

Real-Time Systems

Lecture 19: Wrapup

2013-07-16

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

- 19 - 2013-07-16 - main -

Content

Introduction

- **First-order Logic**
- **Duration Calculus (DC)**
- Semantical Correctness
- Proofs with DC
- DC Decidability
- DC Implementables
- ~~DC Proof Systems~~
- **PLC-Automata**
- **Timed Automata (TA)**, Uppaal
- Networks of Timed Automata
- Region/Zone-Abstraction
- Extended Timed Automata
- Undecidability Results (**TBA**)

$obs : \text{Time} \rightarrow \mathcal{D}(obs)$

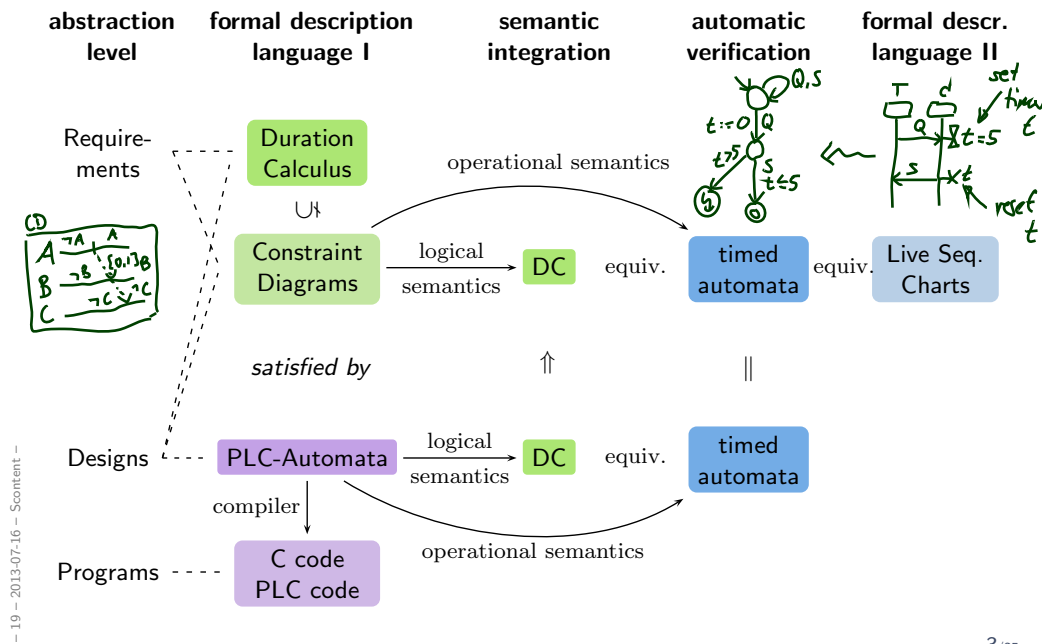
$\langle obs_0, \nu_0 \rangle, t_0 \xrightarrow{\lambda_0} \langle obs_1, \nu_1 \rangle, t_1 \dots$

- **Automatic Verification...**
- ...whether TA satisfies DC formula, observer-based

Recap

- 19 - 2013-07-16 - Content -

Tying It All Together



3/25

Lectures

- Lecture 01: Introduction
- Lecture 02: Timed Behaviour
- Lecture 03: Duration Calculus I (Symbols, State Assertions)
- Lecture 04: Duration Calculus II (Terms, Formulae)
- Lecture 05: Duration Calculus III (Abbreviations, Satisfy/Realise)
- Lecture 06: DC Properties I (RDC in Discrete Time)
- Lecture 07: DC Properties II (RDC in Continuous Time)
- Lecture 08: DC Implementables
- Lecture 09: PLC Automata
- Lecture 10: Timed Automata
- Lecture 11: Networks of Timed Automata
- Lecture 12: Location Reachability (or: The Region Automaton)
- Lecture 13: Zones
- Lecture 14: Extended Timed Automata
- Lecture 15: Timed Büchi Automata
- Lecture 16: The Universality Problem for TBA
- Lecture 17: Automatic Verification of DC Properties for TA I
- Lecture 18: Automatic Verification of DC Properties for TA II

- 19 - 2013-07-16 - SreCAP -

4/25

Lectures

- **Lecture 01: Introduction**
- Lecture 02: Timed Behaviour
- Lecture 03: Duration Calculus I (Symbols, State Assertions)
- Lecture 04: Duration Calculus II (Terms, Formulae)
- Lecture 05: Duration Calculus III (Abbreviations, Satisfy/Realise)
- Lecture 06: DC Properties I (RDC in Discrete Time)
- Lecture 07: DC Properties II (RDC in Continuous Time)
- Lecture 08: DC Implementables
- Lecture 09: PLC Automata
- Lecture 10: Timed Automata
- Lecture 11: Networks of Timed Automata
- Lecture 12: Location Reachability (or: The Region Automaton)
- Lecture 13: Zones
- Lecture 14: Extended Timed Automata
- Lecture 15: Timed Büchi Automata
- Lecture 16: The Universality Problem for TBA
- Lecture 17: Automatic Verification of DC Properties for TA I
- Lecture 18: Automatic Verification of DC Properties for TA II

