Propositional Satisfiability (SAT): Ordered Resolution

Tjark Weber

+ slides by Daniel Le Berre

The resolution principle and classical simplification rules

John Alan Robinson, "A Machine-Oriented Logic Based on the Resolution Principle", Communications of the ACM, 5:23-41, 1965.

resolution: $$\frac{x_1 \vee x_2 \vee x_3 \qquad x_1 \vee \neg x_2 \vee x_4}{x_1 \vee x_1 \vee x_3 \vee x_4}$$

merging: $$\frac{x_1 \vee x_1 \vee x_3 \vee x_4}{x_1 \vee x_3 \vee x_4}$$

subsumption: $$\frac{\alpha \vee \beta \qquad \alpha}{\alpha}$$

The resolution principle and classical simplification rules

John Alan Robinson, "A Machine-Oriented Logic Based on the Resolution Principle", Communications of the ACM, 5:23-41, 1965.

resolution: $$\frac{x_1 \vee x_2 \vee x_3 \qquad x_1 \vee \neg x_2 \vee x_4}{x_1 \vee x_1 \vee x_3 \vee x_4}$$

merging: $$\frac{x_1 \vee x_1 \vee x_3 \vee x_4}{x_1 \vee x_3 \vee x_4}$$

subsumption: $$\frac{\alpha \vee \beta \qquad \alpha}{\alpha}$$

What happens if we apply resolution between $\neg x_1 \vee x_2 \vee x_3$ and $x_1 \vee \neg x_2 \vee x_4$?

The resolution principle and classical simplification rules

John Alan Robinson, "A Machine-Oriented Logic Based on the Resolution Principle", Communications of the ACM, 5:23-41, 1965.

resolution:
$$\frac{x_1 \vee x_2 \vee x_3 \qquad x_1 \vee \neg x_2 \vee x_4}{x_1 \vee x_1 \vee x_3 \vee x_4}$$

merging:
$$\frac{x_1 \vee x_1 \vee x_3 \vee x_4}{x_1 \vee x_3 \vee x_4}$$

subsumption:
$$\frac{\alpha \vee \beta \qquad \alpha}{\alpha}$$

What happens if we apply resolution between $\neg x_1 \vee x_2 \vee x_3$ and $x_1 \vee \neg x_2 \vee x_4$?

A tautology: $x_2 \vee \neg x_2 \vee x_3 \vee x_4$.

Applying resolution to decide satisfiability

- Apply resolution between clauses with exactly one opposite literal
- possible outcome:
  - a new clause is derived: remove subsumed clauses
  - the resolvent is subsumed by an existing clause
- until empty clause derived or no new clause derived
- Main issues of the approach:
  - In which order should the resolution steps be performed?
  - huge memory consumption!

The Davis and Putnam procedure: basic idea

Davis, Martin; Putnam, Hillary (1960). "A Computing Procedure for Quantification Theory". Journal of the ACM 7 (3): 201-215.

Resolution used for variable elimination: $(A \vee x) \wedge (B \vee \neg x) \wedge R$ is satisfiable iff $(A \vee B) \wedge R$ is satisfiable.

- Iteratively apply the following steps:
  - Select variable x
  - Apply resolution between every pair of clauses of the form $(x \vee \alpha)$ and $(\neg x \vee \beta)$
  - Remove all clauses containing either x or $\neg x$
- Terminate when either the empty clause or the empty formula is derived

Proof system: ordered resolution

Variable elimination – An Example

$$(x_1 \lor \neg x_2 \lor \neg x_3) \land (\neg x_1 \lor \neg x_2 \lor \neg x_3) \land (x_2 \lor x_3) \land (x_3 \lor x_4) \land (x_3 \lor \neg x_4) \quad \models$$

Variable elimination – An Example

$$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_3) \wedge (x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \quad \models$$

$$(\neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_3) \wedge (x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \quad \models$$

Variable elimination – An Example

$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_3) \wedge (x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \models$

$(\neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_3) \wedge (x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \models$

$(x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \models$

Variable elimination – An Example

$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_3) \wedge (x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \quad \models$

$(\neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_3) \wedge (x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \quad \models$

$(x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \quad \models$

$x_3 \quad \models$

UNIVERSITÉ D'ARTOIS

Variable elimination – An Example

$$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_3) \wedge (x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \quad \models$$

$$(\neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_3) \wedge (x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \quad \models$$

$$(x_3 \vee x_4) \wedge (x_3 \vee \neg x_4) \quad \models$$

$$x_3 \quad \models$$

$$\top$$

- Formula is SAT

DP60: The limits

- The approach runs easily out of memory.
- Even recent attempts using a ROBDD representation [Simon and Chatalic 2000] do not scale well.
- The solution: using backtrack search!