ENS 2016 - 2017

Satisfiability Modulo Theories (SMT)

Sylvain Conchon

LRI (UMR 8623), Université Paris-Sud
Équipe Toccata, INRIA Saclay – Île-de-France

Road map

- The SMT problem
- Modern efficient SAT solvers
- CDCL(T)
- Examples of decision procedures: equality (CC) and difference logic (NCCD)
- Quantifiers

What is the SMT problem ?

SMT

Satisfiability Modulo Theories

=

SAT solver + Decision Procedures

Checking satisfiability of formulas in a decidable combination of first-order theories (e.g. arithmetic, uninterpreted functions, etc.)

SMT Solving

Input: a (quantifier-free) first-order formula F

Output: the status of F (sat or unsat), and optionally a model (when sat) or a proof (when unsat)

Basic SMT Solving

Given a quantifier-free formula F

$x + y \geq 0 \wedge (x = z \Rightarrow y + z = -1) \wedge z > 3t$ satisfiable ?

1. Convert F to CNF form
2. Replace every literal by a Boolean variable
3. Ask a SAT solver for a Boolean model M
4. Convert back M and call a decision procedure for the union of theories

if M is satisfiable modulo theories, then so is F

otherwise, add $\neg M$ to F and go to step 2

Basic SMT Solving : Example

$$x + y \geq 0 \wedge (x = z \Rightarrow y + z = -1) \wedge z > 3t$$

Basic SMT Solving : Example

$$x + y \geq 0 \land (x = z \Rightarrow y + z = -1) \land z > 3t$$

1. CNF conversion

Basic SMT Solving : Example

$$x + y \geq 0 \wedge (x \neq z \vee y + z = -1) \wedge z > 3t$$

1. CNF conversion

Basic SMT Solving : Example

$$x + y \geq 0 \land (x \neq z \lor y + z = -1) \land z > 3t$$

1. CNF conversion
2. Replace arithmetic constraints by Boolean variables

Basic SMT Solving : Example

$$p_1 \wedge (p_2 \vee p_3) \wedge p_4$$

1. CNF conversion
2. Replace arithmetic constraints by Boolean variables

Basic SMT Solving : Example

$$p_1 \wedge (p_2 \vee p_3) \wedge p_4$$

1. CNF conversion
2. Replace arithmetic constraints by Boolean variables
3. Ask the SAT solver for a model

Basic SMT Solving : Example

$$M = \{p_1 = true,\, p_2 = false,\, p_3 = true,\, p_4 = true\}$$

1. CNF conversion
2. Replace arithmetic constraints by Boolean variables
3. Ask the SAT solver for a model

Basic SMT Solving : Example

$$M = \{p_1 = true, p_2 = false, p_3 = true, p_4 = true\}$$

1. CNF conversion
2. Replace arithmetic constraints by Boolean variables
3. Ask the SAT solver for a model
4. Convert the model back to arithmetic

Basic SMT Solving : Example

$$M = \{x + y \geq 0,\ x = z,\ y + z = -1,\ z > 3t\}$$

1. CNF conversion
2. Replace arithmetic constraints by Boolean variables
3. Ask the SAT solver for a model
4. Convert the model back to arithmetic

Basic SMT Solving : Example

$$M = \{x + y \geq 0,\ x = z,\ y + z = -1,\ z > 3t\}$$

1. CNF conversion
2. Replace arithmetic constraints by Boolean variables
3. Ask the SAT solver for a model
4. Convert the model back to arithmetic
5. Check its consistency with the appropriate decision procedure for arithmetic

Basic SMT Solving : Example

M is unsatisfiable modulo arithmetic!

1. CNF conversion
2. Replace arithmetic constraints by Boolean variables
3. Ask the SAT solver for a model
4. Convert the model back to arithmetic
5. Check its consistency with the appropriate decision procedure for arithmetic

Basic SMT Solving : Example

M is unsatisfiable modulo arithmetic!

1. CNF conversion
2. Replace arithmetic constraints by Boolean variables
3. Ask the SAT solver for a model
4. Convert the model back to arithmetic
5. Check its consistency with the appropriate decision procedure for arithmetic
6. Add $\neg M$ to F and go back to step 2

Basic SMT Solving : Example

$$x + y \geq 0 \land (x \neq z \lor y + z = -1) \land z > 3t \land$$
$$\neg(x + y \geq 0 \land x = z \land y + z = -1 \land z > 3t)$$

1. CNF conversion
2. Replace arithmetic constraints by Boolean variables
3. Ask the SAT solver for a model
4. Convert the model back to arithmetic
5. Check its consistency with the appropriate decision procedure for arithmetic
6. Add $\neg M$ to F and go back to step 2

Main Issues

- Size of formulas
- Complex Boolean structure
- Combination of theories
- Efficient decision procedures
- (Quantifiers)

SMT-Lib

The Satisfiability Modulo Theory Library

http://www.smtlib.org/

International initiative:

- Rigorous description of background theories
- Common input and output languages for SMT solvers
- Large benchmarks

The SMT Revolution

70's:	Stanford Pascal Verifier (Nelson-Oppen combination)
1984:	Shostak algorithm
1992:	Simplify
1995:	SVC
2001:	ICS
2002:	CVC, haRVey
2004:	CVC Lite
2005:	Barcelogic, MathSAT
2005:	Yices
2006:	CVC3, Alt-Ergo
2007:	Z3, MathSAT4
2008:	Boolector, OpenSMT, Beaver,Yices2
2009:	STP, VeriT
2010:	MathSAT5, SONOLAR
2011:	STP2, SMTInterpol
2012:	CVC4

SMT : Building Blocks

Three main blocks:

- SAT Solver
- Decision Procedures
- Combining Decision Procedures framework (CDP)

Modern SAT solvers

SAT Solvers

Is $(p \vee q \vee \neg r) \wedge (r \vee \neg p)$ satisfiable?

- Truth tables
- Resolution-based procedure (DP [1960])
- Backtracking-based procedure (DPLL [1962])
- 80's - 90's: focus on variable selection heuristics
- Search-pruning techniques: Non-chronological backtracking, Learning clauses (Grasp [1996]) CDCL
- Indexing: two-watched literals (Zchaff, 2001)
- Scoring: deletion of bad learning clauses (Glucose, 2009)

Propositional Logic : Notations

p, q, r, s are propositional variables or atoms

l is a literal (p or $\neg p$)

$$\neg l = \begin{cases} \neg p & \text{if } l \text{ is } p \\ p & \text{if } l \text{ is } \neg p \end{cases}$$

A disjunction of literals $l_1 \vee \ldots \vee l_n$ is a clause

The empty clause is written $\perp$

A conjunction of clauses is a CNF

To improve readability, we sometime

- denote atoms by natural numbers and negation by overlining
- write CNF as sets of clauses

e.g. $(\neg l_1 \vee L_2 \vee \neg l_3) \wedge (l_4 \vee \neg 2)$ is simply written $\{\bar{1} \vee 2 \vee \bar{3}, 4 \vee \bar{2}\}$

Propositional Logic : Assignments

An assignment M is a set of literals such that if $l \in M$ then $\neg l \notin M$

A literal l is true in M if $l \in M$, and false if $\neg l \in M$

A literal l is defined in M if it is either true or false in M

A clause is true in M if at least one of its literal is true in M, it is false if all its literals are false in M, it is undefined otherwise

The empty clause $\bot$ is not satisfiable

A clause $C \vee l$ is a unit clause in M if C is false in M and l is undefined in M

Propositional Logic : Satisfiability

A CNF F is satisfied by M (or M is a model of F), written $M \models F$, if all clauses of F are true in M

If F has no model then it is unsatisfiable

F' is entailed by F, written $F \models F'$, if F' is true in all models of F

F and F' are equivalent when $F \models F'$ and $F' \models F$

F and F' are equisatisfiable when
F is satisfiable if and only if F' is satisfiable

F is valid if and only if $\neg F$ is unsatisfiable

Resolution

- Proof-finder procedure
- Works by saturation until the empty clause is derived

Exhaustive resolution is not practical:

exponential amount of memory

Resolution : State of the Procedure

The state of the procedure is represented by a variable (imperative style) F containing a set of clauses (CNF)

Resolution : Algorithm

$$\textsc{Resolve}\ \frac{C \vee l \in F \qquad D \vee \neg l \in F \qquad C \vee D \notin F}{F := F \cup \{C \vee D\}}$$

$$\textsc{Empty}\ \frac{l \in F \qquad \neg l \in F}{F := F \cup \bot}$$

$$\textsc{Tauto}\ \frac{F = F' \uplus \{C \vee l \vee \neg l\}}{F := F'}$$

$$\textsc{Subsume}\ \frac{F = F' \uplus \{C \vee D\} \qquad C \in F'}{F := F'}$$

$$\textsc{Fail}\ \frac{\bot \in F}{\mathsf{return}\,\textsc{Unsat}}$$

Resolution : Example

$$F = \{\bar{1} \vee \bar{2} \vee 3,\ \bar{1} \vee 2,\ 1 \vee 3,\ \bar{3}\}$$

Resolution : Example

$$\textsc{Resolve}\ \frac{\bar{1} \vee \bar{2} \vee 3 \in F \qquad 1 \vee 3 \in F}{F := F \cup \{\bar{2} \vee 3\}}$$

$$F = \{\bar{1} \vee \bar{2} \vee 3,\ \bar{1} \vee 2,\ 1 \vee 3,\ \bar{3}\}$$

Resolution : Example

$$\textsc{Resolve}\ \frac{\bar{1} \vee \bar{2} \vee 3 \in F \qquad 1 \vee 3 \in F}{F := F \cup \{\bar{2} \vee 3\}}$$

$$F = \{\bar{1} \vee \bar{2} \vee 3,\ \bar{1} \vee 2,\ 1 \vee 3,\ \bar{3},\ \bar{2} \vee 3\}$$

Resolution : Example

$$\textsc{Subsume}\ \frac{F = F' \uplus \{\bar{1} \lor \bar{2} \lor 3\} \qquad \bar{2} \lor 3 \in F'}{F := F'}$$

$$F = \{\bar{1} \lor \bar{2} \lor 3,\ \bar{1} \lor 2,\ 1 \lor 3,\ \bar{3},\ \bar{2} \lor 3\}$$

Resolution : Example

$$\textsc{Subsume}\ \frac{F = F' \uplus \{\bar{1} \vee \bar{2} \vee 3\} \qquad \bar{2} \vee 3 \in F'}{F := F'}$$

$$F = \{\bar{1} \vee 2,\ 1 \vee 3,\ \bar{3},\ \bar{2} \vee 3\}$$

Resolution : Example

$$\textsc{Resolve}\ \frac{\bar{1} \vee 2 \in F \qquad 1 \vee 3 \in F}{F := F \cup \{2 \vee 3\}}$$

$$F = \{\bar{1} \vee 2,\ 1 \vee 3,\ \bar{3},\ \bar{2} \vee 3\}$$

Resolution : Example

$$\textsc{Resolve}\ \frac{\bar{1} \vee 2 \in F \qquad 1 \vee 3 \in F}{F := F \cup \{2 \vee 3\}}$$

$$F = \{\bar{1} \vee 2,\ 1 \vee 3,\ \bar{3},\ \bar{2} \vee 3, 2 \vee 3\}$$

Resolution : Example

$$\textsc{Resolve}\ \frac{\bar{2} \vee 3 \in F \qquad 2 \vee 3 \in F}{F := F \cup \{3\}}$$

$$F = \{\bar{1} \vee 2,\ 1 \vee 3,\ \ \bar{3},\ \bar{2} \vee 3, 2 \vee 3\}$$

Resolution : Example

$$\textsc{Resolve}\ \frac{\bar{2} \vee 3 \in F \qquad 2 \vee 3 \in F}{F := F \cup \{3\}}$$

$$F = \{\bar{1} \vee 2,\ 1 \vee 3,\ \bar{3},\ \bar{2} \vee 3, 2 \vee 3, 3\}$$

Resolution : Example

$$\textsc{Empty}\ \frac{3 \in F \qquad \bar{3} \in F}{F := F \cup \{\bot\}}$$

$$F = \{\bar{1} \vee 2,\ 1 \vee 3,\ \bar{3},\ \bar{2} \vee 3, 2 \vee 3, 3\}$$

Resolution : Example

$$\textsc{Empty}\ \frac{3 \in F \qquad \bar{3} \in F}{F := F \cup \{\bot\}}$$

$$F = \{\bar{1} \vee 2,\ 1 \vee 3,\ \bar{3},\ \bar{2} \vee 3, 2 \vee 3, 3, \bot\}$$