Motivation/Big Picture

Lecture 1:

- What is a real-time system?
- In contrast to reactive, hybrid, ...?
- What is a safety-critical system?
- When do we call a real-time system correct?
- What is an approach to the development of correct real-time systems? What prerequisites does it have?
- What could justify this high effort?
- What are hard/soft deadlines?
- How did we partition reactive systems?
- Can you give an example for a "plant" from the tutorials.
- What's discrete and what's continuous time? Which did we use and why?

Lectures

- Lecture 01: Introduction
- **Lecture 02: Timed Behaviour**
- Lecture 03: Duration Calculus I (Symbols, State Assertions)
- Lecture 04: Duration Calculus II (Terms, Formulae)
- Lecture 05: Duration Calculus III (Abbreviations, Satisfy/Realise)
- Lecture 06: DC Properties I (RDC in Discrete Time)
- Lecture 07: DC Properties II (RDC in Continuous Time)
- Lecture 08: DC Implementables
- Lecture 09: PLC Automata
- Lecture 10: Timed Automata
- Lecture 11: Networks of Timed Automata
- Lecture 12: Location Reachability (or: The Region Automaton)
- Lecture 13: Zones
- Lecture 14: Extended Timed Automata
- Lecture 15: Timed Büchi Automata
- Lecture 16: The Universality Problem for TBA
- Lecture 17: Automatic Verification of DC Properties for TA I
- Lecture 18: Automatic Verification of DC Properties for TA II

Timed Behaviour

Lecture 02:

- **Educational Objectives:**
 - Get acquainted with one (simple but powerful) formal model of timed behaviour.
 - What is the idea of Time-dependent State Variables?
 - What is a timing diagram?
 - (Can you formalise this requirement using first order predicate-logic?)
 - What classes of timed properties did we distinguish?
 - To what classes of timed properties does this property belong?
 - Why is it useful to consider classes of properties?

Lectures

- Lecture 01: Introduction
- Lecture 02: Timed Behaviour
- **Lecture 03: Duration Calculus I (Symbols, State Assertions)**
- **Lecture 04: Duration Calculus II (Terms, Formulae)**
- **Lecture 05: Duration Calculus III (Abbreviations, Satisfy/Realise)**
- Lecture 06: DC Properties I (RDC in Discrete Time)
- Lecture 07: DC Properties II (RDC in Continuous Time)
- Lecture 08: DC Implementables
- Lecture 09: PLC Automata
- Lecture 10: Timed Automata
- Lecture 11: Networks of Timed Automata
- Lecture 12: Location Reachability (or: The Region Automaton)
- Lecture 13: Zones
- Lecture 14: Extended Timed Automata
- Lecture 15: Timed Büchi Automata
- Lecture 16: The Universality Problem for TBA
- Lecture 17: Automatic Verification of DC Properties for TA I
- Lecture 18: Automatic Verification of DC Properties for TA II

Duration Calculus

Lecture 03, 04, 05:

- **Educational Objectives:** Capabilities for following tasks/questions.
 - What does **this** Duration Calculus formula mean? (Intuitively and formally.)
 - Please formalise **this** requirement/design in DC. (In particular: get the syntax right.)
 - Why is DC called **duration** calculus? What's special about DC?
 - What's an interval logic?
 - What's the difference between global variables and state variables? What's their semantics?
 - Is a DC term a DC formula?
 - What's a rigid term?
 - What does **this** DC abbreviation "unfold" to?
 - There was the question whether the DC semantics is well-defined. What was the issue and how did we address it?
 - Please give an interpretation of the state variable which satisfies/realises (from 0) **this** DC formula.

