SWORD v1.0

Jean Christoph Jung
André Sülflow
Robert Wille
Rolf Drechsler

University of Bremen
jeanjung@informatik.uni-bremen.de
SWORD SMT-COMP 2008

- DPLL-solver
- Extended by module concept
- Without any preprocessing

Modules:
- Sub-units with functionality for propagation and decision heuristics
- E.g. ADD, MUL, and DIV
Lessons Learned

- Module concept yields speed-up compared to pure bit blasting
- Preprocessing is very important
- Some instances can be “solved” using term rewriting and unit propagation only
SWORD Architecture

Word-level Rewriting

Bit-level Rewriting

Solve Engine
SWORD Architecture

\[ x + 0 \rightarrow x \]
\[ a \cdot x + b \cdot x \rightarrow (a + b) \cdot x \]
\[ a + (x + b) \rightarrow (a + b) + x \]

- Word-level Rewriting
- Bit-level Rewriting
- Solve Engine
**SWORD Architecture**

\[
\begin{align*}
x + 0 & \leadsto x \\
a \cdot x + b \cdot x & \leadsto (a + b) \cdot x \\
a + (x + b) & \leadsto (a + b) + x
\end{align*}
\]

**Word-level Rewriting**

\[
\begin{align*}
a \land \neg a & \leadsto false \\
\neg(a \land b) \land \neg a & \leadsto \neg a \\
a \leftrightarrow (a \land b) & \leadsto a \rightarrow b
\end{align*}
\]

**Bit-level Rewriting**

**Solve Engine**
SWORD Architecture

Word-level Rewriting

\[ x + 0 \leadsto x \]
\[ a \cdot x + b \cdot x \leadsto (a + b) \cdot x \]
\[ a + (x + b) \leadsto (a + b) + x \]

Bit-level Rewriting

\[ a \land \neg a \leadsto false \]
\[ \neg(a \land b) \land \neg a \leadsto \neg a \]
\[ a \leftrightarrow (a \land b) \leadsto a \rightarrow b \]

Solve Engine

SAT

free var? yes
select var (solver)
select var (module)
propagation
ok fail
resolve conflict fail UNSAT
Current + Future Work

- Study new possibilities given by modules
- API + library

- Support of QF_AUFBV?