max planck institut
informatik

The Technology

Prof. Dr. Christoph Weidenbach

Propositional Logic

- Language: propositional variables can be true (1) or false (0)
- Connectives: $\Rightarrow$ implication, $\neg$ negation, $\vee$ disjunction, $\wedge$ 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

$$
\begin{array}{l}
\text{UProp}(N,M) \\
\quad \text{while (there is a clause } C' \lor L \in N \text{ such that} \\
\qquad\qquad M \models \neg C' \text{ and } L \notin M \text{ and } \neg L \notin M) \\
\qquad\quad M := M \cup \{L\}; \\
\quad \text{return } M;
\end{array}
$$

$$
\begin{array}{l}
\text{UProp}(\{\neg A \lor \neg B \lor E, \quad \neg A \lor B, \quad \neg E, \quad D, \quad A\},\emptyset) \\
\qquad \to M = \emptyset \\
\qquad \to M = \{\neg E\} \\
\qquad \to M = \{\neg E, D\} \\
\qquad \to M = \{\neg E, D, A\} \\
\qquad \to M = \{\neg E, D, A, B\}
\end{array}
$$

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 occuring in N but not in M;
  if (DPLL(N, UProp(N, $M \cup \{P\}$))) then
    return true;
  else
    return DPLL(N, UProp(N, $M \cup \{\neg P\}$));

$\neg A \vee \neg B \vee E$
$\neg A \vee B$
$\neg E$
$A \vee D$

DPLL(N, $\emptyset$)

DPLL(N, UProp(N,$\{A\}$)) DPLL(N, 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

$$\mathsf{XK120} \Rightarrow \mathsf{Model} \wedge \mathsf{Engines}$$

$$\mathsf{OTS} \Rightarrow \mathsf{Model}$$
$$\mathsf{FHC} \Rightarrow \mathsf{Model}$$
$$\mathsf{DHC} \Rightarrow \mathsf{Model}$$
$$\mathsf{Model} \Rightarrow \mathsf{OTS} \oplus \mathsf{FHC} \oplus \mathsf{DHC}$$

$$\mathsf{3.4l8C} \Rightarrow \mathsf{Engines}$$
$$\mathsf{3.4l9C} \Rightarrow \mathsf{Engines}$$
$$\mathsf{3.4l9CW} \Rightarrow \mathsf{Engines}$$

$$\mathsf{3.4l9CW} \Rightarrow \mathsf{Race}$$
$$\mathsf{Race} \Rightarrow \mathsf{OTS} \wedge \mathsf{Sports}$$

Reasoning: $\mathsf{XK120} \rightarrow \mathsf{Model}, \mathsf{Engines}$

$\mathsf{FHC} \rightarrow \neg\mathsf{OTS}, \neg\mathsf{DHC}, \neg\mathsf{3.4l9CW}$

Challenge: Scalability

- worst case SAT searches 2^n nodes
- before 2009: approx. 1500 nodes
- in 2012: v.control + SPASS-SATT 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