Resolution : Example

$$\textsc{Fail}\ \frac{\bot \in F}{\text{return } \textsc{Unsat}}$$

$$F = \{\bar{1} \vee 2,\ 1 \vee 3,\ \ \bar{3},\ \bar{2} \vee 3, 2 \vee 3, 3, \bot\}$$

DPLL

DPLL is a model-finder procedure that builds incrementally a model M for a CNF formula F by

- deducing the truth value of a literal l from M and F by Boolean Constraint Propagations (BCP)

  If $C \vee l \in F$ and $M \models \neg C$ then l must be true

- guessing the truth value of an unassigned literal

  If $M \cup \{l\}$ leads to a model for which F is unsatisfiable then backtrack and try $M \cup \{\neg l\}$

DPLL : State of the Procedure

The state of the procedure is represented by

- a variable F containing a set of clauses (CNF)
- a variable M containing a list of literals

DPLL : Algorithm

$$\textsc{Success}\ \frac{M \models F}{\text{return}\ \ \textsc{Sat}}$$

$$\textsc{Unit}\ \frac{C \vee l \in F \qquad M \models \neg C \qquad l \text{ is undefined in } M}{M := l :: M}$$

$$\textsc{Decide}\ \frac{l \text{ is undefined in } M \qquad l \text{ (or } \neg l) \in F}{M := l^{@} :: M}$$

$$\textsc{Backtrack}\ \frac{\begin{array}{c} C \in F \quad M \models \neg C \quad M = M_1 :: l^{@} :: M_2 \\ M_1 \text{ contains no decision literals} \end{array}}{M := \neg l :: M_2}$$

$$\textsc{Fail}\ \frac{C \in F \qquad M \models \neg C \qquad M \text{ contains no decision literals}}{\text{return}\ \ \textsc{Unsat}}$$

DPLL : Example

$$M = []$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Decide}\ \frac{1 \text{ is undefined in } M \qquad \bar{1} \in F}{M := 1^{@} :: M}$$

$$M = []$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Decide}\ \frac{1 \text{ is undefined in } M \qquad \bar{1} \in F}{M := 1^{@} :: M}$$

$M = [1^{@}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

DPLL : Example

$$\textsc{Unit}\ \frac{\bar{1} \vee 2 \in F \qquad M \models 1 \qquad 2 \text{ is undefined in } M}{M := 2 :: M}$$

$$M = [1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Unit}\ \frac{\bar{1} \vee 2 \in F \qquad M \models 1 \qquad 2 \text{ is undefined in } M}{M := 2 :: M}$$

$$M = [2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Decide}\ \frac{3 \text{ is undefined in } M \qquad \bar{3} \in F}{M := 3^{@} :: M}$$

$$M = [2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Decide}\ \frac{3 \text{ is undefined in } M \qquad \bar{3} \in F}{M := 3^{@} :: M}$$

$M = [3^{@}; 2; 1^{@}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

DPLL : Example

$$\textsc{Unit}\ \frac{\bar{3} \vee 4 \in F \qquad M \models 3 \qquad 4 \text{ is undefined in } M}{M := 4 :: M}$$

$M = [3^{@}; 2; 1^{@}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

DPLL : Example

$$\textsc{Unit}\ \frac{\bar{3} \vee 4 \in F \qquad M \models 3 \qquad 4 \text{ is undefined in } M}{M := 4 :: M}$$

$$M = [4; 3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Decide}\ \frac{5 \text{ is undefined in } M \qquad \bar{5} \in F}{M := 5^{@} :: M}$$

$$M = [4; 3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Decide}\ \frac{5 \text{ is undefined in } M \qquad \bar{5} \in F}{M := 5^{@} :: M}$$

$M = [5^{@}; 4; 3^{@}; 2; 1^{@}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

DPLL : Example

$$\textsc{Unit}\ \frac{\bar{5} \vee \bar{6} \in F \qquad M \models 5 \qquad \bar{6} \text{ is undefined in } M}{M := \bar{6} :: M}$$

$$M = [5^{@}; 4; 3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Unit}\ \frac{\bar{5} \vee \bar{6} \in F \qquad M \models 5 \qquad \bar{6} \text{ is undefined in } M}{M := \bar{6} :: M}$$

$$M = [\bar{6}; 5^{@}; 4; 3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Backtrack}\ \frac{\begin{array}{c} 6 \vee \bar{5} \vee \bar{2} \in F \\ M \models \bar{6} \wedge 5 \wedge 2 \qquad M = [6] :: 5^{@} :: [4; 3^{@}; 2; 1^{@}] \end{array}}{M := \bar{5} :: [4; 3^{@}; 2; 1^{@}]}$$

$$M = [\bar{6}; 5^{@}; 4; 3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Backtrack}\ \frac{\begin{array}{c} 6 \vee \bar{5} \vee \bar{2} \in F \\ M \models \bar{6} \wedge 5 \wedge 2 \qquad M = [6] :: 5^{@} :: [4; 3^{@}; 2; 1^{@}] \end{array}}{M := \bar{5} :: [4; 3^{@}; 2; 1^{@}]}$$

$$M = [\bar{5}; 4; 3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Unit}\ \frac{5 \vee 7 \in F \qquad M \models \bar{5} \qquad 7 \text{ is undefined in } M}{M := 7 :: M}$$

$$M = [\bar{5}; 4; 3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Unit}\ \frac{5 \vee 7 \in F \qquad M \models \bar{5} \qquad 7 \text{ is undefined in } M}{M := 7 :: M}$$

$$M = [7; \bar{5}; 4; 3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Backtrack}\ \frac{\begin{array}{c} 5 \vee \bar{7} \vee \bar{2} \in F \\ M \models \bar{5} \wedge 7 \wedge 2 \qquad M = [7; \bar{5}; 4] :: 3^{@} :: [2; 1^{@}] \end{array}}{M := \bar{3} :: [2; 1^{@}]}$$

$$M = [7; \bar{5}; 4; 3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Backtrack}\ \frac{5 \vee \bar{7} \vee \bar{2} \in F \qquad M \models \bar{5} \wedge 7 \wedge 2 \qquad M = [7;\bar{5};4] :: 3^{@} :: [2;1^{@}]}{M := \bar{3} :: [2;1^{@}]}$$

$M = [\bar{3};2;1^{@}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

DPLL : Example

$$\textsc{Decide}\ \frac{5 \text{ is undefined in } M \qquad \bar{5} \in F}{M := 5^{@} :: M}$$

$$M = [\bar{3}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Decide}\ \frac{5 \text{ is undefined in } M \qquad \bar{5} \in F}{M := 5^{@} :: M}$$

$M = [5^{@}; \bar{3}; 2; 1^{@}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

DPLL : Example

$$\textsc{Unit}\ \frac{\bar{5} \vee \bar{6} \in F \qquad M \models 5 \qquad \bar{6} \text{ is undefined in } M}{M := \bar{6} :: M}$$

$$M = [5^{@}; \bar{3}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Unit}\ \frac{\bar{5} \vee \bar{6} \in F \qquad M \models 5 \qquad \bar{6}\ \text{is undefined in}\ M}{M := \bar{6} :: M}$$

$$M = [\bar{6}; 5^{@}; \bar{3}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Backtrack}\ \frac{6 \vee \bar{5} \vee \bar{2} \in F \qquad M \models \bar{6} \wedge 5 \wedge 2 \qquad M = [\bar{6}] :: 5^{@} :: [\bar{3}; 2; 1^{@}]}{M := \bar{5} :: [\bar{3}; 2; 1^{@}]}$$

$M = [\bar{6}; 5^{@}; \bar{3}; 2; 1^{@}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

DPLL : Example

$$\textsc{Backtrack}\ \frac{\begin{array}{c}6 \lor \bar{5} \lor \bar{2} \in F \\ M \models \bar{6} \land 5 \land 2 \qquad M = [\bar{6}] :: 5^{@} :: [\bar{3}; 2; 1^{@}]\end{array}}{M := \bar{5} :: [\bar{3}; 2; 1^{@}]}$$

$$M = [\bar{5}; \bar{3}; 2; 1^{@}]$$

$$F = \{\bar{1} \lor 2, \bar{3} \lor 4, \bar{5} \lor \bar{6}, 6 \lor \bar{5} \lor \bar{2}, 5 \lor 7, 5 \lor \bar{7} \lor \bar{2}\}$$

DPLL : Example

$$\textsc{Unit}\ \frac{5 \vee 7 \in F \qquad M \models \bar{5} \qquad 7 \text{ is undefined in } M}{M := 7 :: M}$$

$$M = [\bar{5}; \bar{3}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Unit}\ \frac{5 \vee 7 \in F \qquad M \models \bar{5} \qquad 7 \text{ is undefined in } M}{M := 7 :: M}$$

$$M = [7; \bar{5}; \bar{3}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Backtrack}\ \frac{M \models \bar{5} \wedge 7 \wedge 2 \qquad \overset{5 \vee \bar{7} \vee \bar{2} \in F}{M = [7; 5; \bar{3}; 2] :: 1@ :: []}}{M := \bar{1} :: []}$$

$$M = [7; \bar{5}; \bar{3}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Backtrack}\ \frac{5 \vee \bar{7} \vee \bar{2} \in F \qquad M \models \bar{5} \wedge 7 \wedge 2 \qquad M = [7; 5; \bar{3}; 2] :: 1@ :: []}{M := \bar{1} :: []}$$

$M = [\bar{1}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

DPLL : Example

$$\textsc{Decide}\ \frac{\bar{3} \text{ is undefined in } M \qquad \bar{3} \in F}{M := \bar{3}^{@} :: M}$$

$M = [\bar{1}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

DPLL : Example

$$\textsc{Decide}\ \frac{\bar{3} \text{ is undefined in } M \qquad \bar{3} \in F}{M := \bar{3}^{@} :: M}$$

$$M = [\bar{3}^{@}; \bar{1}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Decide}\ \frac{\bar{5} \text{ is undefined in } M \qquad \bar{5} \in F}{M := \bar{5}^{@} :: M}$$

$M = [\bar{3}^{@}; \bar{1}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

DPLL : Example

$$\textsc{Decide}\ \frac{\bar{5} \text{ is undefined in } M \qquad \bar{5} \in F}{M := \bar{5}^{@} :: M}$$

$M = [\bar{5}^{@}; \bar{3}^{@}; \bar{1}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

DPLL : Example

$$\textsc{Unit}\ \frac{5 \vee 7 \in F \qquad M \models \bar{5} \qquad 7 \text{ is undefined in } M}{M := 7 :: M}$$

$$M = [\bar{5}^{@}; \bar{3}^{@}; \bar{1}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Unit}\ \frac{5 \vee 7 \in F \qquad M \models \bar{5} \qquad 7 \text{ is undefined in } M}{M := 7 :: M}$$

$M = [7; \bar{5}^{@}; \bar{3}^{@}; \bar{1}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

DPLL : Example

$$\textsc{Unit}\ \frac{5 \vee \bar{7} \vee \bar{2} \in F \qquad M \models \bar{5} \wedge 7 \qquad \bar{2} \text{ is undefined in } M}{M := \bar{2} :: M}$$

$$\begin{aligned} M &= [7; \bar{5}^{@}; \bar{3}^{@}; \bar{1}] \\ F &= \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\} \end{aligned}$$

DPLL : Example

$$\textsc{Unit}\ \frac{5 \vee \bar{7} \vee \bar{2} \in F \qquad M \models \bar{5} \wedge 7 \qquad \bar{2} \text{ is undefined in } M}{M := \bar{2} :: M}$$

$$M = [\bar{2}; 7; \bar{5}^{@}; \bar{3}^{@}; \bar{1}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

DPLL : Example

$$\textsc{Success}\ \frac{M \models F}{\text{return } \textsc{Sat}}$$

$$M = [\bar{2}; 7; \bar{5}^{@}; \bar{3}^{@}; \bar{1}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping

- The clause $6 \vee \bar{5} \vee \bar{2}$ is false in $[\bar{6}; 5^{@}; 4; 3^{@}; 2; 1^{@}]$
- It is also false in $[\bar{6}; 5^{@}; \quad ; 2; 1^{@}]$
- Instead of backtracking to $M = [\bar{5}; 4; 3^{@}; 2; 1^{@}]$, we would prefer to backjump directly to $M = [\bar{5}; 2; 1^{@}]$

Backjump Clauses

Conflict are reflected by backjump clauses

For instance, we have the following backjump clauses in the previous example:

$$F \models \bar{1} \vee \bar{5}$$
$$F \models \bar{2} \vee \bar{5}$$

Given a backjump clause $C \vee l$, backjumping can undo several decisions at once: it goes back to the assignment M where $M \models \neg C$ and add l to M