Lectures

- Lecture 01: Introduction
- Lecture 02: Timed Behaviour
- Lecture 03: Duration Calculus I (Symbols, State Assertions)
- Lecture 04: Duration Calculus II (Terms, Formulae)
- Lecture 05: Duration Calculus III (Abbreviations, Satisfy/Realise)
- **Lecture 06: DC Properties I (RDC in Discrete Time)**
- **Lecture 07: DC Properties II (RDC in Continuous Time)**
- Lecture 08: DC Implementables
- Lecture 09: PLC Automata
- Lecture 10: Timed Automata
- Lecture 11: Networks of Timed Automata
- Lecture 12: Location Reachability (or: The Region Automaton)
- Lecture 13: Zones
- Lecture 14: Extended Timed Automata
- Lecture 15: Timed Büchi Automata
- Lecture 16: The Universality Problem for TBA
- Lecture 17: Automatic Verification of DC Properties for TA I
- Lecture 18: Automatic Verification of DC Properties for TA II

DC Properties

Lecture 06 & 07:

- **Educational Objectives:** Capabilities for following tasks/questions.
 - Facts: decidability properties. What is/is not decidable for (R)DC?
 - Why would a decision procedure for **this** problem be useful?
 - How is (un)decidability of the **hmm** problem proved? (What's the idea of the proof? What steps are conducted? What is established?)
 - What's RDC? What is it useful for?
 - What's (R)DC in discrete time?
 - Can we distinguish by a DC formula whether we're in a discrete or continuous time model?

Lectures

- Lecture 01: Introduction
- Lecture 02: Timed Behaviour
- Lecture 03: Duration Calculus I (Symbols, State Assertions)
- Lecture 04: Duration Calculus II (Terms, Formulae)
- Lecture 05: Duration Calculus III (Abbreviations, Satisfy/Realise)
- Lecture 06: DC Properties I (RDC in Discrete Time)
- Lecture 07: DC Properties II (RDC in Continuous Time)
- **Lecture 08: DC Implementables**
- **Lecture 09: PLC Automata**
- Lecture 10: Timed Automata
- Lecture 11: Networks of Timed Automata
- Lecture 12: Location Reachability (or: The Region Automaton)
- Lecture 13: Zones
- Lecture 14: Extended Timed Automata
- Lecture 15: Timed Büchi Automata
- Lecture 16: The Universality Problem for TBA
- Lecture 17: Automatic Verification of DC Properties for TA I
- Lecture 18: Automatic Verification of DC Properties for TA II

DC Implementables

Lecture 08:

- **Educational Objectives:** Capabilities for following tasks/questions.
 - What does **this** standard form mean? Give a satisfying interpretation.
 - What is a control automaton?
 - What's a basic phase of a control automaton?
 - What are implementables?
 - Please specify (and prove correct) a controller which satisfies **this** requirement.
 - Do you like gas burners?
 - What property of implementables is interesting in the context of TA?

Lecture 09:

- **Educational Objectives:** Capabilities for following tasks/questions.
 - What is the “philosophy” of PLC? What did we generalise/abstract them to?
 - Why did we discuss PLC?
 - What if we don’t have a PLC at hand but only a real-time Linux and a C compiler?
 - What would distinguish a real-time from a plain Linux anyway?
 - What is a PLC automaton?
 - What’s the issue with the cycle time in a PLCA?
 - What does **this** PLC automaton do?
 - How would you solve **this** control problem with a PLCA?
 - How does the proposed approach work, from requirements to a correct implementation with DC?
 - *DC- semantics of DC*

Lectures

- Lecture 01: Introduction
- Lecture 02: Timed Behaviour
- Lecture 03: Duration Calculus I (Symbols, State Assertions)
- Lecture 04: Duration Calculus II (Terms, Formulae)
- Lecture 05: Duration Calculus III (Abbreviations, Satisfy/Realise)
- Lecture 06: DC Properties I (RDC in Discrete Time)
- Lecture 07: DC Properties II (RDC in Continuous Time)
- Lecture 08: DC Implementables
- Lecture 09: PLC Automata
- **Lecture 10: Timed Automata**
- **Lecture 11: Networks of Timed Automata**
- Lecture 12: Location Reachability (or: The Region Automaton)
- Lecture 13: Zones
- **Lecture 14: Extended Timed Automata**
- Lecture 15: Timed Büchi Automata
- Lecture 16: The Universality Problem for TBA
- Lecture 17: Automatic Verification of DC Properties for TA I
- Lecture 18: Automatic Verification of DC Properties for TA II

Timed Automata

Lecture 10, 11 & 14:

- **Educational Objectives:** Capabilities for following tasks/questions.
 - What's notable about TA syntax? What's a simple clock constraint?
 - What's a configuration of a TA? When are two in transition relation?
 - Is there something remarkable about the definition of configurations?
 - What's the difference between guard and invariant? Why have both?
 - What's a computation path? A run? Zeno behaviour? Timelock?
 - Does **this** TA have a run? Which/why not?
 - Where does "time pass"?
 - Can you imagine what somebody means by saying "TA are closed under parallel composition"?
 - ~~In how far are Uppaal TA non-compositional?~~
 - What's an urgent/committed location? What's the difference?
 - Is **this** location of **that** TA network reachable?
 - Where has the notion of "input action" and "output action" a correspondence in the formal semantics?
 - Can you give a network of TA which has **this** behaviour?

