Abstract Solvers for Quantified Boolean Formulas and Their Applications

Remi Brochenin and Marco Maratea

DIBRIS, University of Genoa, Italy

AI*IA 2015: Ferrara, Sept 23rd-25th 2015

Motivation

Issue

- Usually solving procedures are presented by means of pseudo-code descriptions, but
- some communities have experienced that analyzing such procedures on this basis may not be fruitful.

Instead ...

- more formal descriptions, based on mathematically precise but possibly simple objects, can be useful, and
- can allow for, e.g. a uniform representation.

Motivation

Issue

- Usually solving procedures are presented by means of pseudo-code descriptions, but
- some communities have experienced that analyzing such procedures on this basis may not be fruitful.

Instead ...

- more formal descriptions, based on mathematically precise but possibly simple objects, can be useful, and
- can allow for, e.g. a uniform representation.

Abstract solvers

Abstract solvers are a relatively new methodology for analyzing, comparing and composing solving procedures in an abstract way via graphs, where

- the states of computation are represented as nodes,
- the solving techniques as arcs between such nodes,
- the solving process as a path in the graph, and
- formal properties of the procedures are reduced to related graph's properties.

DPLL SAT solving: Transition rules of the graph $DPLL_F$

$$\textit{Conclude}: \quad L \Longrightarrow \textit{UNSAT} \quad \text{if} \begin{cases} L \text{ is inconsistent and} \\ L \text{ contains no decision literals} \end{cases}$$

$$\textit{Backtrack}: \quad L l^{\Delta} L' \Longrightarrow L \bar{l} \quad \text{if} \begin{cases} L l^{\Delta} L' \text{ is inconsistent and} \\ L' \text{ contains no decision literals} \end{cases}$$

$$\textit{Unit}: \quad L \Longrightarrow L l \quad \text{if} \begin{cases} l \text{ does not occur in } L \text{ and} \\ F \text{ contains a clause } C \vee l \text{ and} \\ \text{all the literals of } \overline{C} \text{ occur in } L \end{cases}$$

$$\textit{Decide}: \quad L \Longrightarrow L l^{\Delta} \quad \text{if} \begin{cases} L \text{ is consistent and} \\ \text{neither } l \text{ nor } \bar{l} \text{ occur in } L \end{cases}$$

$$\textit{Success}: \quad L \Longrightarrow \textit{SAT} \quad \text{if no other rule applies}$$

DPLL SAT solving: Examples

Initial state :		$\emptyset$
Decide	$\Longrightarrow$	a^Δ
Unit	$\Longrightarrow$	$a^\Delta\ c$
Decide	$\Longrightarrow$	$a^\Delta\ c\ b^\Delta$
Success	$\Longrightarrow$	*SAT*

Initial state :		$\emptyset$
Decide	$\Longrightarrow$	a^Δ
Decide	$\Longrightarrow$	$a^\Delta\ \overline{c}^\Delta$
Unit	$\Longrightarrow$	$a^\Delta\ \overline{c}^\Delta\ c$
Backtrack	$\Longrightarrow$	$a^\Delta\ c$
Decide	$\Longrightarrow$	$a^\Delta\ c\ b^\Delta$
Success	$\Longrightarrow$	*SAT*

Figure : Examples of paths in $DPLL_{\{a \vee b,\ \overline{a} \vee c\}}$.

DPLL SAT solving: Formal result

Theorem

For any CNF formula F,

1. *graph $DPLL_F$ is finite and acyclic,*
2. *any terminal state reachable from $\emptyset$ in $DPLL_F$ other than $UNSAT$ is SAT, and*
3. *$UNSAT$ is reachable from $\emptyset$ in $DPLL_F$ if and only if F is unsatisfiable.*

QBF is the prototypical PSPACE-complete problem.

- In this paper, three abstract solvers for solving QBFs are presented.
- One proposal abstracts the Q-DPLL algorithm for QBF.
- Q-DPLL is an extension of the DPLL algorithm for SAT.

Q-DPLL for QBFs: Graph

We are given a (prenex CNF) QBF formula F.

QBF_F graph

- The nodes are the states defined similarly as for DPLL, but decision literals are either *universal* ($l^{\forall}$) or *existential* ($l^{\exists}$).
- The edges corresponds to updated and additional transition rules wrt DPLL graph.

Q-DPLL for QBFs: Transition rules

Conclude	L	$\Longrightarrow UNSAT$	if $\{$ L is inconsistent and existential free
Backtrack$_\exists$	$Ll^\exists L'$	$\Longrightarrow L\bar{l}$	if $\{$ $Ll^\exists L'$ is inconsistent and $l^\exists$ is the rightmost existential literal
Backtrack$_\forall$	$Ll^\forall L'$	$\Longrightarrow L\bar{l}$	if $\{$ no other rule applies except *Succeed* and $l^\forall$ is the rightmost universal literal
Unit	L	$\Longrightarrow Ll$	if $\{$ l does not occur in L and for some clause C in the formula, l occurs in C and each other unassigned literal of C is universal and each assigned literal of C is contradicted
Monotone1	L	$\Longrightarrow Ll$	if $\{$ the variable of l is existential and l occurs in some clause C and $\bar{l}$ does not occur in any clause C
Monotone2	L	$\Longrightarrow Ll$	if $\{$ the variable of l is universal and $\bar{l}$ occurs in some clause C and l does not occur in any clause C
Decide	L	$\Longrightarrow Ll^Q$	if $\{$ L is consistent and the variable of l is unassigned and the quantifier of the variable of l is Q and for all l' such that $level(l') < level(l)$ the variable of l' is assigned.
Succeed	L	$\Longrightarrow Valid$	if $\{$ no other rule applies

Figure : The transition rules of the QBF_F graph.

Q-DPLL for QBFs: Example

$$F := \exists a \forall d \exists bc \{\{\overline{a}, \overline{d}, b\}, \{\overline{d}, \overline{b}\}, \{b, c\}, \{a, \overline{d}, \overline{c}\}, \{d, b, \overline{c}\}\} \quad (1)$$

Example

A possible path in QBF_F is:

Initial state :		$\emptyset$
Decide	$\Longrightarrow$	$\overline{a}^{\exists}$
Decide	$\Longrightarrow$	$\overline{a}^{\exists}\,\overline{d}^{\forall}$
*Monotone*1	$\Longrightarrow$	$\overline{a}^{\exists}\,\overline{d}^{\forall}\,b$
$Backtrack_{\forall}$	$\Longrightarrow$	$\overline{a}^{\exists}\,d$
Unit	$\Longrightarrow$	$\overline{a}^{\exists}\,d\,\overline{c}$
Unit	$\Longrightarrow$	$\overline{a}^{\exists}\,d\,\overline{c}\overline{b}$

$Backtrack_{\exists}$	$\Longrightarrow$	a
Decide	$\Longrightarrow$	$a\overline{d}^{\forall}$
*Monotone*1	$\Longrightarrow$	$a\overline{d}^{\forall}\,b$
$Backtrack_{\forall}$	$\Longrightarrow$	$a\,d$
Unit	$\Longrightarrow$	$a\,d\,b$
Fail	$\Longrightarrow$	*UNSAT*

Q-DPLL for QBFs: Formal result

For any QBF formula F,

1. the graph QBF_F is finite and acyclic;
2. Any terminal state in QBF_F is either *UNSAT* or *Valid*;
3. If a state *Valid* is reachable from the initial state in QBF_F then F is satisfiable;
4. *UNSAT* is reachable from the initial state in QBF_F if and only if F is not satisfiable.