DPLL + Backjumping

We just replace Backtrack by

$$\textsc{Backjump}\ \frac{C \in F \quad M \models \neg C \quad M = M_1 :: l^{@} :: M_2 \\ F \models C' \vee l' \quad M_2 \models \neg C' \\ l' \text{ is undefined in } M_2 \quad l' \text{ (or } \neg l') \in F}{M := l' :: M_2}$$

where $C' \vee l'$ is a backjump clause

Backjumping : Example

$$M = []$$

$$F = \{\bar{1} \lor 2, \bar{3} \lor 4, \bar{5} \lor \bar{6}, 6 \lor \bar{5} \lor \bar{2}, 5 \lor 7, 5 \lor \bar{7} \lor \bar{2}\}$$

Backjumping : Example

$$\textsc{Decide}\ \frac{1 \text{ is undefined in } M \qquad \bar{1} \in F}{M := 1^{@} :: M}$$

$M = []$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

Backjumping : Example

$$\textsc{Decide}\ \frac{1 \text{ is undefined in } M \qquad \bar{1} \in F}{M := 1^{@} :: M}$$

$M = [1^{@}]$

$F = \{\bar{1} \lor 2, \bar{3} \lor 4, \bar{5} \lor \bar{6}, 6 \lor \bar{5} \lor \bar{2}, 5 \lor 7, 5 \lor \bar{7} \lor \bar{2}\}$

Backjumping : Example

$$\textsc{Unit}\ \frac{\bar{1} \lor 2 \in F \qquad M \models 1 \qquad 2 \text{ is undefined in } M}{M := 2 :: M}$$

$M = [1^{@}]$

$F = \{\bar{1} \lor 2, \bar{3} \lor 4, \bar{5} \lor \bar{6}, 6 \lor \bar{5} \lor \bar{2}, 5 \lor 7, 5 \lor \bar{7} \lor \bar{2}\}$

Backjumping : Example

$$\textsc{Unit}\ \frac{\bar{1} \vee 2 \in F \qquad M \models 1 \qquad 2 \text{ is undefined in } M}{M := 2 :: M}$$

$$M = [2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Decide}\ \frac{3 \text{ is undefined in } M \qquad \bar{3} \in F}{M := 3^{@} :: M}$$

$M = [2; 1^{@}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

Backjumping : Example

$$\textsc{Decide}\ \frac{3 \text{ is undefined in } M \qquad \bar{3} \in F}{M := 3^{@} :: M}$$

$$M = [3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Unit}\ \frac{\bar{3} \vee 4 \in F \qquad M \models 3 \qquad 4 \text{ is undefined in } M}{M := 4 :: M}$$

$$M = [3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Unit}\ \frac{\bar{3} \vee 4 \in F \qquad M \models 3 \qquad 4 \text{ is undefined in } M}{M := 4 :: M}$$

$M = [4; 3^{@}; 2; 1^{@}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

Backjumping : Example

$$\textsc{Decide}\ \frac{5 \text{ is undefined in } M \qquad \bar{5} \in F}{M := 5^{@} :: M}$$

$$M = [4; 3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Decide}\ \frac{5 \text{ is undefined in } M \qquad \bar{5} \in F}{M := 5^{@} :: M}$$

$$M = [5^{@}; 4; 3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Unit}\ \frac{\bar{5} \vee \bar{6} \in F \qquad M \models 5 \qquad \bar{6} \text{ is undefined in } M}{M := \bar{6} :: M}$$

$$M = [5^{@}; 4; 3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Unit}\ \frac{\bar{5} \vee \bar{6} \in F \qquad M \models 5 \qquad \bar{6} \text{ is undefined in } M}{M := \bar{6} :: M}$$

$M = [\bar{6}; 5^{@}; 4; 3^{@}; 2; 1^{@}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

Backjumping : Example

$$\textsc{Backjump}\ \frac{\begin{array}{c}6 \vee \bar{5} \vee \bar{2} \in F \qquad M \models \bar{6} \wedge 5 \wedge 2 \\ M = [6; 5^{@}; 4] :: 3^{@} :: [2; 1^{@}] \qquad F \models \bar{2} \vee \bar{5} \\ [2; 1^{@}] \models 2 \qquad \bar{5} \text{ is undefined in } [2; 1^{@}]\end{array}}{M := \bar{5} :: [2; 1^{@}]}$$

$$M = [\bar{6}; 5^{@}; 4; 3^{@}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Backjump}\ \frac{\begin{array}{c} 6 \vee \bar{5} \vee \bar{2} \in F \qquad M \models \bar{6} \wedge 5 \wedge 2 \\ M = [6; 5^{@}; 4] :: 3^{@} :: [2; 1^{@}] \qquad F \models \bar{2} \vee \bar{5} \\ [2; 1^{@}] \models 2 \qquad \bar{5} \text{ is undefined in } [2; 1^{@}] \end{array}}{M := \bar{5} :: [2; 1^{@}]}$$

$$M = [\bar{5}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Unit}\ \frac{5 \vee 7 \in F \qquad M \models \bar{5} \qquad 7 \text{ is undefined in } M}{M := 7 :: M}$$

$$M = [\bar{5}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Unit}\ \frac{5 \vee 7 \in F \qquad M \models \bar{5} \qquad 7 \text{ is undefined in } M}{M := 7 :: M}$$

$M = [7; \bar{5}; 2; 1^{@}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

Backjumping : Example

$$\textsc{Backjump}\ \frac{\begin{array}{c} 5 \vee \bar{7} \vee \bar{2} \in F \\ M \models \bar{5} \wedge 7 \wedge 2 \qquad M = [7; \bar{5}; 2] :: 1^{@} :: [] \\ F \models \bar{1} \qquad [] \models \mathit{true} \qquad \bar{1} \text{ is undefined in } [] \end{array}}{M := \bar{1} :: []}$$

$$M = [7; \bar{5}; 2; 1^{@}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Backjump}\ \frac{\begin{array}{c} 5 \vee \bar{7} \vee \bar{2} \in F \\ M \models \bar{5} \wedge 7 \wedge 2 \qquad M = [7; \bar{5}; 2] :: 1^{@} :: [] \\ F \models \bar{1} \qquad [] \models \mathit{true} \qquad \bar{1} \text{ is undefined in } [] \end{array}}{M := \bar{1} :: []}$$

$$M = [\bar{1}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Decide}\ \frac{\bar{3} \text{ is undefined in } M \qquad \bar{3} \in F}{M := \bar{3}^{@} :: M}$$

$M = [\bar{1}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

Backjumping : Example

$$\textsc{Decide}\ \frac{\bar{3} \text{ is undefined in } M \qquad \bar{3} \in F}{M := \bar{3}^{@} :: M}$$

$M = [\bar{3}^{@}; \bar{1}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

Backjumping : Example

$$\textsc{Decide}\ \frac{\bar{5} \text{ is undefined in } M \qquad \bar{5} \in F}{M := \bar{5}^{@} :: M}$$

$M = [\bar{3}^{@}; \bar{1}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

Backjumping : Example

$$\textcolor{red}{\textsc{Decide}}\ \frac{\bar{5} \text{ is undefined in } M \qquad \bar{5} \in F}{M := \bar{5}^{@} :: M}$$

$M = [\bar{5}^{@}; \bar{3}^{@}; \bar{1}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

Backjumping : Example

$$\textsc{Unit}\ \frac{5 \vee 7 \in F \qquad M \models \bar{5} \qquad 7 \text{ is undefined in } M}{M := 7 :: M}$$

$$M = [\bar{5}^{@}; \bar{3}^{@}; \bar{1}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Unit}\ \frac{5 \vee 7 \in F \qquad M \models \bar{5} \qquad 7 \text{ is undefined in } M}{M := 7 :: M}$$

$M = [7; \bar{5}^{@}; \bar{3}^{@}; \bar{1}]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

Backjumping : Example

$$\textsc{Unit}\ \frac{5 \vee \bar{7} \vee \bar{2} \in F \qquad M \models \bar{5} \wedge 7 \qquad \bar{2} \text{ is undefined in } M}{M := \bar{2} :: M}$$

$$M = [7; \bar{5}^{@}; \bar{3}^{@}; \bar{1}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Unit}\ \frac{5 \vee \bar{7} \vee \bar{2} \in F \qquad M \models \bar{5} \wedge 7 \qquad \bar{2} \text{ is undefined in } M}{M := \bar{2} :: M}$$

$$M = [\bar{2}; 7; \bar{5}^{@}; \bar{3}^{@}; \bar{1}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

Backjumping : Example

$$\textsc{Success}\ \frac{M \models F}{\text{return } \textsc{Sat}}$$

$$M = [\bar{2}; 7; \bar{5}^{@}; \bar{3}^{@}; \bar{1}]$$

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

CDCL

Conflict-Driven Clause Learning SAT solvers (CDCL) add backjump clauses to M as learned clauses (or lemmas) to prevent future similar conflicts.

$$\textsc{Learn}\ \frac{F \models C \quad \text{each atom of } C \text{ occurs in } F \text{ or } M}{F := F \cup \{C\}}$$

Lemmas can also be removed from M

$$\textsc{Forget}\ \frac{F = F' \uplus C \qquad F' \models C}{F := F'}$$

How to Find Backjump Clauses?

1. Build an implication graph that captures the way propagation literals have been derived from decision literals
2. Use the implication graph to explain a conflict (by a specific cutting technique) and extract backjump clauses

Implication Graph

An implication graph G is a DAG that can be built during the run of DPLL as follows:

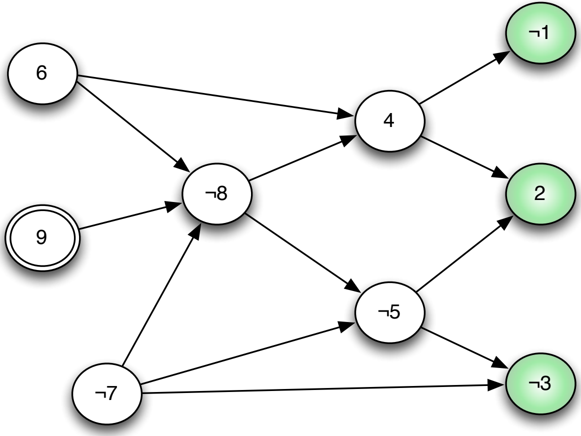
1. Create a node for each decision literal
2. For each clause $l_1 \vee \ldots \vee l_n \vee l$ such that $\neg l_1, \ldots, \neg l_n$ are nodes in G, add a node for l (if not already present in the graph), and add edges $\neg l_i \to l$, for $1 \leq i \leq n$ (if not already present)

Implication Graph : Example

(Partial) implication graph for the following state of DPLL

$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$

$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$



Cutting the Implication Graph

To extract backjump clauses, we first cut the implication graph in two parts:

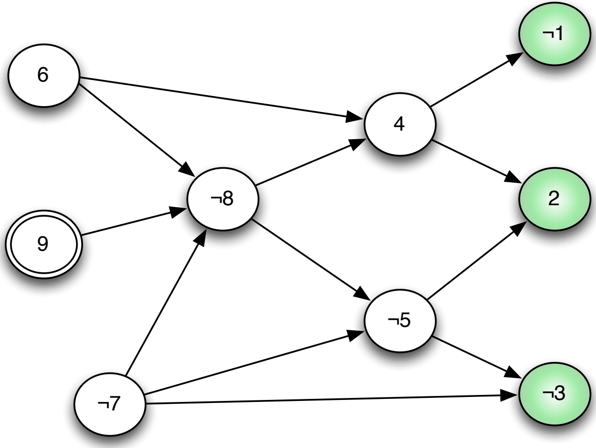
- the first part must contains (at least) all the nodes with no incoming arrows
- the second part must contains (at least) all the nodes with no outgoing arrows

The literals whose outgoing edges are cut form a backjump clause provided that exactly one of these literals belongs to the current decision level.

