The TLA+ Toolbox

Celebrating its 10th Anniversary

Markus Alexander Kuppe

Microsoft

October 7, 2019

Outline

TLA$^+$ & PlusCal

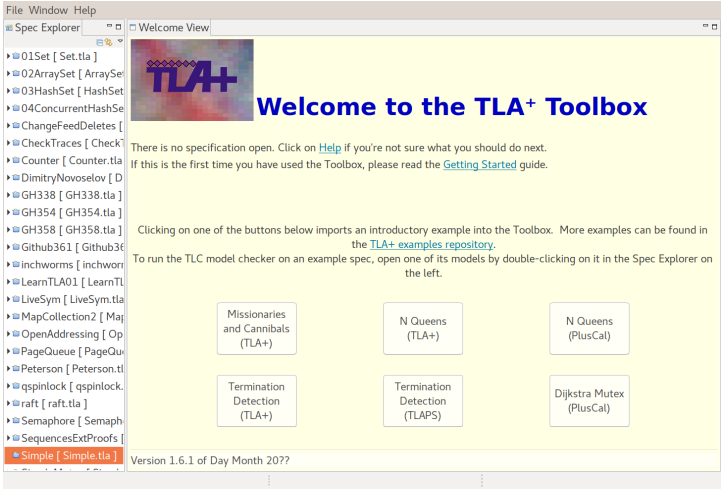
- TLA$^+$ [Lamport, 1994]
  - High-level specification language
    - Design above the code level
  - Math-based, Untyped
  - Linear-time framework: Temporal Logic of Actions

- PlusCal [Lamport, 2009a]
  - "*A gateway drug for programmers*" (C. Newcombe)
  - (Imperative-style) pseudo-code with precise semantics
  - Transpiles to TLA$^+$
    - $\Rightarrow$ Verifiable with Tools

TLA$^+$ Tools

- PlusCal transpiler, SANY, Pretty-Printer, ...
- TLC [Yu et al., 1999]
  - Explicit state model-checker
  - Addresses state space explosion by scaling (safety) checking
- TLAPS [Chaudhuri et al., 2008]
  - Unbounded Domains
  - Proof Manager on top of
    - CVC3, Zenon, Isabelle, ...
- Apalache [Konnov et al., 2019]
  - Symbolic model-checker (z3)
  - Supplement or successor to TLC?!

Demo[1]



[1] Recordings at
https://www.youtube.com/playlist?list=PLWLcqZLzY8u8-g47T_zHiK2zvPpIohRsA

Architecture

- Toolbox implemented on the Eclipse Rich Client Platform [McAffer et al., 2010]
  - Platform-independent[2]
  - Easy deployment (bundles Java & TLC)
- Eclipse "opinionated" framework
  - Eclipse is open source => Do not work around limitations
- Early years of Toolbox development too little emphasis on testing
  - UI testing remains hard
  - => Still paying the price with every new feature
- Eclipse was the right choice 10 years ago

[2] "Write once, test everywhere"

Back-ends

- Back-end integration:
  - Nested process
  - Low-level, text-based "API"
    - Custom input/output parser
- Flexible, can accommodate any back-end
- Schema-less challenge to evolve
- Alternative: IPC (structured data)

Conclusion

- Toolbox one-stop solution for:
  - Model-Checking with TLC
    - Profiler to find and remove performance bottlenecks and spec errors
    - CloudTLC for large-scale model checking & parallelized design space exploration
  - Interactive Theorem Proving with TLAPS
    - MC aids in finding inductive invariants
    - Proof decomposition
- $\sim 20k$ downloads in ~ 12 months indicate usefulness

Outlook

- Integrate Trace Exploration with third-party tools [Schultz, 2018]
- Study usefulness of Profiler on real-world specs
- Toolbox workflow for probabilistic validation of inductive invariants
- Integrate Apalache as additional back-end into Toolbox
  - TLA$^+$ type inference engine for Toolbox and back-ends
- Toolbox 2.0 as "cloud native" application
  - Lets specify a *Verification Server Protocol* (inspired by LSP)?!

Outlook

- Integrate Trace Exploration with third-party tools [Schultz, 2018]
- Study usefulness of Profiler on real-world specs
- Toolbox workflow for probabilistic validation of inductive invariants
- Integrate Apalache as additional back-end into Toolbox
  - TLA$^+$ type inference engine for Toolbox and back-ends
- Toolbox 2.0 as “cloud native” application
  - Lets specify a *Verification Server Protocol* (inspired by LSP)?!

