7 Outlook

Further topics in automated reasoning.

7.1 Satisfiability Modulo Theories (SMT)

DPLL checks satisfiability of propositional formulas.

DPLL can also be used for ground first-order formulas without equality:

Ground first-order atoms are treated like propositional variables.

Truth values of $P(a), Q(a), Q(f(a))$ are independent.

For ground formulas with equality, independence is lost:

If $b \approx c$ is true, then $f(b) \approx f(c)$ must also be true.

Similarly for other theories, e. g. linear arithmetic: $b > 5$ implies $b > 3$.

We can still use DPLL, but we must combine it with a decision procedure for the theory part T:

$M \models_T C$: M and the theory axioms T entail C.

New DPLL rules:

T-Propagate:

$M \parallel N \;\Rightarrow_{\text{DPLL(T)}}\; M\ L \parallel N$

if $M \models_T L$ where L is undefined in M and L or $\overline{L}$ occurs in N.

T-Learn:

$M \parallel N \;\Rightarrow_{\text{DPLL(T)}}\; M \parallel N \cup \{C\}$

if $N \models_T C$ and each atom of C occurs in N or M.

T-Backjump:

$M\ L^{\text{d}}\ M' \parallel N \cup \{C\} \;\Rightarrow_{\text{DPLL(T)}}\; M\ L' \parallel N \cup \{C\}$

if $M\ L^{\text{d}}\ M' \models \neg C$
and there is some "backjump clause" $C' \vee L'$ such that
$N \cup \{C\} \models_T C' \vee L'$ and $M \models \neg C'$,
L' is undefined under M, and
L' or $\overline{L'}$ occurs in N or in $M\ L^{\text{d}}\ M'$.

7.2 Sorted Logics

So far, we have considered only unsorted first-order logic.

In practice, one often considers many-sorted logics:

$read/2$ becomes $read : array \times nat \to data.$

$write/3$ becomes $write : array \times nat \times data \to array.$

Variables: $x : data$

Only one declaration per function/predicate/variable symbol.

All terms, atoms, substitutions must be well-sorted.

Algebras:

Instead of universe $U_{\mathcal{A}}$, one set per sort: $array_{\mathcal{A}}$, $nat_{\mathcal{A}}$.

Interpretations of function and predicate symbols correspond to their declarations:

$read_{\mathcal{A}} : array_{\mathcal{A}} \times nat_{\mathcal{A}} \to data_{\mathcal{A}}$

Proof theory, calculi, etc.:

Essentially as in the unsorted case.

More difficult:

Subsorts

Overloading

7.3 Splitting

Tableau-like rule within resolution to eliminate variable-disjoint (positive) disjunctions:

$$\frac{N \cup \{C_1 \vee C_2\}}{N \cup \{C_1\} \quad | \quad N \cup \{C_2\}}$$

if $var(C_1) \cap var(C_2) = \emptyset$.

Split clauses are smaller and more likely to be usable for simplification.

Splitting tree is explored using intelligent backtracking.

7.4 Integrating Theories into Resolution

Certain kinds of axioms are

important in practice,

but difficult for theorem provers.

Most important case: equality

but also: orderings, (associativity and) commutativity, ...

Idea: Combine ordered resolution and critical pair computation.

Superposition (ground case):

$$\frac{D' \lor t \approx t' \qquad C' \lor s[t] \approx s'}{D' \lor C' \lor s[t'] \approx s'}$$

Superposition (non-ground case):

$$\frac{D' \lor t \approx t' \qquad C' \lor s[u] \approx s'}{(D' \lor C' \lor s[t'] \approx s')\sigma}$$

where $\sigma = \text{mgu}(t, u)$ and u is not a variable.

Advantages:

No variable overlaps (as in KB-completion).

Stronger ordering restrictions:
Only overlaps of (strictly) maximal sides of (strictly) maximal literals are required.

Stronger redundancy criteria.

Similarly for orderings:

Ordered chaining:

$$\frac{D' \;\lor\; t' < t \qquad C' \;\lor\; s < s'}{(D' \;\lor\; C' \;\lor\; t' < s')\sigma}$$

where σ is a most general unifier of t and s.

Integrating other theories:

Black box:

Use external decision procedure.

Easy, but works only under certain restrictions.

White box:

Integrate using specialized inference rules and theory unification.

Hard work.

Often: integrating more theory axioms is better.