Cutting the Implication Graph: Example

$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$

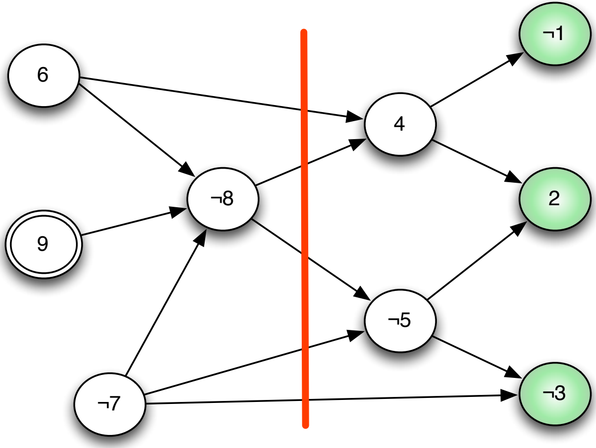
$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$



Cutting the Implication Graph: Example

$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$

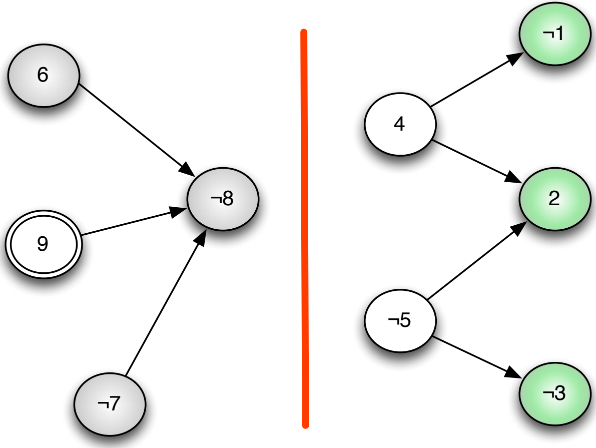
$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$



Cutting the Implication Graph: Example

$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$

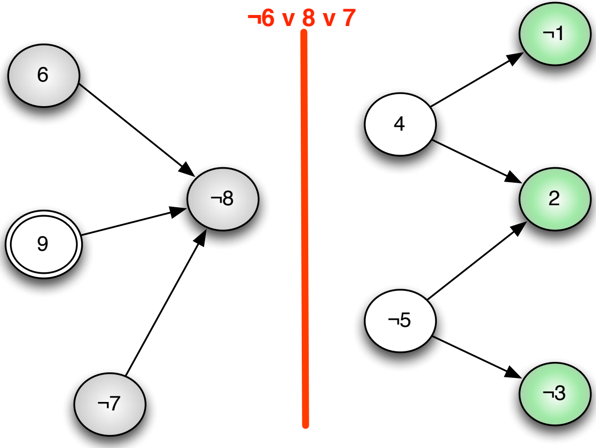
$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$



Cutting the Implication Graph: Example

$$F = \{\bar{9} \lor \bar{6} \lor 7 \lor \bar{8}, 8 \lor 7 \lor \bar{5}, \bar{6} \lor 8 \lor 4, \bar{4} \lor \bar{1}, \bar{4} \lor 5 \lor 2, 5 \lor 7 \lor \bar{3}, 1 \lor \bar{2} \lor 3\}$$

$$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$$

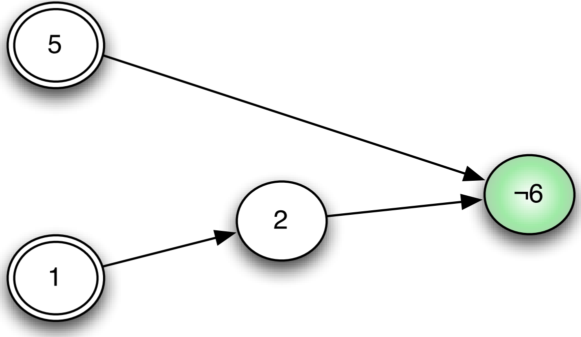


Cutting the Implication Graph : Other Example

In the first example, Backjump is applied for the first time when

$$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$$

$$M = [\bar{6}; 5^{@}; 4; 3^{@}; 2; 1^{@}]$$

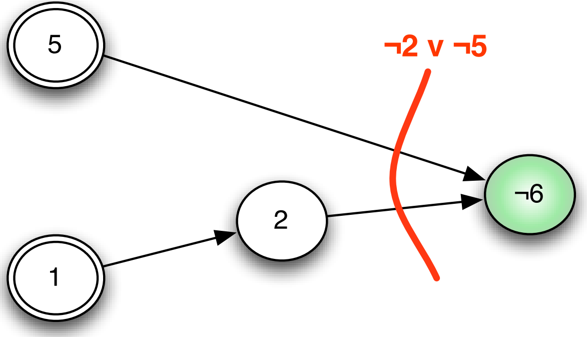


Cutting the Implication Graph : Other Example

In the first example, Backjump is applied for the first time when

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$M = [\bar{6}; 5^{@}; 4; 3^{@}; 2; 1^{@}]$

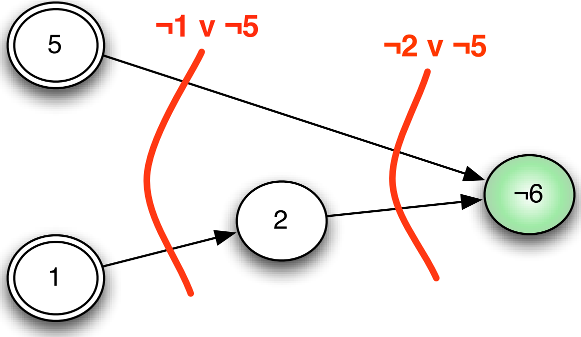


Cutting the Implication Graph : Other Example

In the first example, Backjump is applied for the first time when

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$M = [\bar{6}; 5^{@}; 4; 3^{@}; 2; 1^{@}]$

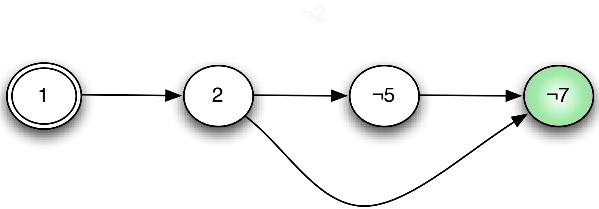


Cutting the Implication Graph : Other Example

When Backjump is applied for the second time, we have

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$M = [7; \bar{5}; 2; 1^{@}]$

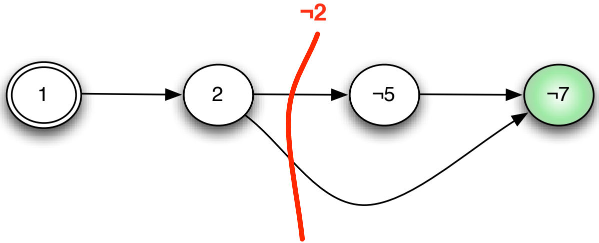


Cutting the Implication Graph : Other Example

When Backjump is applied for the second time, we have

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$M = [7; \bar{5}; 2; 1^{@}]$

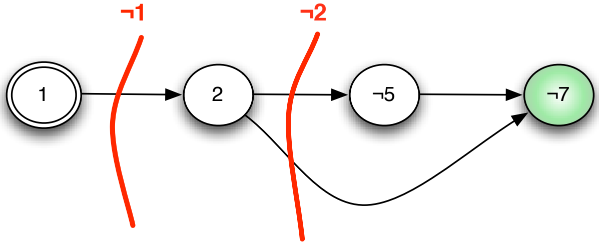


Cutting the Implication Graph : Other Example

When Backjump is applied for the second time, we have

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$M = [7; \bar{5}; 2; 1^{@}]$



Backward Conflict Resolution

Backjump clauses can also be obtained by successive application of resolution steps

Starting from the conflict clause, the (negation of) propagation literals are resolved away in the reverse order with the respective clauses that caused their propagations

We stop when the resolvent contains only one literal in the current decision level

Backward Conflict Resolution: Example

$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$

$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$

$$R = 1 \vee \bar{2} \vee 3$$

Backward Conflict Resolution: Example

$$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$$

$$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$$

$$\textsc{Resolve}\ \frac{R = 1 \vee \bar{2} \vee 3 \qquad 5 \vee 7 \vee \bar{3} \in F}{R := 5 \vee 7 \vee 1 \vee \bar{2}}$$

$$R = 1 \vee \bar{2} \vee 3$$

Backward Conflict Resolution: Example

$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$

$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$

$$\textsc{Resolve}\ \frac{R = 1 \vee \bar{2} \vee 3 \qquad 5 \vee 7 \vee \bar{3} \in F}{R := 5 \vee 7 \vee 1 \vee \bar{2}}$$

$$R = 5 \vee 7 \vee 1 \vee \bar{2}$$

Backward Conflict Resolution: Example

$$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$$

$$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$$

$$\textsc{Resolve}\ \frac{R = 5 \vee 7 \vee 1 \vee \bar{2} \qquad \bar{4} \vee 5 \vee 2 \in F}{R := \bar{4} \vee 5 \vee 7 \vee 1}$$

$$R = 5 \vee 7 \vee 1 \vee \bar{2}$$

Backward Conflict Resolution: Example

$$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$$

$$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$$

$$\textsc{Resolve}\ \frac{R = 5 \vee 7 \vee 1 \vee \bar{2} \qquad \bar{4} \vee 5 \vee 2 \in F}{R := \bar{4} \vee 5 \vee 7 \vee 1}$$

$$R = \bar{4} \vee 5 \vee 7 \vee 1$$

Backward Conflict Resolution: Example

$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$

$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$

$$\textsc{Resolve}\ \frac{R = \bar{4} \vee 5 \vee 7 \vee 1 \qquad \bar{4} \vee \bar{1} \in F}{R := 5 \vee 7 \vee \bar{4}}$$

$$R = \bar{4} \vee 5 \vee 7 \vee 1$$

Backward Conflict Resolution: Example

$F = \{\bar{9} \lor \bar{6} \lor 7 \lor \bar{8}, 8 \lor 7 \lor \bar{5}, \bar{6} \lor 8 \lor 4, \bar{4} \lor \bar{1}, \bar{4} \lor 5 \lor 2, 5 \lor 7 \lor \bar{3}, 1 \lor \bar{2} \lor 3\}$

$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$

$$\textsc{Resolve}\ \frac{R = \bar{4} \lor 5 \lor 7 \lor 1 \qquad \bar{4} \lor \bar{1} \in F}{R := 5 \lor 7 \lor \bar{4}}$$

$$R = 5 \lor 7 \lor \bar{4}$$

Backward Conflict Resolution: Example

$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$

$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$

$$\textsc{Resolve}\ \frac{R = 5 \vee 7 \vee \bar{4} \qquad \bar{6} \vee 8 \vee 4 \in F}{R := \bar{6} \vee 8 \vee 7 \vee 5}$$

$$R = 5 \vee 7 \vee \bar{4}$$

Backward Conflict Resolution: Example

$$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$$

$$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$$

$$\textsc{Resolve}\ \frac{R = 5 \vee 7 \vee \bar{4} \qquad \bar{6} \vee 8 \vee 4 \in F}{R := \bar{6} \vee 8 \vee 7 \vee 5}$$

$$R = \bar{6} \vee 8 \vee 7 \vee 5$$

Backward Conflict Resolution: Example

$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$

$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$

$$\textsc{Resolve}\ \frac{R = \bar{6} \vee 8 \vee 7 \vee 5 \qquad 8 \vee 7 \vee \bar{5} \in F}{R := 8 \vee 7 \vee \bar{6}}$$

$$R = \bar{6} \vee 8 \vee 7 \vee 5$$

Backward Conflict Resolution: Example

$$F = \{\bar{9} \vee \bar{6} \vee 7 \vee \bar{8}, 8 \vee 7 \vee \bar{5}, \bar{6} \vee 8 \vee 4, \bar{4} \vee \bar{1}, \bar{4} \vee 5 \vee 2, 5 \vee 7 \vee \bar{3}, 1 \vee \bar{2} \vee 3\}$$

$$M = [\bar{3}; 2; \bar{1}; 4; \bar{5}; \bar{8}; 9^{@}; \ldots; \bar{7}; \ldots; 6; \ldots]$$

$$\textsc{Resolve}\ \frac{R = \bar{6} \vee 8 \vee 7 \vee 5 \qquad 8 \vee 7 \vee \bar{5} \in F}{R := 8 \vee 7 \vee \bar{6}}$$

$$R = 8 \vee 7 \vee \bar{6}$$

CDCL + Resolution + Learning + Restart

When $Mode = \text{search}$

$$\textsc{Success}\ \frac{M \models F}{\text{return}\ \ \textsc{Sat}}$$

$$\textsc{Unit}\ \frac{C \vee l \in F \qquad M \models \neg C \qquad l \text{ is undefined in } M}{M := l_{C \vee l} :: M}$$

$$\textsc{Decide}\ \frac{l \text{ is undefined in } M \qquad l \text{ (or } \neg l) \in F}{M := l :: M}$$

$$\textsc{Conflict}\ \frac{C \in F \qquad M \models \neg C}{R := C;\ \ Mode := \text{resolution}}$$