Lectures

- Lecture 01: Introduction
- Lecture 02: Timed Behaviour
- Lecture 03: Duration Calculus I (Symbols, State Assertions)
- Lecture 04: Duration Calculus II (Terms, Formulae)
- Lecture 05: Duration Calculus III (Abbreviations, Satisfy/Realise)
- Lecture 06: DC Properties I (RDC in Discrete Time)
- Lecture 07: DC Properties II (RDC in Continuous Time)
- Lecture 08: DC Implementables
- Lecture 09: PLC Automata
- Lecture 10: Timed Automata
- Lecture 11: Networks of Timed Automata
- **Lecture 12: Location Reachability (or: The Region Automaton)**
- **Lecture 13: Zones**
- Lecture 14: Extended Timed Automata
- Lecture 15: Timed Büchi Automata
- Lecture 16: The Universality Problem for TBA
- Lecture 17: Automatic Verification of DC Properties for TA I
- Lecture 18: Automatic Verification of DC Properties for TA II

Regions and Zones

Lecture 12:

- **Educational Objectives:** Capabilities for following tasks/questions.
 - What are decidable problems of TA?
 - How can we show this? What are the essential premises of decidability?
 - What is a region? What is the region automaton of this TA?
 - What's the time abstract system of a TA? Why did we consider this?
 - What can you say about the complexity of Region-automaton based reachability analysis?

Lecture 13:

- **Educational Objectives:** Capabilities for following tasks/questions.
 - What's a zone? In contrast to a region?
 - Motivation for having zones?
 - What's a DBM? Who needs to know DBMs?

Lectures

- Lecture 01: Introduction
- Lecture 02: Timed Behaviour
- Lecture 03: Duration Calculus I (Symbols, State Assertions)
- Lecture 04: Duration Calculus II (Terms, Formulae)
- Lecture 05: Duration Calculus III (Abbreviations, Satisfy/Realise)
- Lecture 06: DC Properties I (RDC in Discrete Time)
- Lecture 07: DC Properties II (RDC in Continuous Time)
- Lecture 08: DC Implementables
- Lecture 09: PLC Automata
- Lecture 10: Timed Automata
- Lecture 11: Networks of Timed Automata
- Lecture 12: Location Reachability (or: The Region Automaton)
- Lecture 13: Zones
- Lecture 14: Extended Timed Automata
- **Lecture 15: Timed Büchi Automata**
- **Lecture 16: The Universality Problem for TBA**
- Lecture 17: Automatic Verification of DC Properties for TA I
- Lecture 18: Automatic Verification of DC Properties for TA II

Undecidable TA Problems

Lecture 15 & 16:

- **Educational Objectives:** Capabilities for following tasks/questions.
 - What's a TBA and what's the difference to (extended) TA?
 - What is a timed (regular) language?
 - What language does **this** TBA accept?
 - Can you give a TBA with **this** language?
 - What's undecidable for timed (Büchi) automata?
 - Why is this unfortunate?
 - What's the idea of the proof?
 - What's the universality problem?

Lectures

- Lecture 01: Introduction
- Lecture 02: Timed Behaviour
- Lecture 03: Duration Calculus I (Symbols, State Assertions)
- Lecture 04: Duration Calculus II (Terms, Formulae)
- Lecture 05: Duration Calculus III (Abbreviations, Satisfy/Realise)
- Lecture 06: DC Properties I (RDC in Discrete Time)
- Lecture 07: DC Properties II (RDC in Continuous Time)
- Lecture 08: DC Implementables
- Lecture 09: PLC Automata
- Lecture 10: Timed Automata
- Lecture 11: Networks of Timed Automata
- Lecture 12: Location Reachability (or: The Region Automaton)
- Lecture 13: Zones
- Lecture 14: Extended Timed Automata
- Lecture 15: Timed Büchi Automata
- Lecture 16: The Universality Problem for TBA
- **Lecture 17: Automatic Verification of DC Properties for TA I**
- **Lecture 18: Automatic Verification of DC Properties for TA II**

Automatic Verification of DC Properties for TA

Lecture 17 & 18:

- **Educational Objectives:** Capabilities for following tasks/questions.
 - How can we relate TA and DC formulae?
 - What's a bit tricky about that (regarding semantics and intuition)?
 - Can we use Uppaal to check whether **this** TA satisfies **this** DC formula?
 - How? What do we have to be careful with?
 - What is a testable DC formula?
 - What could **this** monitor/observer/test automaton be useful for?
 - Can the TA and DC formulae for which we can check something be (syntactically) characterised?

References

References

[Olderog and Dierks, 2008] Olderog, E.-R. and Dierks, H. (2008). *Real-Time Systems - Formal Specification and Automatic Verification*. Cambridge University Press.