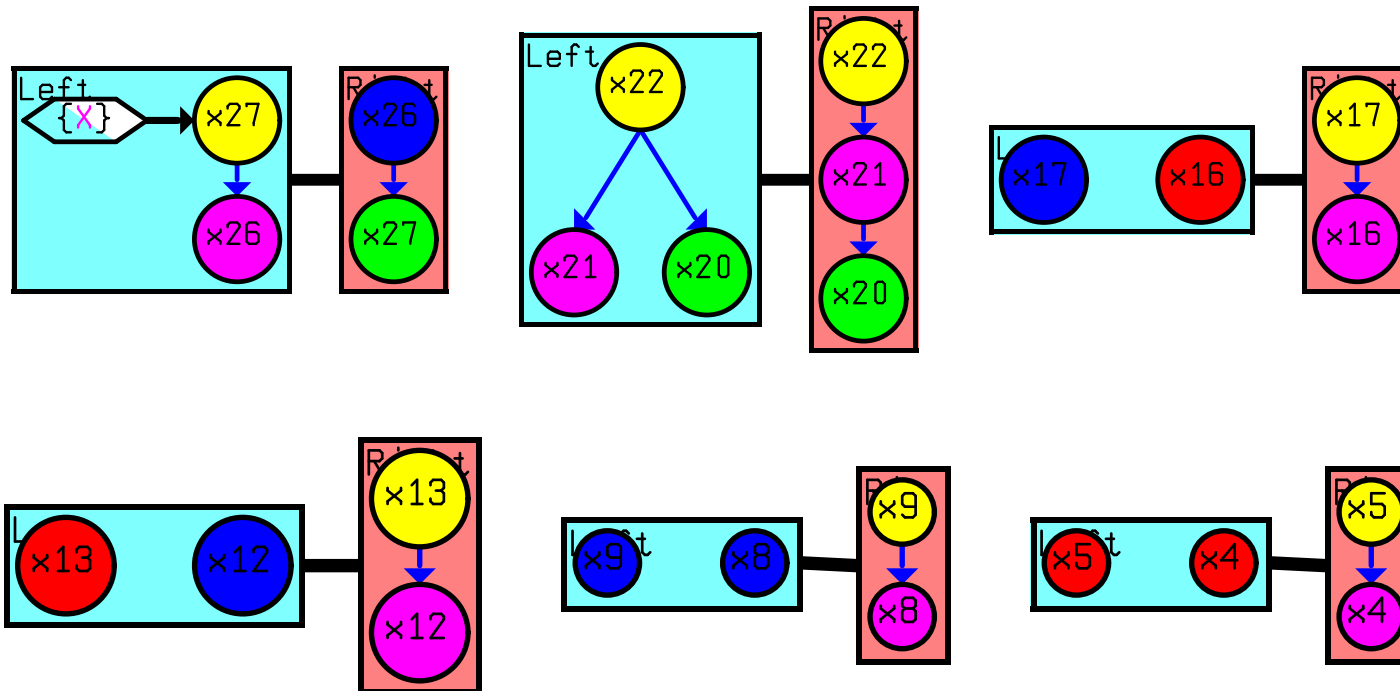
AVACS S2: Jörg Bauer, Werner Damm, Andreas Podelski, Ina Schaefer, Tobe Toben, Bernd Westphal, Thomas Wies, Reinhard Wilhelm

Overview

- Platoons as Graph Transformation
- Abstract Interpretation of GTS
- Properties

Platoons as Graph Transformation

- Platoon merge as GTS
- Labels: free agent, leader, follower, back leader, front leader



Graph Transformation Systems (GTS)

- Lab is a finite set of node labels, consider directed, node-labeled graphs
- A GTS is a set \mathcal{R} of rules
- A rule is a four-tuple (L, h, p, R) , where
 - L is the left graph
 - $h : V_L \rightarrow V_R$ is a mapping
 - $p : V_L \rightarrow \mathcal{P}(Lab)$ are partner constraints
 - R is the right graph
- (L, h, p, R) matches graph G , if L is a subgraph of G and partner constraints are satisfied
- Result of rule application: Replace L in G with R