CDCL + Resolution + Learning + Restart

When $Mode =$ resolution

$$\textsc{Fail}\ \frac{R = \bot}{\text{return}\quad \textsc{Unsat}}$$

$$\textsc{Resolve}\ \frac{R = C \vee \neg l \qquad l_{D \vee l} \in M}{R := C \vee D}$$

$$\textsc{Backjump}\ \frac{\begin{array}{cc} R = C \vee l & M = M_1 :: l' :: M_2 \\ M_2 \models \neg C & l \text{ is undefined in } M_2 \end{array}}{M := l_{C \vee l} :: M_2;\ Mode := \text{search}}$$

CDCL + Resolution + Learning + Restart

When $Mode$ = resolution

$$\textsc{Learn}\ \frac{R \notin F}{F := F \cup \{R\}}$$

When $Mode$ = search

$$\textsc{Forget}\ \frac{C \text{ is a learned clause}}{F := F \setminus \{C\}}$$

$$\textsc{Restart}\ \frac{}{M := \emptyset}$$

CDCL + Resolution : Example

$Mode = search$

$M = []$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R =$

CDCL + Resolution : Example

$$\textsc{Decide}\ \frac{1 \text{ is undefined in } M \qquad \bar{1} \in F}{M := 1 :: M}$$

$Mode = \mathit{search}$

$M = []$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R =$

CDCL + Resolution : Example

$$\textsc{Decide}\ \frac{1 \text{ is undefined in } M \qquad \bar{1} \in F}{M := 1 :: M}$$

$Mode = search$

$M = [1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R =$

CDCL + Resolution : Example

$$\textsc{Unit}\ \frac{\bar{1} \vee 2 \in F \qquad M \models 1 \qquad 2 \text{ is undefined in } M}{M := 2_{\bar{1}\vee 2} :: M}$$

$Mode = \textit{search}$

$M = [1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R =$

CDCL + Resolution : Example

$$\textsc{Unit}\ \frac{\bar{1} \vee 2 \in F \qquad M \models 1 \qquad 2 \text{ is undefined in } M}{M := 2_{\bar{1} \vee 2} :: M}$$

$Mode = search$

$M = [2_{\bar{1} \vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R =$

CDCL + Resolution : Example

$$\textsc{Decide}\ \frac{3 \text{ is undefined in } M \qquad \bar{3} \in F}{M := 3 :: M}$$

$Mode = \mathit{search}$

$M = [2_{\bar{1} \vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R =$

CDCL + Resolution : Example

$$\text{Decide}\ \frac{3 \text{ is undefined in } M \qquad \bar{3} \in F}{M := 3 :: M}$$

$Mode = search$

$M = [3; 2_{\bar{1}\vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R =$

CDCL + Resolution : Example

$$\textsc{Unit}\ \frac{\bar{3} \lor 4 \in F \qquad M \models 3 \qquad 4 \text{ is undefined in } M}{M := 4_{\bar{3}\lor 4} :: M}$$

$Mode = search$

$M = [3; 2_{\bar{1}\lor 2}; 1]$

$F = \{\bar{1} \lor 2, \bar{3} \lor 4, \bar{5} \lor \bar{6}, 6 \lor \bar{5} \lor \bar{2}, 5 \lor 7, 5 \lor \bar{7} \lor \bar{2}\}$

$R =$

CDCL + Resolution : Example

$$\textsc{Unit}\ \frac{\bar{3} \vee 4 \in F \qquad M \models 3 \qquad 4 \text{ is undefined in } M}{M := 4_{\bar{3}\vee 4} :: M}$$

$Mode = search$

$M = [4_{\bar{3}\vee 4}; 3; 2_{\bar{1}\vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R =$

CDCL + Resolution : Example

$$\textsc{Decide}\ \frac{5 \text{ is undefined in } M \qquad \bar{5} \in F}{M := 5 :: M}$$

$Mode = search$

$M = [4_{\bar{3}\vee 4}; 3; 2_{\bar{1}\vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R =$

CDCL + Resolution : Example

$$\textsc{Decide}\ \frac{5 \text{ is undefined in } M \qquad \bar{5} \in F}{M := 5 :: M}$$

$Mode = search$

$M = [5; 4_{\bar{3}\vee 4}; 3; 2_{\bar{1}\vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R =$

CDCL + Resolution : Example

$$\textsc{Unit}\ \frac{\bar{5} \vee \bar{6} \in F \qquad M \models 5 \qquad \bar{6} \text{ is undefined in } M}{M := \bar{6}_{\bar{5} \vee \bar{6}} :: M}$$

$Mode = search$

$M = [5; 4_{\bar{3} \vee 4}; 3; 2_{\bar{1} \vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R =$

CDCL + Resolution : Example

$$\textsc{Unit}\ \frac{\bar{5} \vee \bar{6} \in F \qquad M \models 5 \qquad \bar{6} \text{ is undefined in } M}{M := \bar{6}_{\bar{5}\vee\bar{6}} :: M}$$

$Mode = search$

$M = [6_{\bar{5}\vee\bar{6}}; 5; 4_{\bar{3}\vee 4}; 3; 2_{\bar{1}\vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$
$R =$

CDCL + Resolution : Example

$$\textsc{Conflict}\ \frac{6 \vee \bar{5} \vee \bar{2} \in F \qquad M \models \bar{6} \wedge 5 \wedge 2}{R := 6 \vee \bar{5} \vee \bar{2}; Mode := \textsf{resolution}}$$

$Mode = search$

$M = [6_{\bar{5}\vee\bar{6}}; 5; 4_{\bar{3}\vee 4}; 3; 2_{\bar{1}\vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$
$R =$

CDCL + Resolution : Example

$$\textsc{Conflict}\ \frac{6 \vee \bar{5} \vee \bar{2} \in F \qquad M \models \bar{6} \wedge 5 \wedge 2}{R := 6 \vee \bar{5} \vee \bar{2}; Mode := \text{resolution}}$$

$Mode = resolution$

$M = [6_{\bar{5} \vee \bar{6}}; 5; 4_{\bar{3} \vee 4}; 3; 2_{\bar{1} \vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R = 6 \vee \bar{5} \vee \bar{2}$

CDCL + Resolution : Example

$$\textsc{Resolve}\ \frac{R = 6 \vee \bar{5} \vee \bar{2} \qquad 6_{\bar{5}\vee\bar{6}} \in M}{R := \bar{2} \vee \bar{5}}$$

$Mode = resolution$

$M = [6_{\bar{5}\vee\bar{6}}; 5; 4_{\bar{3}\vee 4}; 3; 2_{\bar{1}\vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R = 6 \vee \bar{5} \vee \bar{2}$

CDCL + Resolution : Example

$$\textsc{Resolve}\ \frac{R = 6 \vee \bar{5} \vee \bar{2} \qquad 6_{\bar{5}\vee\bar{6}} \in M}{R := \bar{2} \vee \bar{5}}$$

$Mode = resolution$

$M = [6_{\bar{5}\vee\bar{6}}; 5; 4_{\bar{3}\vee 4}; 3; 2_{\bar{1}\vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R = \bar{2} \vee \bar{5}$

CDCL + Resolution : Example

$$\textcolor{red}{\textsc{Backjump}}\ \frac{\begin{array}{c} R = \bar{2} \vee \bar{5} \\ M = [6_{\bar{5}\vee\bar{6}}; 5; 4_{\bar{3}\vee 4}] :: 3 :: [2_{\bar{1}\vee 2}; 1] \\ [2_{\bar{1}\vee 2}; 1] \models 2 \\ \bar{5} \text{ undefined in } [2_{\bar{1}\vee 2}; 1] \end{array}}{M := \bar{5}_{\bar{2}\vee\bar{5}} :: [2_{\bar{1}\vee 2}; 1]; Mode := \text{search}}$$

$Mode = \textcolor{blue}{resolution}$

$M = [6_{\bar{5}\vee\bar{6}}; 5; 4_{\bar{3}\vee 4}; 3; 2_{\bar{1}\vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R = \bar{2} \vee \bar{5}$

CDCL + Resolution : Example

$$\textsc{Backjump}\ \frac{\begin{array}{c} R = \bar{2} \vee \bar{5} \\ M = [6_{\bar{5}\vee\bar{6}}; 5; 4_{\bar{3}\vee 4}] :: 3 :: [2_{\bar{1}\vee 2}; 1] \\ [2_{\bar{1}\vee 2}; 1] \models 2 \\ \bar{5} \text{ undefined in } [2_{\bar{1}\vee 2}; 1] \end{array}}{M := \bar{5}_{\bar{2}\vee\bar{5}} :: [2_{\bar{1}\vee 2}; 1]; Mode := \text{search}}$$

$Mode = search$

$M = [\bar{5}_{\bar{2}\vee\bar{5}}; 2_{\bar{1}\vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R =$

CDCL + Resolution : Example

$etc.$

$Mode = search$

$M = [\bar{5}_{\bar{2} \vee \bar{5}}; 2_{\bar{1} \vee 2}; 1]$

$F = \{\bar{1} \vee 2, \bar{3} \vee 4, \bar{5} \vee \bar{6}, 6 \vee \bar{5} \vee \bar{2}, 5 \vee 7, 5 \vee \bar{7} \vee \bar{2}\}$

$R =$

Strategies

The inference rules given for DPLL and CDCL are flexible

Basic strategy :

- apply DECIDE only if UNIT or FAIL cannot be applied

Conflict resolution :

- Learn only one clause per conflict (the clause used in BACKJUMP)
- Use BACKJUMP as soon as possible (FUIP)
- When applying RESOLVE, use the literals in M in the reverse order they have been added

Decision heuristic : VSIDS

The Variable State Independent Decaying Sum (VSIDS) heuristic associates a score to each literal in order to select the literal with the highest score when Decide is used

- Each literal has a counter, initialized to 0
- Increase the counters of
  - the literal l when Resolve is used
  - the literals of the clause in R when Backjump is used
- Counters are divided by a constant, periodically

Scoring Learned Clauses

CDCL performances are tightly related to their learning clause management

- Keeping too many clauses decrease the BCP efficiency
- Cleaning out too many clauses break the overall learning benefit

Quality measures for learning clauses are based on scores associated with learned clauses

- VSIDS (dynamic): increase the score of clauses involved in Resolve
- LBD (static): number of different decision levels in a learned clause

Indexing

BCP = 80% of SAT-solver runtime

How to implement efficiently $M \models C$ (in UNIT and CONFLICT) ?

Two watched literals technique:

- assign two non-false watched literals per clause
- only if one of the two watched literal becomes false, the clause is inspected :
    - if the other watched literal is assigned to true, then do nothing
    - otherwise, try to find another watched literal
    - if no such literal exists, then apply Backjump
    - if the only possible literal is the other watched literal of the clause, then apply UNIT

Main advantages :

- clauses are inspected only when watched literal are assigned
- no updating when backjumping

CDCL(T)

First-Order Logic : Signature and Terms

- A signature Σ is a finite set of function and predicate symbols with an arity
- Constants are just function symbols of arity 0
- We assume that Σ contains the binary predicate $=$
- We assume a set $\mathcal{V}$ of variables, distinct from Σ
- $T(\Sigma, \mathcal{V})$ is the set of terms, *i.e.* the smallest set which contains $\mathcal{V}$ and such that $f(t_1, \ldots, t_n) \in T(\Sigma, \mathcal{V})$ whenever $t_1, \ldots, t_n \in T(\Sigma, \mathcal{V})$ and $f \in \Sigma$
- $T(\Sigma, \emptyset)$ is the set of ground terms
- Terms are just trees. Given a term t and a position π in a tree, we write t_π for the sub-term of t at position π. We also write $t[\pi \mapsto t']$ for the replacement of the sub-term of t at position π by the term t'

First-Order Logic : Formulas

- An atomic formula is $P(t_1, \dots, t_n)$, where $t_1, \dots, t_n$ are terms in $T(\Sigma, \mathcal{V})$ and P is a predicate symbol of Σ
- Literals are atomic formulas or their negation
- Formulas are inductively constructed from atomic formulas with the help of Boolean connectives and quantifiers $\forall$ and $\exists$
- Ground formulas contain only ground terms
- A variable is free if it is not bound by a quantifier
- A sentence is a formula with no free variables

First-Order Logic : Models

A model $\mathcal{M}$ for a signature Σ is defined by

- a domain $\mathcal{D}_{\mathcal{M}}$
- an interpretation $f^{\mathcal{M}}$ for each function symbol $f \in \Sigma$
- a subset $P^{\mathcal{M}}$ of $\mathcal{D}_{\mathcal{M}}^{n}$ for each predicate $P \in \Sigma$ of arity n
- an assignment $\mathcal{M}(x)$ for each variable $x \in \mathcal{V}$

The cardinality of model $\mathcal{M}$ is the the cardinality of $\mathcal{D}_{\mathcal{M}}$

