

Do SAT Solvers Make Good Configurators?

Mikoláš Janota

Lero
University College Dublin
Ireland

ASPL at SPLC 2008

lero

SFI grant no. 03/CE2/I303.1

Configuration

- a *feature models* represent the set of products we are interested in and the dependencies between them
- customer selects a product in the *configuration process*
- the product should fulfill the desires of the customer but must respect the *constraints* imposed by the feature model

Interactive feedback

<input checked="" type="checkbox"/> Car	<input checked="" type="checkbox"/> Body	<input checked="" type="checkbox"/> Engine	Partial Valuation
<input checked="" type="checkbox"/> Gear	<input type="checkbox"/> Gas	<input type="checkbox"/> Electric	Valuation Completion
<input type="checkbox"/> Manual	<input type="checkbox"/> Automatic	<input type="checkbox"/> Keyless	Quit
<input type="checkbox"/> PowerLocks			
Explanation (Gear)	Unsatisfied constraints		
!Car Gear; Car;	Manual Automatic !Gear; Electric !Engine Gas;		

Interactive feedback

<input checked="" type="checkbox"/> Car	<input checked="" type="checkbox"/> Body	<input checked="" type="checkbox"/> Engine	Partial Valuation Valuation Completion Quit
<input checked="" type="checkbox"/> Gear	<input type="checkbox"/> Gas	<input type="checkbox"/> Electric	
<input type="checkbox"/> Manual	<input type="checkbox"/> Automatic	<input type="checkbox"/> Keyless	
<input type="checkbox"/> PowerLocks			
Explanation (Gear)	Unsatisfied constraints		
!Car Gear; Car;	Manual Automatic !Gear; Electric !Engine Gas;		

Functionality

- feedback takes place as the choices are being made

Interactive feedback

<input checked="" type="checkbox"/> Car	<input checked="" type="checkbox"/> Body	<input checked="" type="checkbox"/> Engine	Partial Valuation Valuation Completion Quit
<input checked="" type="checkbox"/> Gear	<input type="checkbox"/> Gas	<input type="checkbox"/> Electric	
<input type="checkbox"/> Manual	<input type="checkbox"/> Automatic	<input type="checkbox"/> Keyless	
<input type="checkbox"/> PowerLocks			
Explanation (Gear)	Unsatisfied constraints		
!Car Gear; Car;	Manual Automatic !Gear; Electric !Engine Gas;		

Functionality

- feedback takes place as the choices are being made
- disabling choices that do not lead to a solution

Interactive feedback

<input checked="" type="checkbox"/> Car	<input checked="" type="checkbox"/> Body	<input checked="" type="checkbox"/> Engine	Partial Valuation Valuation Completion Quit
<input checked="" type="checkbox"/> Gear	<input type="checkbox"/> Gas	<input type="checkbox"/> Electric	
<input type="checkbox"/> Manual	<input type="checkbox"/> Automatic	<input type="checkbox"/> Keyless	
<input type="checkbox"/> PowerLocks			
Explanation (Gear)		Unsatisfied constraints	
!Car Gear; Car;		Manual Automatic !Gear; Electric !Engine Gas;	

Functionality

- feedback takes place as the choices are being made
- disabling choices that do not lead to a solution
- never let the user violate the constraints (*backtrack freeness*)

Interactive feedback

<input checked="" type="checkbox"/> Car	<input checked="" type="checkbox"/> Body	<input checked="" type="checkbox"/> Engine	Partial Valuation Valuation Completion Quit
<input checked="" type="checkbox"/> Gear	<input type="checkbox"/> Gas	<input type="checkbox"/> Electric	
<input type="checkbox"/> Manual	<input type="checkbox"/> Automatic	<input type="checkbox"/> Keyless	
<input type="checkbox"/> PowerLocks			
Explanation (Gear)		Unsatisfied constraints	
!Car Gear; Car;		Manual Automatic !Gear; Electric !Engine Gas;	

Functionality

- feedback takes place as the choices are being made
- disabling choices that do not lead to a solution
- never let the user violate the constraints (*backtrack freeness*)
- explaining why a value is locked

How Do We Go About This?