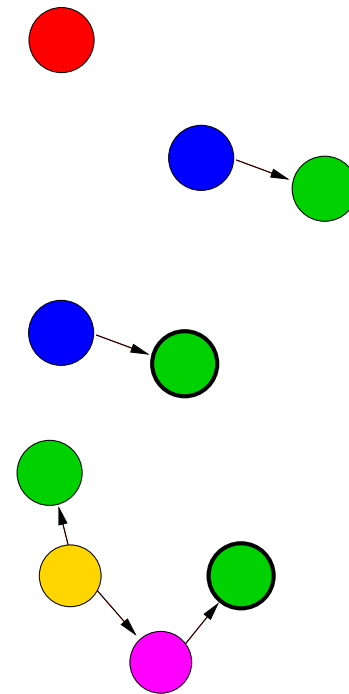
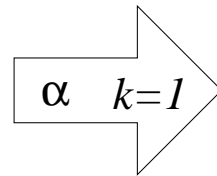
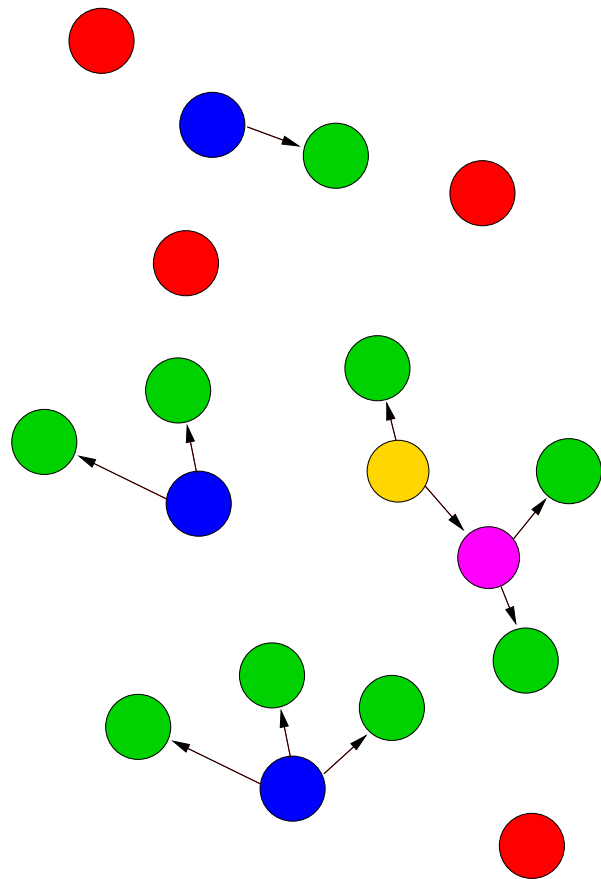
Overview

- Platoons as Graph Transformation
- Abstract Interpretation of GTS
 - Abstraction of Single Graphs
 - Abstract Transformers
- Properties

Abstraction of Single Graphs

- Per connected component:
 - Two nodes are **partner equivalent**, if they have **same label** and the sets of their **partners' labels** are equal
 - Abstraction: **Quotient graph** building yields **abstract clusters**
 - Keep **multiplicity information** up to some k
- Summarize **isomorphic abstract clusters**
- Size of abstraction is **bounded**

Example



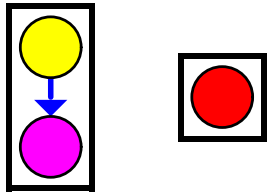
Overview

- Platoons as Graph Transformation
- Abstract Interpretation of GTS
 - Abstraction of Single Graphs
 - [Abstract Transformers](#)
- Properties

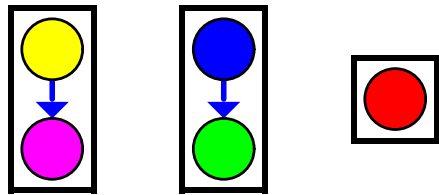
Abstract Updates

- **Non-injective** matches due to abstraction
- **Materialization** locally undoes abstraction: injective match
 - **Node** materialization
 - **Cluster** materialization
- Update concretely
- **Abstract** to guarantee boundedness