First-Order Logic : Semantics

Interpretation of terms:

$$\begin{array}{lcl} \mathcal{M}[x] & = & \mathcal{M}(x) \\ \mathcal{M}[f(t_1,\ldots,t_n)] & = & f^{\mathcal{M}}(\mathcal{M}[t_1],\ldots,\mathcal{M}[t_n]) \end{array}$$

Interpretation of formulas:

$$\begin{array}{lcl} \mathcal{M} \models t_1 = t_2 & = & \mathcal{M}[t_1] = \mathcal{M}[t_2] \\ \mathcal{M} \models P(t_1,\ldots,t_n) & = & (\mathcal{M}[t_1],\ldots,\mathcal{M}[t_n]) \in P^{\mathcal{M}} \\ \mathcal{M} \models \neg F & = & \mathcal{M} \not\models F \\ \mathcal{M} \models F_1 \wedge F_2 & = & \mathcal{M} \models F_1 \text{ and } \mathcal{M} \models F_2 \\ \mathcal{M} \models F_1 \vee F_2 & = & \mathcal{M} \models F_1 \text{ or } \mathcal{M} \models F_2 \\ \mathcal{M} \models \forall x.F & = & \mathcal{M}\{x \mapsto v\} \models F \text{ for all } v \in \mathcal{D}_{\mathcal{M}} \\ \mathcal{M} \models \exists x.F & = & \mathcal{M}\{x \mapsto v\} \models F \text{ for some } v \in \mathcal{D}_{\mathcal{M}} \end{array}$$

First-Order Logic : Validity

- A formula F is satisfiable if there a model $\mathcal{M}$ such that $\mathcal{M} \models F$, otherwise F is unsatisfiable
- A formula F is valid if $\neg F$ is unsatisfiable

First-Order Logic : Theories

A first-order theory T over a signature Σ is a set of sentences

A theory is consistent if it has (at least) a model

A formula F is satisfiable in T (or T-satisfiable) if there exists a model $\mathcal{M}$ for $T \wedge F$, written $\mathcal{M} \models_T F$

A formula F is T-validity, denoted $\models_T F$, if $\neg F$ is T-unsatisfiable

Decision Procedures

A decision procedure is an algorithm used to determine whether a formula F in a theory T is satisfiable

Many decision procedures work on conjunctions of (ground) literals

CDCL(T)

We assume a fix theory T

The state of the procedure is similar to CDCL

- F contains quantifier-free clauses in T
- M is a list of literals in T

CDCL(T) : Rules

CDCL(T) has the same rules than CDCL, augmented with

$$\textsc{T-conflict}\ \frac{\begin{array}{c} Mode = \text{search} \\ l_1, \ldots, l_n \in M \qquad l_1, \ldots, l_n \models_T \bot \end{array}}{R := \neg l_1 \vee \ldots \vee \neg l_n; Mode = \text{resolution}}$$

$$\textsc{T-propagate}\ \frac{\begin{array}{c} Mode = \text{search} \\ l (or \neg l) \in F \qquad l \text{ is undefined in } M \\ l_1, \ldots, l_n \in M \qquad l_1, \ldots, l_n \models_T l \end{array}}{M := l_{\neg l_1 \vee \ldots \vee \neg l_n \vee l} :: M}$$

CDCL(T) : Example

$Mode = \textit{search}$

$M = []$

$F = \{3 < x,\ x < 0 \vee x < y,\ y < 0 \vee x \geq y)\}$

$R =$

CDCL(T) : Example

$$\textsc{Unit}\ \frac{3 < x \in F \qquad 3 < x \text{ is undefined in } M}{M := 3 < x_{3<x} :: M}$$

$Mode = \mathit{search}$

$M = []$

$F = \{3 < x,\ x < 0 \vee x < y,\ y < 0 \vee x \geq y)\}$

$R =$

CDCL(T) : Example

$$\textsc{Unit}\ \frac{3 < x \in F \qquad 3 < x \text{ is undefined in } M}{M := 3 < x_{3<x} :: M}$$

$Mode = \mathit{search}$

$M = [3 < x_{3<x}]$

$F = \{3 < x,\ x < 0 \lor x < y,\ y < 0 \lor x \geq y)\}$

$R =$

CDCL(T) : Example

$$\textsc{T-Propagate}\ \frac{x < 0 \in F \text{ is undefined in } M \quad 3 < x \in M \qquad 3 < x \models_T x \geq 0}{M := x \geq 0_{(3 \geq x \vee x \geq 0)} :: M}$$

$Mode = \mathit{search}$

$M = [3 < x_{3<x}]$

$F = \{3 < x,\, x < 0 \vee x < y,\, y < 0 \vee x \geq y)\}$

$R =$

CDCL(T) : Example

$$\textsc{T-Propagate}\ \frac{\begin{array}{c} x < 0 \in F \text{ is undefined in } M \\ 3 < x \in M \qquad 3 < x \models_T x \geq 0 \end{array}}{M := x \geq 0_{(3 \geq x \vee x \geq 0)} :: M}$$

$Mode = \mathit{search}$

$M = [x \geq 0_{(3 \geq x \vee x \geq 0)};\, 3 < x_{3<x}]$

$F = \{3 < x,\, x < 0 \vee x < y,\, y < 0 \vee x \geq y)\}$

$R =$

CDCL(T) : Example

$$\textsc{Unit}\ \frac{x < 0 \lor x < y \in F \qquad M \models_T x \geq 0 \qquad x < y \text{ is undefined in } M}{M := x < y_{(x<0 \lor x<y)} :: M}$$

$Mode = search$

$M = [x \geq 0_{(3 \geq x \lor x \geq 0)};\ 3 < x_{3<x}]$

$F = \{3 < x,\ x < 0 \lor x < y,\ y < 0 \lor x \geq y)\}$

$R =$

CDCL(T) : Example

$$\textsc{Unit}\ \frac{\begin{array}{c} x < 0 \lor x < y \in F \\ M \models_T x \geq 0 \qquad x < y \text{ is undefined in } M \end{array}}{M := x < y_{(x<0 \lor x<y)} :: M}$$

$Mode = search$

$M = [x < y_{(x<0 \lor x<y)};\, x \geq 0_{(3 \geq x \lor x \geq 0)};\, 3 < x_{3<x}]$

$F = \{3 < x,\, x < 0 \lor x < y,\, y < 0 \lor x \geq y)\}$

$R =$

CDCL(T) : Example

$$\textsc{Unit}\ \frac{\begin{array}{c} y < 0 \lor x \geq y \in F \\ M \models_T x < y \qquad y < 0 \text{ is undefined in } M \end{array}}{M := y < 0_{(y<0 \lor x \geq y)} :: M}$$

$Mode = search$

$M = [x < y_{(x<0 \lor x<y)};\, x \geq 0_{(3 \geq x \lor x \geq 0)};\, 3 < x_{3<x}]$

$F = \{3 < x,\, x < 0 \lor x < y,\, y < 0 \lor x \geq y)\}$

$R =$

CDCL(T) : Example

$$\textsc{Unit}\ \frac{y < 0 \vee x \geq y \in F \qquad M \models_T x < y \qquad y < 0 \text{ is undefined in } M}{M := y < 0_{(y<0 \vee x \geq y)} :: M}$$

$Mode = search$

$M = [y < 0_{(y<0\vee x\geq y)};\ x < y_{(x<0\vee x<y)};\ x \geq 0_{(3\geq x\vee x\geq 0)};\ 3 < x_{3<x}]$

$F = \{3 < x,\ x < 0 \vee x < y,\ y < 0 \vee x \geq y)\}$

$R =$

CDCL(T) : Example

$$\textsf{T-CONFLICT}\ \frac{3 < x,\ x < y,\ y < 0 \in M \qquad 3 < x,\ x < y,\ y < 0 \models_T \bot}{R := 3 \geq x \vee x \geq y \vee y \geq 0;\ Mode := \textsf{resolution}}$$

$Mode = search$

$M = [y < 0_{(y<0 \vee x \geq y)};\ x < y_{(x<0 \vee x<y)};\ x \geq 0_{(3 \geq x \vee x \geq 0)};\ 3 < x_{3<x}]$

$F = \{3 < x,\ x < 0 \vee x < y,\ y < 0 \vee x \geq y)\}$

$R =$

CDCL(T) : Example

$$\textsc{T-Conflict}\ \frac{\begin{array}{c}3 < x,\, x < y,\, y < 0 \in M \\ 3 < x,\, x < y,\, y < 0 \models_T \bot\end{array}}{R := 3 \geq x \vee x \geq y \vee y \geq 0;\ Mode := \textsf{resolution}}$$

$Mode = resolution$

$M = [y < 0_{(y<0 \vee x \geq y)};\ x < y_{(x<0 \vee x<y)};\ x \geq 0_{(3 \geq x \vee x \geq 0)};\ 3 < x_{3<x}]$

$F = \{3 < x,\ x < 0 \vee x < y,\ y < 0 \vee x \geq y)\}$

$R = 3 \geq x \vee x \geq y \vee y \geq 0$

CDCL(T) : Example

$$\textsc{Resolve}\ \frac{R = 3 \geq x \vee x \geq y \vee y \geq 0 \qquad y < 0_{(y<0 \vee x \geq y)} \in M}{R := 3 \geq x \vee x \geq y}$$

$Mode = resolution$

$M = [y < 0_{(y<0 \vee x \geq y)};\ x < y_{(x<0 \vee x<y)};\ x \geq 0_{(3 \geq x \vee x \geq 0)};\ 3 < x_{3<x}]$

$F = \{3 < x,\ x < 0 \vee x < y,\ y < 0 \vee x \geq y)\}$

$R = 3 \geq x \vee x \geq y \vee y \geq 0$

CDCL(T) : Example

$$\textsc{Resolve}\ \frac{R = 3 \geq x \vee x \geq y \vee y \geq 0 \qquad y < 0_{(y<0 \vee x \geq y)} \in M}{R := 3 \geq x \vee x \geq y}$$

$Mode = resolution$

$M = [y < 0_{(y<0 \vee x \geq y)};\ x < y_{(x<0 \vee x<y)};\ x \geq 0_{(3 \geq x \vee x \geq 0)};\ 3 < x_{3<x}]$

$F = \{3 < x,\ x < 0 \vee x < y,\ y < 0 \vee x \geq y)\}$

$R = 3 \geq x \vee x \geq y$

CDCL(T) : Example

$$\text{RESOLVE}\ \frac{R = 3 \geq x \vee x \geq y \qquad x < y_{(x<0 \vee x<y)} \in M}{R := 3 \geq x}$$

$Mode = resolution$

$M = [y < 0_{(y<0 \vee x \geq y)};\ x < y_{(x<0 \vee x<y)};\ x \geq 0_{(3 \geq x \vee x \geq 0)};\ 3 < x_{3<x}]$

$F = \{3 < x,\ x < 0 \vee x < y,\ y < 0 \vee x \geq y)\}$

$R = 3 \geq x \vee x \geq y$

CDCL(T) : Example

$$\textsc{Resolve}\ \frac{R = 3 \geq x \vee x \geq y \qquad x < y_{(x<0 \vee x<y)} \in M}{R := 3 \geq x}$$

$Mode = \mathit{resolution}$

$M = [y < 0_{(y<0 \vee x \geq y)};\ x < y_{(x<0 \vee x<y)};\ x \geq 0_{(3 \geq x \vee x \geq 0)};\ 3 < x_{3<x}]$

$F = \{3 < x,\ x < 0 \vee x < y,\ y < 0 \vee x \geq y)\}$

$R = 3 \geq x$

CDCL(T) : Example

$$\textsc{Resolve}\ \frac{R = 3 \geq x \qquad 3 < x_{3<x} \in M}{R := \bot}$$

$Mode = resolution$

$M = [y < 0_{(y<0 \vee x \geq y)};\ x < y_{(x<0 \vee x<y)};\ x \geq 0_{(3 \geq x \vee x \geq 0)};\ 3 < x_{3<x}]$

$F = \{3 < x,\ x < 0 \vee x < y,\ y < 0 \vee x \geq y)\}$

$R = 3 \geq x$

CDCL(T) : Example

$$\textsc{Resolve}\ \frac{R = 3 \geq x \qquad 3 < x_{3<x} \in M}{R := \bot}$$

$Mode = \mathit{resolution}$

$M = [y < 0_{(y<0 \vee x \geq y)};\ x < y_{(x<0 \vee x<y)};\ x \geq 0_{(3 \geq x \vee x \geq 0)};\ 3 < x_{3<x}]$

$F = \{3 < x,\ x < 0 \vee x < y,\ y < 0 \vee x \geq y)\}$

$R = \bot$

CDCL(T) : Example

$$\textsc{Resolve}\ \frac{R = \bot}{\textsf{return}\ \textsc{Unsat}}$$

$Mode = \mathit{resolution}$

$M = [y < 0_{(y<0 \vee x \geq y)};\ x < y_{(x<0 \vee x<y)};\ x \geq 0_{(3 \geq x \vee x \geq 0)};\ 3 < x_{3<x}]$

$F = \{3 < x,\ x < 0 \vee x < y,\ y < 0 \vee x \geq y)\}$

$R = \bot$

Explanations

How to find efficiently $l_1, \ldots, l_n \in M$ such that $l_1, \ldots, l_n \models \bot$?

