Temporal Logics

http://d3s.mff.cuni.cz

Pavel Parízek

Department of Distributed and Dependable Systems

CHARLES UNIVERSITY IN PRAGUE
faculty of mathematics and physics
Modal logic

- Possibly
  - $<> P$

- Necessarily
  - $[] P$
Dynamic logic

• Formulas
  - $<a> P$
  - $[a] P$

• Special actions
  - Constant: $[1] P$
  - Block: $[0] P$
Temporal logic

- Variants: LTL, CTL, ...

- Operators
  - Globally: G p
  - Eventually: F p
  - Next step: X p

- Details: course NSWI101
TLA: Temporal Logic of Actions

- TLA+ specification language
  - Low-level language based on logic and sets
  - Enables users to define a transition system
- PlusCal algorithm language
  - Syntax much closer to C/C#/Java
  - Writing and testing pseudo-code

- Home page

- TLA Toolbox (IDE)
PlusCal

- **Features**
  - control-flow statements, non-determinism, simple identification of atomic steps (for concurrency), procedure call and return

- **Example algorithms**
  - Euclid’s GCD, mutual exclusion, alternating bit

- **Translation into TLA+ specification**

- **Analyzing with TLC model checker**

- **Further reading**
TLA+

- Features
  - variables, constants, arithmetic
  - common set and logic operators
  - functions, control statements
  - sequences, tuples, arrays, records
  - non-deterministic choice
  - basic temporal operators

- Example translation: Euclid’s GCD

- Further reading
  - L. Lamport. *Euclid Writes an Algorithm: A Fairytale*
Advanced topics

• Liveness (termination)
• Fairness (scheduling)

• See the course NSWI101
  - [http://d3s.mff.cuni.cz/teaching/system_behaviour_models/](http://d3s.mff.cuni.cz/teaching/system_behaviour_models/)
• Analyzing distributed concurrent algorithms, protocols and systems

• Case study: Amazon
  - http://doi.acm.org/10.1145/2699417