Practical Course: SMT Solving
Introductory Meeting

Erika Ábrahám, Florian Corzilius, Gereon Kremer

Theory of Hybrid Systems
Informatik 2

WS 2014/2015
Goals of this practical course

- Understanding of SMT solving
- Understanding of theories: QF_UF, QF_NRA, QF_UFNRA, ...
- Understanding of different decision procedures for equality logic and uninterpreted functions
Goals of this practical course

- Understanding of SMT solving
- Understanding of theories: QF_UF, QF_NRA, QF_UFNRA, ...
- Understanding of different decision procedures for equality logic and uninterpreted functions
- Implementation of these procedures as theory modules in SMT-RAT
- Implementation in clean and modern C++
- Debugging, evaluation and documentation of theory modules
- Presentation of results
We have two teams ($X \in \{a, b\}$)

- A mailinglist smt-X@ths.informatik.rwth-aachen.de
- Read access to CArL and SMT-RAT repositories
- A git repository containing a clone of SMT-RAT:
  
  https://srv-i2.informatik.rwth-aachen.de:8443/git/smt-X.git
We have two teams ($X \in \{a, b\}$)

- A mailinglist smt-X@ths.informatik.rwth-aachen.de
- Read access to CArL and SMT-RAT repositories
- A git repository containing a clone of SMT-RAT: https://srv-i2.informatik.rwth-aachen.de:8443/git/smt-X.git
- Access to our cluster: Direct or indirect?
- Anything else? Trac? Wiki?
Setup

We have two teams ($X \in \{a, b\}$)

- A mailinglist smt-X@ths.informatik.rwth-aachen.de
- Read access to CArL and SMT-RAT repositories
- A git repository containing a clone of SMT-RAT:
  https://srv-i2.informatik.rwth-aachen.de:8443/git/smt-X.git
- Access to our cluster: Direct or indirect?
- Anything else? Trac? Wiki?
- You need: Linux or MacOS with the following software:
  git, cmake, ccmake, cln, gmp, eigen3, g++ ($\geq 4.8$) or clang ($\geq 3.4$), boost, doxygen, gtest
We have two teams ($X \in \{a, b\}$)

- A mailinglist smt-X@ths.informatik.rwth-aachen.de
- Read access to CArL and SMT-RAT repositories
- A git repository containing a clone of SMT-RAT:
  https://srv-i2.informatik.rwth-aachen.de:8443/git/smt-X.git
- Access to our cluster: Direct or indirect?
- Anything else? Trac? Wiki?
- You need: Linux or MacOS with the following software:
  `git`, `cmake`, `ccmake`, `cln`, `gmp`, `eigen3`, `g++ (\geq 4.8)` or `clang (\geq 3.4)`, `boost`, `doxygen`, `gtest`

Changes to CArL or the core of SMT-RAT will be committed by us and available to both teams
Roadmap

- Implement theory module for equality logic
- Extend theory module for uninterpreted functions
- Implement DTC module
- Implement Ackermann module
- Compare approaches on standard benchmarks
- Presentation of results: February 2015
Building groups

<table>
<thead>
<tr>
<th>Team A</th>
<th>Team B</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Meetings

Weekly:

- Meeting in the seminar room
- Not mandatory, but encouraged
- You can discuss, ask for help, work/implement, ...
Meetings

Weekly:
- Meeting in the seminar room
- Not mandatory, but encouraged
- You can discuss, ask for help, work/implement, ...

Monthly (every fourth meeting):
- Progress report
- Mandatory
- Each group gives a short presentation
- What works? What doesn’t? Current problems?
URLs

- **Homepage:**
  http://ths.rwth-aachen.de/teaching/ws-14/praktikum-smt-solving/

- **Supervisors:**  smt-orga@ths.informatik.rwth-aachen.de

- **Everyone:**  smt@ths.informatik.rwth-aachen.de

- **Your team:**  smt-X@ths.informatik.rwth-aachen.de

- **CArL:**
  https://<user>@srv-i2.informatik.rwth-aachen.de:8443/git/carl.git

- **SMT-RAT:**
  https://<user>@srv-i2.informatik.rwth-aachen.de:8443/git/smtrat.git

- **Your git:**
  https://<user>@srv-i2.informatik.rwth-aachen.de:8443/git/smt-X.git

- **Documentation for CArL (includes introduction to our build process):**
  http://ths.informatik.rwth-aachen.de/doxygen/carl/html/
Roberto Bruttomesso, Alessandro Cimatti, Anders Franzén, Alberto Griggio, Alessandro Santuari, and Roberto Sebastiani.
To Ackermann-ize or not to Ackermann-ize? On Efficiently Handling Uninterpreted Function Symbols in SMT (EUF $\cup T$).

Florian Corzilius, Ulrich Loup, Sebastian Junges, and Erika Ábrahám.
SMT-RAT: An SMT-Compliant Nonlinear Real Arithmetic Toolbox.

Daniel Kroening and Ofer Strichman.
That’s it...

Questions?