

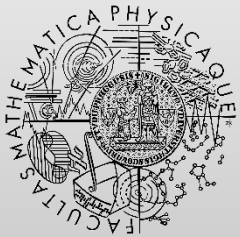
Temporal Logics

<http://d3s.mff.cuni.cz>

Department of
Distributed and
Dependable
Systems



Pavel Parízek



FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

Modal logic

- Possibly

- $\langle \rangle P$

- Necessarily

- $[] P$

Dynamic logic

- Formulas
 - $\langle a \rangle P$
 - $[a] P$
- 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
 - <http://lamport.azurewebsites.net/tla/tla.html>
- TLA Toolbox (IDE)
 - <http://lamport.azurewebsites.net/tla/tools.html>

- 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
 - L. Lamport. **The PlusCal Algorithmic Language**. ICTAC 2009
 - <http://lamport.azurewebsites.net/pubs/pubs.html#pluscal>

- 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**
 - <http://lamport.azurewebsites.net/pubs/pubs.html#euclid>

Advanced topics

- Liveness (termination)
- Fairness (scheduling)

- See the course NSWI101
 - <http://d3s.mff.cuni.cz/teaching/nswi101>

Practice

- Analyzing distributed concurrent algorithms, protocols and systems
- Case study: Amazon
 - C. Newcombe, T. Rath, F. Zhang, B. Munteanu, M. Brooker, and M. Deardeuff. **How Amazon Web Services Uses Formal Methods**. Communications of the ACM, 58(4), April 2015
 - <http://doi.acm.org/10.1145/2699417>