- In practice, we check for $M \models \bot$ and, if that's true, then we ask the theory solver to produce an explanation, that is, a set of literals $\{l_1, \ldots, l_n\} \subseteq M$ such that $\{l_1, \ldots, l_n\} \models \bot$
- There may be several explanations and some of them may contain irrelevant literals
- Decision procedures try to produce minimal explanations

Theory Propagation

- Similarly to rule Unit, rule T-propagate is optional
- Contrary to rule Unit, the implementation of rule T-propagate can be very costly

How to find efficiently l and $l_1, \ldots, l_n \in M$ s.t $l_1, \ldots, l_n \models l$?

- Theory solver are instrumented to find a literal l implied by M and to return an explanation of the unsatisfiability of $M \wedge \neg l$
- The explanation is also expected to be minimal
- In practice, decision procedures find some implied literals, not all as this can be very costly

Decision Procedures for SMT

Decision procedures found in articles or textbooks need usually to be adapted for being used in SMT solvers

- Incrementally : decision procedures are called successively on set of literals $M_0 \subset M_1 \subset \ldots \subset M_k$

  To gain for efficiency, we don't want to restart from scractch for each M_i but try to reuse work done for M_i when processing M_{i+1}
- Backtracking : operations for going back to a previous state of the decision procedure should be very efficient
- Propagation : find the good tradeoff between precision and performance
- Explanations : find an efficient generation mechanism that removes irrelevant literals (decidability issues)

Examples of decision procedures

The Free Theory of Equality with Uninterpreted Symbols

Axioms:

- Reflexivity $\forall x. x = x$
- Symmetry $\forall x, y. x = y \Rightarrow y = x$
- Transitivity $\forall x, y, z. x = y \wedge y = z \Rightarrow x = z$
- Congruence

$$
\begin{aligned}
&\forall x_1, \ldots, x_n, y_1, \ldots, y_n. \\
&\quad x_1 = y_1 \wedge \cdots \wedge x_n = y_n \Rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)
\end{aligned}
$$

Examples:

$$g(y, x) = y \wedge g(g(y, x), x) \neq y$$

$$f(f(f(a))) = a \wedge f(f(f(f(f(a))))) = a \wedge f(a) \neq a$$

Congruence Closure

Let $\mathcal{R}$ an equivalence relation on terms. The domain of $\mathcal{R}$, denoted by $\mathtt{dom}(\mathcal{R})$, is the set of all terms and subterms of R

- Congruence

  Two terms t and u are congruent by $\mathcal{R}$ if they are respectively of the form $f(t_1, \ldots, t_n)$ and $f(u_1, \ldots, u_n)$ and $(t_i, u_i) \in \mathcal{R}$ for all i

  $\mathcal{R}$ is closed by congruence if for all terms $t, u \in \mathtt{dom}(\mathcal{R})$ congruent par $\mathcal{R}$ we have $(t, u) \in \mathcal{R}$

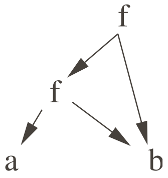
- Congruence Closure

  The congruence closure of $\mathcal{R}$ is the smallest relation containing $\mathcal{R}$ and which is closed by congruence

Representation of Terms and Equality Relation

1. Terms are represented by **DAG** (directed acyclic graphs)

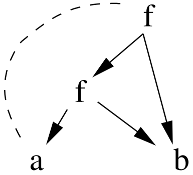
   For instance, $f(f(a,b),b)$ is represented by the following graph



Representation of Terms and Equality Relation

1. Terms are represented by **DAG** (directed acyclic graphs)

   For instance, $f(f(a,b),b)$ is represented by the following graph

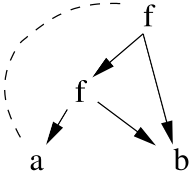
2. $\mathcal{R}$ is represented by dotted lines

   For instance, $f(f(a,b),b) = a$ is represented by a dotted line between f and a

Representation of Terms and Equality Relation

1. Terms are represented by **DAG** (directed acyclic graphs)

   For instance, $f(f(a,b),b)$ is represented by the following graph

2. $\mathcal{R}$ is represented by dotted lines

   For instance, $f(f(a,b),b) = a$ is represented by a dotted line between f and a

3. DAG associated with an equivalence relation are called E-DAG (equality DAG)

Naive Congruence Closure

The equivalent relation $\mathcal{R}$ (the dotted lines) is implemented as a union-find data structure on the nodes of the DAG

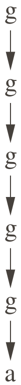
- $\texttt{find}(n)$ returns the representative of the node n
- $\texttt{union}(n, m)$ merges the equivalence classes of n and m

Naive congruence closure algorithm:

```
For every nodes $n$ and $m$ such that $\texttt{find}(n) \neq \texttt{find}(m)$,
 if $n$ and $m$ are labeled with the same symbol and
   they have the same number of children and
   $\texttt{find}(n_i) = \texttt{find}(m_i)$ for every children $n_i$ and $m_i$ of $n$ and $m$
 then, merge the classes of $n$ and $m$ by $\texttt{union}(n, m)$
```

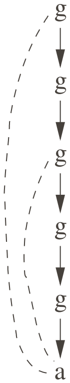
Example

$g(g(g(a))) = a \wedge g(g(g(g(g(a))))) = a \wedge g(a) \neq a$ satisfiable?



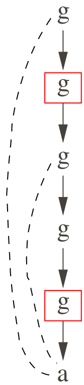
Example

$g(g(g(a))) = a \wedge g(g(g(g(g(a))))) = a \wedge g(a) \neq a$ satisfiable?



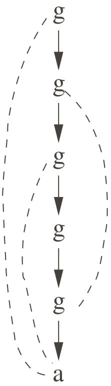
Example

$g(g(g(a))) = a \land g(g(g(g(g(a))))) = a \land g(a) \neq a$ satisfiable?



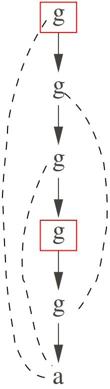
Example

$g(g(g(a))) = a \wedge g(g(g(g(g(a))))) = a \wedge g(a) \neq a$ satisfiable?



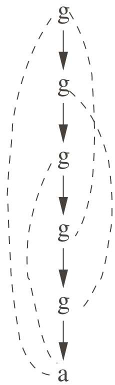
Example

$g(g(g(a))) = a \land g(g(g(g(g(a))))) = a \land g(a) \neq a$ satisfiable?



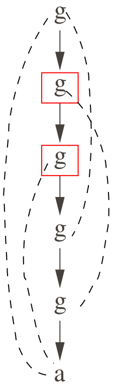
Example

$g(g(g(a))) = a \wedge g(g(g(g(g(a))))) = a \wedge g(a) \neq a$ satisfiable?



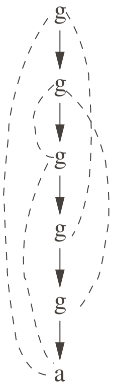
Example

$g(g(g(a))) = a \wedge g(g(g(g(g(a))))) = a \wedge g(a) \neq a$ satisfiable?



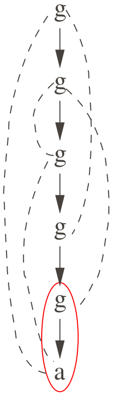
Example

$g(g(g(a))) = a \wedge g(g(g(g(g(a))))) = a \wedge g(a) \neq a$ satisfiable?



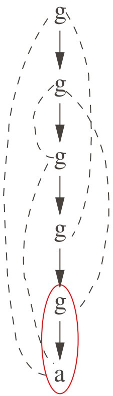
Example

$g(g(g(a))) = a \wedge g(g(g(g(g(a))))) = a \wedge g(a) \neq a$ satisfiable?



Example

$g(g(g(a))) = a \wedge g(g(g(g(g(a))))) = a \wedge g(a) \neq a$ satisfiable?



$g(a) = a$ is implied by the E-DAG

Difference logic

Difference Logic (DL)

$x - y \leq c$ where $x, y, c \in (\mathbb{Q}$ or $\mathbb{Z})$

Strict inequalities

- in $\mathbb{Z}$, $x - y < c$ is replaced $x - y \leq c - 1$
- in $\mathbb{Q}$, $x - y < c$ is replaced $x - y \leq c - \delta$ where δ is a symbolic sufficiently small parameter

Equalities

- $x = y$ is the same as $x - y \leq c \wedge y - x \leq -c$

One variable constraints

- $x \leq c$ is replaced by $x - x_{zero} \leq c$, where x_{zero} is a fresh variable whose value must be 0 in any solution

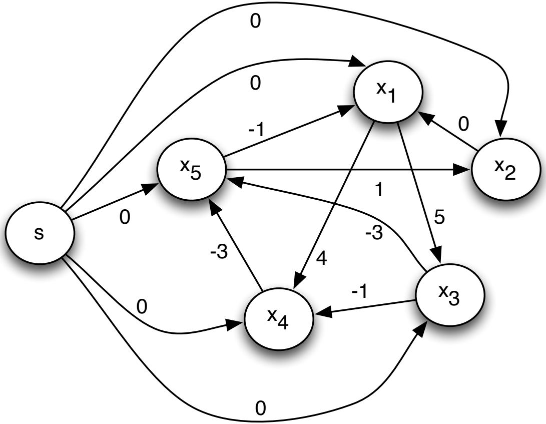
DL : Graph Interpretation

Given a set of difference constraints M, we construct a weighted directed graph $\mathcal{G}_M(V, E)$ as follows :

- the set of vertices V contains the variables of the problem plus an extra variable s
- the set of weighted edges E contains an edge $y \xrightarrow{c} x$ for each constraint $x - y \leq c$, plus an edge $s \xrightarrow{0} x$ for each variable x of the problem

DL : Example

$$x_1 - x_2 \leq 0$$
$$x_1 - x_5 \leq -1$$
$$x_2 - x_5 \leq 1$$
$$x_3 - x_1 \leq 5$$
$$x_4 - x_1 \leq 4$$
$$x_4 - x_3 \leq -1$$
$$x_5 - x_3 \leq -3$$
$$x_5 - x_4 \leq -3$$



DL : Satisfiability and Models

A negative cycle in $\mathcal{G}_M(V, E)$ is a path

$$x_0 \xrightarrow{c_0} x_1 \xrightarrow{c_1} \dots \xrightarrow{c_{n-1}} x_n \xrightarrow{c_n} x_0$$

such that $c_0 + c_1 + \cdots + c_{n-1} + c_n < 0$

Theorem

If $\mathcal{G}_M(V, E)$ has a negative cycle then M is unsatisfiable, otherwise a solution is

$$x_1 = \delta(s, x_1), \dots, x_n = \delta(s, x_n)$$

where $\delta(s, x_i)$ is the shortest-path weight from s to x_i

DL : Correctness

Proof.

