Variability Management

Prof. Dr. Christoph Weidenbach
Automation of Logic

Application

Logic

Calculus

Implementation

Automatic Computation of Properties
Examples by PhD Thesis

- Dr. Matthias Horbach: second-order logic decidability
- Dr. Carsten Ihlemann: local theory extensions
- Tinxiang Lu: verifying correctness of PASTRY
- Arnaud Fietzke: combining first-order and prob. reasoning
- Patrick Wischnewski: reasoning in large ontologies
- Ching Hoo Tang: variability management for steel factories (Siemens)
Reasoning in Large Ontologies

Develop “semantic” GOOGLE:

SPASS YAGO
Configuration Today

The car industry:

Opel Configuration
Today’s Architecture

- Sales Views
- Logistics Views
- Engineering Views
- … Views

- Sales Software
- MM Software
- CAX Software
- …

Product

Application
Application + DB
Paper, People
Future’s Architecture

Sales Views

Logistics Views

Engineering Views

XXX Views

Product Interface

Product Specification

Product Reasoning

DB

Application
Scientific View

Sales Views

Logistics Views

Engineering Views

XXX Views

Reasoning Interface

Logic Formulas

Theorem Provers

DB

Application
Concrete Example

v.control

v.Control SPASS Interface

Propositional Logic

SPASS

DB

Application

Application

Application
V. CONTROL & SPASS

Logic for Business

in cooperation with
Prof. Dr. Georg Rock, Uni App Sc Trier, PROSTEP IMP
Karsten Theis, PROSTEP IMP
Patrick Wischewski, MPI-INF
Propositional Logic

- Language: propositional variables can be true (1) or false (0)
- Connectives: $\Rightarrow$ implication, $\neg$ negation, $\lor$ disjunction, $\land$ conjunction
- Clause: disjunction of variables or their negations (literal)
- Validity: a formula is valid iff it is true for all possible assignments
- Assignment: setting all propositional variables 1 or 0, can also be expressed by showing the true literals
- we write $M \models C$ if the clause $C$ is true by assignment $M$
- SAT: propositional satisfiability, find an assignment such that for a set of clauses all clauses are valid in the assignment
Unit Propagation

$\text{UProp}(N, M)$

while (there is a clause $C' \lor L \in N$ such that

$M \models \neg C'$ and $L \notin M$ and $\neg L \notin M$)

$M := M \cup \{L\}$;

return $M$;

$\text{UProp}([-A \lor \neg B \lor E, \quad \neg A \lor B, \quad \neg E, \quad D, \quad A], \emptyset)$

$\rightarrow M = \emptyset$

$\rightarrow M = \{\neg E\}$

$\rightarrow M = \{\neg E, D\}$

$\rightarrow M = \{\neg E, D, A\}$

$\rightarrow M = \{\neg E, D, A, B\}$
DPLL Procedure

DPLL(\(N, M\))
if for all \(C \in N\) we have \(M \models C\) return true;
if there is some \(C \in N\) with \(M \models \neg C\) return false;
select a variable \(P\) occurring in \(N\) but not in \(M\);
if (DPLL(\(N, \text{UProp}(N, M \cup \{P\})\))) then
return true;
else
return DPLL(\(N, \text{UProp}(N, M \cup \{\neg P\})\));

\[-A \lor \neg B \lor E\]
\[-A \lor B\]
\[-E\]
\[A \lor D\]
DPLL(\(N, \emptyset\))
DPLL(\(N, \text{UProp}(N, \{A\})\))
DPLL(\(N, \text{UProp}(N, \{\neg A\})\))
DPLL(\(N, \{A, B, \neg E\}\))
DPLL(\(N, \{\neg A, D, \neg E\}\))

DPLL is sound and complete and terminating for SAT.
Propositional Logic Formulas

Corsa $\Rightarrow$ Wheels $\land$ Engines

4-Holes $\Rightarrow$ Wheels
5-Holes $\Rightarrow$ Wheels
4-Holes $\Rightarrow$ $\neg$5-Holes
5-Holes $\Rightarrow$ $\neg$4-Holes

Diesel $\Rightarrow$ Engines
Gasoline $\Rightarrow$ Engines
Diesel $\Rightarrow$ $\neg$Gasoline
Gasoline $\Rightarrow$ $\neg$Diesel

Diesel $\Rightarrow$ $\neg$4-Holes

Reasoning:

Corsa $\Rightarrow$ Wheels, Engines
4-Holes $\Rightarrow$ $\neg$5-Holes, $\neg$Diesel, Gasoline
Gasoline $\Rightarrow$ $\neg$Diesel
Challenge: Scalability

- worst case SAT searches $2^n$ nodes
- before 2009: approx. 1500 nodes
- in 2011: v.control + SPASS approx. 6000 nodes
- in x years: for a reasonable product approx. 60000 nodes
• Automated Reasoning Lecture:
  http://www.mpi-inf.mpg.de/departments/rg1/teaching/

• contact us on student assistant jobs, bachelor-master-PhD thesis

Thank you for your attention