Q&A

(TLA$^+$ @ Tools Exhibit)

Bibliography I

Kaustuv C. Chaudhuri, Damien Doligez, Leslie Lamport, and Stephan Merz. A TLA+ Proof System. *arXiv:0811.1914 [cs]*, November 2008. URL `http://arxiv.org/abs/0811.1914`.

Igor Konnov, Jure Kukovec, and Thanh Hai Tran. TLA+ model checking made symbolic. 2019. URL `https://2019.splashcon.org/details/splash-2019-oopsla/7/TLA-model-checking-made-symbolic`.

Leslie Lamport. The Temporal Logic of Actions. *ACM Transactions on Programming Languages and Systems*, 16(3):872–923, May 1994. ISSN 01640925. doi: 10.1145/177492.177726. URL `http://portal.acm.org/citation.cfm?doid=177492.177726`.

Bibliography II

Leslie Lamport. The PlusCal Algorithm Language. In Martin Leucker and Carroll Morgan, editors, *Theoretical Aspects of Computing - ICTAC 2009*, volume 5684, pages 36–60. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009a. ISBN 978-3-642-03465-7 978-3-642-03466-4. URL http://link.springer.com/10.1007/978-3-642-03466-4_2.

Leslie Lamport. Teaching concurrency. *ACM SIGACT News*, 40(1):58, February 2009b. ISSN 01635700. doi: 10.1145/1515698.1515713. URL http://portal.acm.org/citation.cfm?doid=1515698.1515713.

Jeff McAffer, Jean-Michel Lemieux, and Chris Aniszczyk. *Eclipse Rich Client Platform*. The Eclipse Series. Addison-Wesley, Upper Saddle River, NJ, 2nd ed edition, 2010. ISBN 978-0-321-60378-4. OCLC: ocn262433527.

William Schultz. An Animation Module for TLA+, 2018. URL https://easychair.org/smart-slide/slide/8V76#.

Bibliography III

Yuan Yu, Panagiotis Manolios, and Leslie Lamport. Model Checking TLA+ Specifications. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Laurence Pierre, and Thomas Kropf, editors, *Correct Hardware Design and Verification Methods*, volume 1703, pages 54–66. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999. ISBN 978-3-540-66559-5 978-3-540-48153-9. URL `http://link.springer.com/10.1007/3-540-48153-2_6`.

TLA+

MODULE M

VARIABLE v

$Init \triangleq \ldots$ Defines initial states

$Next \triangleq v' = v + 1 \land \ldots$ Constraints allowed transitions

$Spec \triangleq Init \land \Box[Next]_v$ Defines system executions

$\land F$ and optionally Fairness

$Safety \triangleq \Box \ldots$

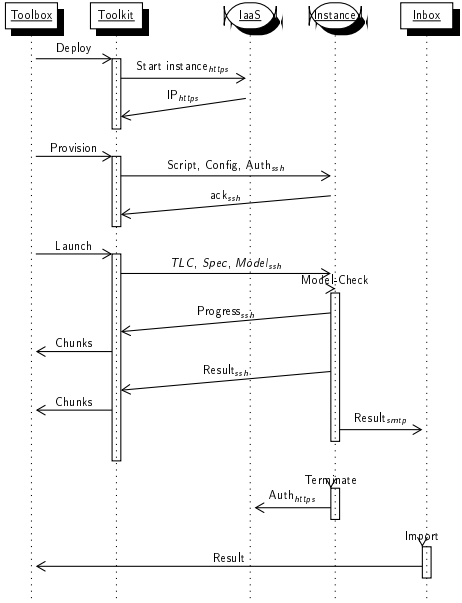
$Liveness \triangleq \Diamond\Box \ldots$

PlusCal

```
--algorithm Euclid{
  variables x = M, y = N ; {
    while (x ≠ y){
      if (x < y){y := y − x}
      else       {x := x − y}
    }
  }
}
```

Sequential algorithms need no labels (atomicy via labels)

CloudTLC


Toolbox
Toolkit
IaaS
Instance
Inbox
Deploy
Start instance$_{https}$
IP$_{https}$
Provision
Script, Config, Auth$_{ssh}$
ack$_{ssh}$
Launch
TLC, Spec, Model$_{ssh}$
Model-Check
Progress$_{ssh}$
Chunks
Result$_{ssh}$
Chunks
Result$_{smtp}$
Terminate
Auth$_{https}$
Import
Result