Any negative-weight cycle $v_1 \xrightarrow{c_1} v_2 \xrightarrow{c_2} \ldots \xrightarrow{c_{n-1}} v_n \xrightarrow{c_n} v_1$ corresponds to a set of difference constraints

$$\begin{aligned} v_2 - v_1 &\leq c_1 \\ v_3 - v_2 &\leq c_2 \\ \ldots \\ v_1 - v_n &\leq c_n \end{aligned}$$

If we sum them all, we get $0 \leq c_1 + c_2 + \cdots + c_n$ which is impossible since a negative cycle implies $c_1 + c_2 + \cdots + c_n < 0$

Now, if $\mathcal{G}_M(V, E)$ has no negative cycle, for any edge $x_i \xrightarrow{c} x_j$ we have $\delta(s, x_j) \leq \delta(s, x_i) + c$, or equivalently $\delta(s, x_j) - \delta(s, x_i) \leq c$. Thus, letting $x_i = \delta(s, x_i)$ and $x_j = \delta(s, x_j)$ satifies the constraints $x_j - x_i \leq c$

DL : Example (cont)

$x_1 - x_2 \leq 0$

$x_1 - x_5 \leq -1$

$x_2 - x_5 \leq 1$

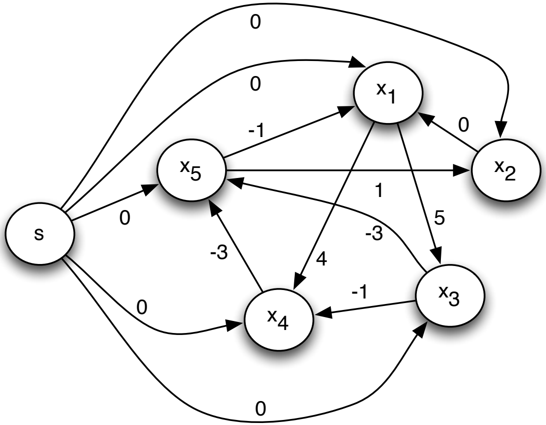
$x_3 - x_1 \leq 5$

$x_4 - x_1 \leq 4$

$x_4 - x_3 \leq -1$

$x_5 - x_3 \leq -3$

$x_5 - x_4 \leq -3$



DL : Example (cont)

$x_1 - x_2 \leq 0$

$x_1 - x_5 \leq -1$

$x_2 - x_5 \leq 1$

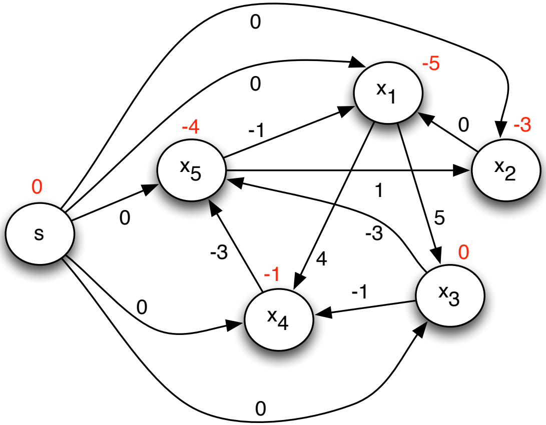
$x_3 - x_1 \leq 5$

$x_4 - x_1 \leq 4$

$x_4 - x_3 \leq -1$

$x_5 - x_3 \leq -3$

$x_5 - x_4 \leq -3$



DL : Example (cont)

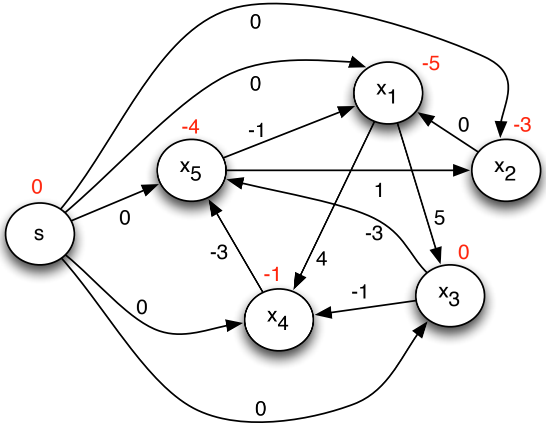
$x_1 = -5$

$x_2 = -3$

$x_3 = 0$

$x_4 = -1$

$x_5 = -4$



Negative Cycle Detection

Negative cycle can be detected with shortest path algorithms

Most algorithms are based on the technique of relaxation

- For each vertex x, we maintain an upper bound $d[x]$ on the weight of a shortest path from s to x
- Relaxing an edge $x \xrightarrow{c} y$ consists in testing whether we can improve the shortest path to y found so far by going through x
- Additionally, shortest paths are saved in an array π that gives the predecessor of each vertex

```
if $d[y] > d[x] + c$ then
    $d[y] := d[x] + c$
    $\pi[y] := x$
```

Bellman-Ford Algorithm

```
for each $x_i \in V$ do $d[x_i] := \infty$ done
$d[s] := 0$

for $i := 1$ to $|V| - 1$ do
    for each $x_i \xrightarrow{c} x_j \in E$ do
        if $d[x_j] > d[x_i] + c$ then
            $d[x_j] := d[x_i] + c$
            $\pi[x_j] := u$
    done
done

for each $x_i \xrightarrow{c} x_j \in E$ do
    if $d[x_j] > d[x_i] + c$ then
        return Negative Cycle Detected
               Follow $\pi$ to reconstruct the cycle
done
```

Bellman-Ford Algorithm : Correctness

Proof.

Suppose that $\mathcal{G}_M(V, E)$ contains a negative cycle $x_0 \xrightarrow{c_0} x_1 \xrightarrow{c_1} \ldots \xrightarrow{c_{k-1}} x_k$ with $x_0 = x_k$. Assume Bellman-Ford does not find the cycle. Thus, $d[x_i] \leq d[x_{i-1}] + c_{i-1}$ for all $i = 1, 2, \ldots, k$. Summing these inequalities, we get

$$\sum_{i=1}^{k} d[x_i] \leq \sum_{i=1}^{k} d[x_{i-1}] + \sum_{i=1}^{k} c_{i-1}$$

Bellman-Ford Algorithm : Correctness

Proof.

Suppose that $\mathcal{G}_M(V, E)$ contains a negative cycle $x_0 \xrightarrow{c_0} x_1 \xrightarrow{c_1} \dots \xrightarrow{c_{k-1}} x_k$ with $x_0 = x_k$. Assume Bellman-Ford does not find the cycle. Thus, $d[x_i] \leq d[x_{i-1}] + c_{i-1}$ for all $i = 1, 2, \dots, k$. Summing these inequalities, we get

$$\sum_{i=1}^{k} d[x_i] - \sum_{i=1}^{k} d[x_{i-1}] \leq \sum_{i=1}^{k} c_{i-1}$$

Bellman-Ford Algorithm : Correctness

Proof.

Suppose that $\mathcal{G}_M(V, E)$ contains a negative cycle $x_0 \xrightarrow{c_0} x_1 \xrightarrow{c_1} \dots \xrightarrow{c_{k-1}} x_k$ with $x_0 = x_k$. Assume Bellman-Ford does not find the cycle. Thus, $d[x_i] \leq d[x_{i-1}] + c_{i-1}$ for all $i = 1, 2, \dots, k$. Summing these inequalities, we get

$$\sum_{i=1}^{k} d[x_i] - \sum_{i=1}^{k} d[x_{i-1}] \leq \sum_{i=1}^{k} c_{i-1}$$

but, since $x_0 = x_k$, we have

$$\sum_{i=1}^{k} d[x_i] = \sum_{i=1}^{k} d[x_{i-1}]$$

Bellman-Ford Algorithm : Correctness

Proof.

Suppose that $\mathcal{G}_M(V, E)$ contains a negative cycle $x_0 \xrightarrow{c_0} x_1 \xrightarrow{c_1} \ldots \xrightarrow{c_{k-1}} x_k$ with $x_0 = x_k$. Assume Bellman-Ford does not find the cycle. Thus, $d[x_i] \leq d[x_{i-1}] + c_{i-1}$ for all $i = 1, 2, \ldots, k$. Summing these inequalities, we get

$$0 \leq \sum_{i=1}^{k} c_{i-1}$$

which is impossible since the cycle is negative

Bellman-Ford Algorithm (cont)

- Checking satisfiability can be performed in time $O(|V|.|E|)$
- Inconsistency explanations are negative cycles (irredundant but not minimal explanations)
- Incremental and backtrackable extensions exist

Quantifiers

Quantified Formulas

Consider the following axiomatization (in Alt-Ergo's syntax) for an ordering relation le

```
logic le: int,int -> prop
axiom refl: forall x:int. le(x,x)
axiom trans:
  forall x,y,z:int. le(x,y) and le(y,z) -> le(x,z)
axiom antisym:
  forall x,y:int. le(x,y) and le(y,x) -> x = y
```

Quantified Formulas

Consider the following axiomatization (in Alt-Ergo's syntax) for an ordering relation le

```
logic le: int,int -> prop
axiom refl: forall x:int. le(x,x)
axiom trans:
  forall x,y,z:int. le(x,y) and le(y,z) -> le(x,z)
axiom antisym:
  forall x,y:int. le(x,y) and le(y,x) -> x = y
```

and some goals we want to prove:

```
goal g1:  le(2,5) and le(5,10) -> le(2,10)
goal g2:
  forall a:int.
    le(a,5) and le(5,8) and le(8,a) -> a=5
```

Guiding Quantifier Instantiation

Many SMT solvers handle universal formulas through an instantiation mechanism

Guiding Quantifier Instantiation

Many SMT solvers handle universal formulas through an instantiation mechanism

Questions:

- How to find good instances to prove a goal?
- How to limit the (prohibitive) number of instances?

Guiding Quantifier Instantiation

Many SMT solvers handle universal formulas through an instantiation mechanism

Questions:

- How to find good instances to prove a goal?
- How to limit the (prohibitive) number of instances?

A possible answer: find good heuristics!

Guiding Quantifier Instantiation

Many SMT solvers handle universal formulas through an instantiation mechanism

Questions:

- How to find good instances to prove a goal?
- How to limit the (prohibitive) number of instances?

A possible answer: find good heuristics!

- In practice, heuristics for choosing new instances are based on triggers : lists of patterns (terms with variables) that guide (or restrict) instantiations to known ground terms that have a given form

Triggers: Example

If `P(x)` is used as trigger in the following axiom ax1

```
logic P,Q,R: int -> prop
axiom ax1:  forall x:int.  (P(x) or Q(x)) -> R(x)
goal g3:  P(1) -> R(1)
goal g4:  Q(2) -> R(2)
```

then, among the set of known terms {`P(1)`,`R(1)`,`P(2)`,`R(2)`}, only `P(1)` can be used to create the following instance of ax1

```
( P(1) or Q(1) ) -> R(1)
```

which implies that only goal g3 is proved

Explicit Triggers

SMT solvers' input syntax provides the possibility for a user to specify its own triggers

For instance, in Alt-Ergo, the list of terms `[f(x), Q(y)]` is an explicit trigger for the following axiom ax2

```
logic P,Q,R: int -> prop
logic f:  int -> int
axiom ax2:
    forall x,y:int [f(x), Q(y)].
                P(f(x)) and Q(y) -> R(x)
```

Matching

We use a matching algorithm to create new instances of universal formulas

Given a ground term t and a pattern p, the matching algorithm returns a set S of substitutions over the variables of p such that

$$t = \sigma(p) \quad \text{for all} \quad \sigma \in S$$

Limitation of Matching

Purely syntactic matching is very limited!

Consider for instance the following formulas:

```
logic P,R : int -> prop
logic f :  int -> int
axiom ax :  forall x:int [P(f(x))].  P(f(x)) -> R(x)
goal g1 :  forall a:int.  P(a) -> a = f(2) -> R(2)
```

The trigger P(f(x)) prevents the creation of instances of axiom ax since there is no ground term of the form P(f(_)) in the problem

To prove such goals, we need to extend the matching algorithm to find substitutions modulo (ground) equalities

E-Matching

Given a set of ground equations E, a ground term t and a pattern p, the e-matching algorithm returns a set S of substitutions over the variables of p such that

$$E \models t = \sigma(p) \quad \text{for all} \quad \sigma \in S$$

In the previous example

```
logic P,R : int -> prop
logic f :  int -> int
axiom ax :  forall x:int [P(f(x))].  P(f(x)) -> R(x)
goal g1 :  forall a:int.  P(a) -> a = f(2) -> R(2)
```

e-matching takes advantage of ground equality `a = f(2)` and returns the substitution $\sigma = \{\mathtt{x} \mapsto 2\}$ which is used to create the instance `P(f(2)) -> R(2)` of axiom ax

Ground Terms

Known ground terms are extracted from literals assumed or implied by the SAT solver

Instantiation based mechanisms are strongly impacted by the number and the relevance of known ground terms :

- more ground terms, more instances of lemmas
- irrelevant ground terms, irrelevant instances

Ground Terms and Linear CNF

The shape of formulas to be proved, and in particular the conversion process used to produce a CNF, has a strong impact on the number of known ground terms

Consider for instance the following formula

$$A \vee (B \wedge C)$$

When A is assumed to be true, terms of A become known and the rest of the (terms of the) formula $(B \wedge C)$ can be ignored

Ground Terms and Linear CNF

The shape of formulas to be proved, and in particular the conversion process used to produce a CNF, has a strong impact on the number of known ground terms

Consider for instance the following formula

$$A \vee (B \wedge C)$$

When A is assumed to be true, terms of A become known and the rest of the (terms of the) formula $(B \wedge C)$ can be ignored

However, because of the shape of the CNF conversion

$$(A \vee X) \wedge (X \Leftrightarrow (B \wedge C))$$

the SMT solver will assign a value to X (even when A is true) and terms from B and C will be considered has known terms :-(