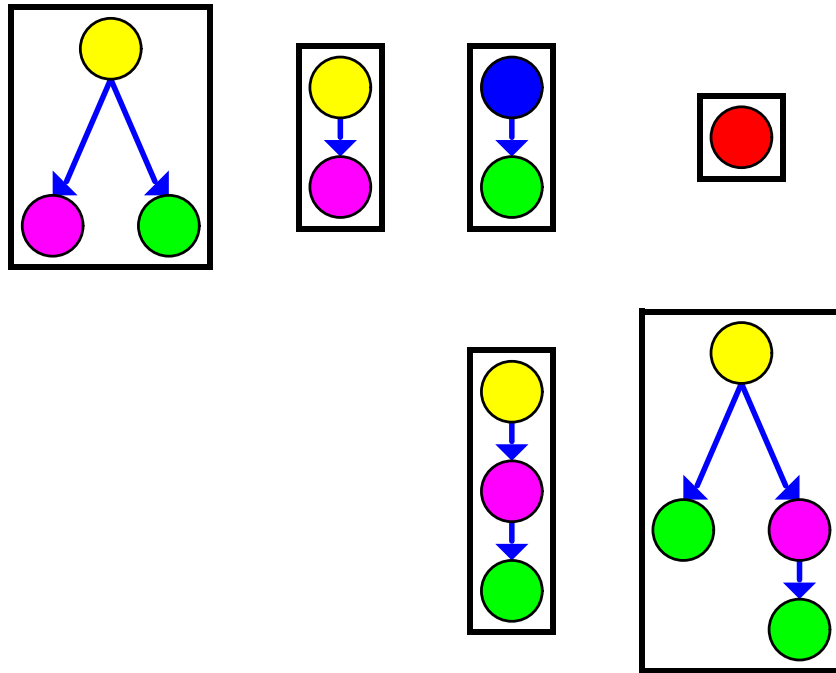
Example Computation: Iteration 0



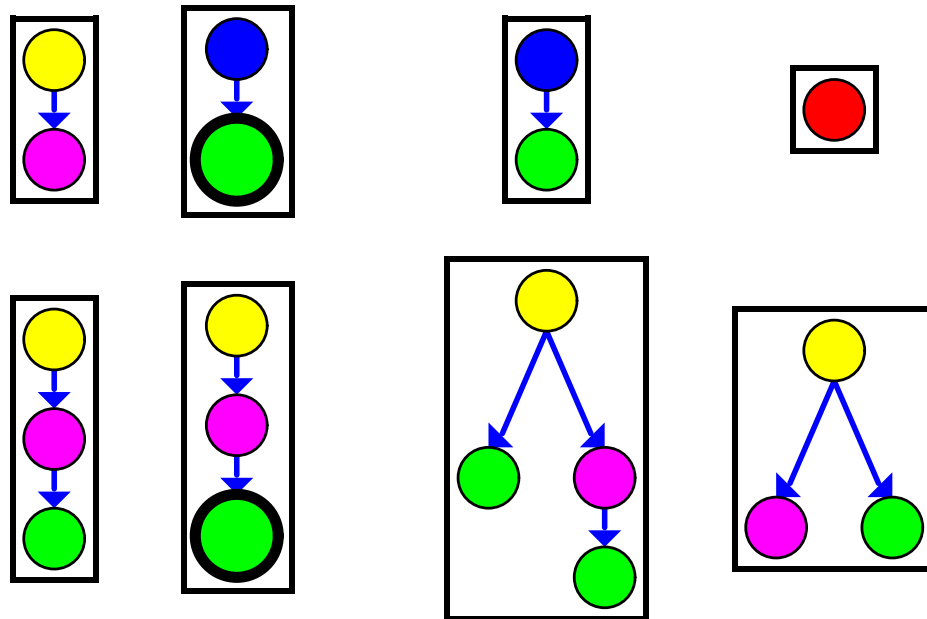
Example Computation: Iteration 1



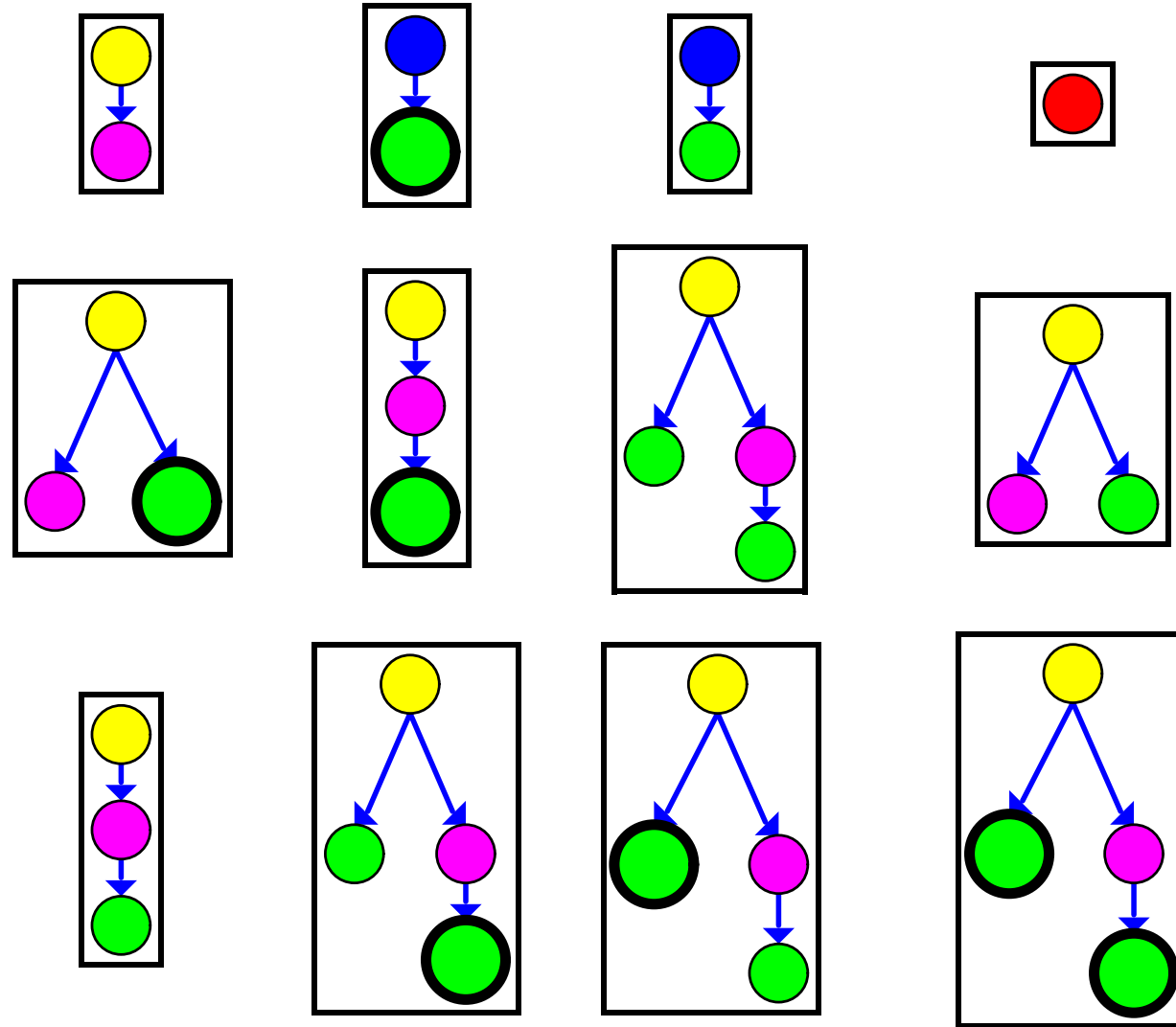
Example Computation: Iteration 2



Example Computation: Iteration 3



Example Computation: Iteration 4



Overview

- Platoons as Graph Transformation
- Abstract Interpretation of GTS
 - Abstraction of Single Graphs
 - Abstract Transformers
- Properties

Properties

- Boundedness
- Soundness
- Preservation of **existential positive** formulas
- For a given set of rules and graphs: **abstract matches** \Leftrightarrow **concrete matches**

\Rightarrow Statically checkable **completeness notions**:

- No **spurious clusters**
- **Combinatorial** completeness
- **Decidability** of the word problem

Conclusion

- GTS are **intuitive** and **formally defined**
 - AI-GTS is a **powerful** verification method for GTS
 - **Implementation**
 - Theoretical results: **property preservation** and **completeness**
-
- **Success story**: AI + MC (broader sense: **simulation and verification**)
 - Integrating **tool chain**
 - **Missing**: more links between **real world** and **simulation**

Related Work

- [Within AVACS](#)
 - embedded, real-time (Wilhelm)
 - hybrid systems (Hungar, Ratschan, Podelski, Wagner)
 - systems of systems (Podelski, Hermanns)
- [GTS: Handbook](#) (Rozenberg et.al.)
- [Verification of GTS](#): König, Rensink, Distefano
- [Big Picture Paper](#) at ACSD '06 [BTBW]

Benefits

For you

- GTS as modeling technique for simulation and verification
- generate invariants
- generate start conditions for simulation

For me

- exciting applications
- move (even) closer to the real world
- add probabilism
- π
- quantitative reasoning