Use a SAT Solver

- determines the satisfiability of a given Boolean formula
- operates on Conjunctive Normal Form (CNF)
- a certification of the response is produced
- nowadays SAT solvers are *very* efficient

How Do We Go About This?

Use a SAT Solver

- determines the satisfiability of a given Boolean formula
- operates on Conjunctive Normal Form (CNF)
- a certification of the response is produced
- nowadays SAT solvers are *very* efficient

Assumptions

- constraints encoded in a CNF
- decisions so far encoded as a conjunction of literals

$$\phi \equiv f_1 \wedge \neg f_8 \wedge \dots$$

SAT Solver for Configuration

Testing all free features after each user's decision

```
TEST-VARS()  
1  foreach  $x$  that was not assigned to by the user  
2    do  $CanBeTrue \leftarrow$  TEST-SAT( $\phi, x$ )  
3       $CanBeFalse \leftarrow$  TEST-SAT( $\phi, \neg x$ )  
4      if  $\neg CanBeTrue \wedge \neg CanBeFalse$   
5        then error "Unsatisfiable constraint!"  
6      if  $\neg CanBeTrue$  then SET( $x$ , FALSE)  
7      if  $\neg CanBeFalse$  then SET( $x$ , TRUE)  
8      if  $CanBeTrue \wedge CanBeFalse$   
9        then RESET( $x$ )  
10       UNLOCK( $x$ )  
11     else LOCK( $x$ )
```

Can We Improve This?

SAT

- For satisfiable queries, the SAT solver returns with a satisfying assignment.
- All the values in this assignment are satisfiable and don't need to be queried for.

Can We Improve This?

SAT

- For satisfiable queries, the SAT solver returns with a satisfying assignment.
- All the values in this assignment are satisfiable and don't need to be queried for.

UNSAT

- Can a negative response of the solver help in the future?
- Example

$$\left. \begin{array}{l} f_1 \Rightarrow f_2 \\ \neg f_2 \\ \dots \end{array} \right\} \neg f_1$$

- Recording disabled values *may* help with further queries.

Satisfiability with Caching

- *KnownValues* represent values known to be SAT
- *DisabledValues* represent values known to be UNSAT

TEST-SAT(ϕ : *Formula*, l : *Literal*) : *Boolean*

```
1  if  $l \in \text{KnownValues}$  then return TRUE
2  if  $l \in \text{DisabledValues}$  then return FALSE
3   $L \leftarrow \text{SAT}(\phi \wedge l \wedge \bigwedge_{k \in \text{DisabledValues}} \neg k)$ 
4  if  $L \neq \text{null}$ 
5     then  $\text{KnownValues} \leftarrow \text{KnownValues} \cup L$ 
6     else  $\text{DisabledValues} \leftarrow \text{DisabledValues} \cup \{l\}$ 
7  return  $L \neq \text{null}$ 
```

Explanations

- The solver produces a unsatisfiable subset of given formulas.
- This may not be minimal, several techniques how to minimize.
- In the tool an iterative technique by Zhang and Malik.

Discussion and Future Work

Comparing to Binary Decision Diagrams (BDDs)

- It is expected to be slower than *but* much less likely to choke.
- The form of the formula is preserved and hence can be used in the explanations.
- Requires CNF *however* any formula can be *clausified* in polynomial size.

Discussion and Future Work

Comparing to Binary Decision Diagrams (BDDs)

- It is expected to be slower than *but* much less likely to choke.
- The form of the formula is preserved and hence can be used in the explanations.
- Requires CNF *however* any formula can be *clausified* in polynomial size.

Future work

- Use proofs for more efficient cache discarding.
- Could this approach work for non-Boolean domains?

Discussion and Future Work

Comparing to Binary Decision Diagrams (BDDs)

- It is expected to be slower than *but* much less likely to choke.
- The form of the formula is preserved and hence can be used in the explanations.
- Requires CNF *however* any formula can be *clausified* in polynomial size.

Future work

- Use proofs for more efficient cache discarding.
- Could this approach work for non-Boolean domains?

A java implementation, including a SAT solver, available at
kind.ucd.ie

When to Discard Caches?

- With a new decision asserted, *KnownValues* may change and thus get discarded whereas *DisabledValues* remain the same.
- When a decision is retracted, *KnownValues* remain whereas *DisabledValues* are discarded.