How to Complete an Interactive Configuration Process?

Mikoláš Janota  Goetz Botterweck
Radu Grigore  Joao Marques-Silva

Lero
University College Dublin
Ireland

lero

SFI grant no. 03/CE2/I303_1
Feature Diagrams

Car

Gear

Automatic xor Manual

Power Locks Keyless Entry

requires (⇐)
Configuration

Legend
- Selection
- Deselection
- Autoselect
- Autodeselect

Car
- Gear
  - Automatic
  - Manual
- Power Locks
- Keyless Entry

requires
Configuration

Legend

- Selection
- Deselection
- Autoselect
- Autodeselect
Configuration

Legend
- Selection
- Deselection
- Autoselect
- Autodeselect

Car

Gear

Automatic

Manual

Power Locks

Keyless Entry

requires
Car

Gear

Automatic

Manual

Power Locks

Keyless Entry

requires

Legend

Selection

Deselection

Autoselect

Autodeselect
"I'm Done"

Legend
- Selection
- Deselection
- Autoselect
- Autodeselect
“I’m Done” sometimes doesn’t work
“I’m Done” sometimes doesn’t work
How is a configuration process completed?

**Conditions**

- All features must have a value.
- Values must conform to the constraints.
How is a configuration process completed?

**Conditions**

- All features must have a value.
- Values must conform to the constraints.

**Scenarios**

- **manual** — user fills in everything
How is a configuration process completed?

**Conditions**

- All features must have a value.
- Values must conform to the constraints.

**Scenarios**

- **manual** — user fills in everything
- **blind completion** — automated tool fills in everything
How is a configuration process completed?

**Conditions**

- All features must have a value.
- Values must conform to the constraints.

**Scenarios**

- **manual** — user fills in everything
- **blind completion** — automated tool fills in everything
- **smart completion** — automated tool fills as much as possible *without making decisions* for the user
Smart Completion

Legend
- Selection
- Deselection
- Autoselect
- Autodeselect
Decisions and Dispensable Variables

- A set of variables is **deselectable** iff they can be all deselected all at once.

\[
\mathcal{D}(\phi, X) \overset{\text{def}}{=} SAT\left( \phi \land \bigwedge_{v \in X} \neg v \right)
\]

- A variable is dispensable iff it does not belong to any set that must be decided.

Mikolás Janota et al.
Completing Interactive Configuration
Decisions and Dispensable Variables

- A set of variables is **deselectable iff** they can be all deselected all at once.

\[
\mathcal{D}(\phi, X) \overset{\text{def}}{=} SAT\left(\phi \land \bigwedge_{v \in X} \neg v \right)
\]

- A set of variables \(X\) **must be decided iff**
  1. It is not deselectable.
  2. All of the proper subsets of \(X\) are deselectable.

\[
\neg \mathcal{D}(\phi, X) \land (\forall Y \subsetneq X) (\mathcal{D}(\phi, Y))
\]
Decisions and Dispensable Variables

- A set of variables is deselectable iff they can be all deselected all at once.

\[ D(\phi, X) \overset{\text{def}}{=} SAT\left(\phi \land \bigwedge_{v \in X} \neg v\right) \]

- A set of variables \( X \) must be decided iff
  1. \( X \) it is not deselectable.
  2. All of the proper subsets of \( X \) are deselectable.

\[ \neg D(\phi, X) \land (\forall Y \subsetneq X)(D(\phi, Y)) \]

- A variable is dispensable iff it does not belong to any set that must be decided.
Examples

\[ x \lor y \lor z \]

- Deselectable: \( \{x, y\}, \{x, z\}, \{y, z\}, \{x\}, \{y\}, \{z\}, \) and \( \emptyset \)
- Not deselectable: \( \{x, y, z\} \)
- \( \{x, y, z\} \) must be decided and none of the variables are dispensable.
Examples

\[ x \lor y \lor z \]

- Deselectable: \( \{x, y\}, \{x, z\}, \{y, z\}, \{x\}, \{y\}, \{z\}, \) and \( \emptyset \)
- Not deselectable: \( \{x, y, z\} \)
- \( \{x, y, z\} \) must be decided and none of the variables are dispensable.

\[ x \Rightarrow (y \lor z) \]

- \( \{x, y, z\} \) is deselectable therefore does not have to be decided and all variables are dispensable.
Examples

\[ x \lor y \lor z \]

- Deselectable: \( \{x, y\}, \{x, z\}, \{y, z\}, \{x\}, \{y\}, \{z\}, \text{and } \emptyset \)
- Not deselectable: \( \{x, y, z\} \)
- \( \{x, y, z\} \) must be decided and none of the variables are dispensable.

\[ x \Rightarrow (y \lor z) \]

- \( \{x, y, z\} \) is deselectable therefore does not have to be decided and all variables are dispensable.

\[ (x \Rightarrow (y \lor z)) \land x \]

- \( \{y, z\} \) must be decided.
Relation to Minimal Models

- A model of a formula is (point-wise) minimal iff flipping some True values to False, yields a non-model.
A model of a formula is *(point-wise) minimal* iff flipping some *true* values to *false*, yields a non-model.
A model of a formula is *(point-wise) minimal* iff flipping some *True* values to *False*, yields a non-model.

\[ x \lor (y \land z) \]

\[
\begin{array}{c}
[T,T,T] \\
\uparrow \\
[T,T,F] \\
\downarrow \\
[T,F,F] \\
\uparrow \\
[F,T,F] \\
\downarrow \\
[F,F,F] \\
\downarrow \\
[F,F,T] \\
\downarrow \\
[F,F,F]
\end{array}
\]
A model of a formula is (point-wise) minimal iff flipping some True values to False, yields a non-model.
A variable is dispensable iff it is False in all minimal models.
A variable is dispensable *iff* it is False in all minimal models.

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
Dispensable Variables and Minimal Models

A variable is dispensable *iff* it is False in all minimal models.

\[
x \lor y \lor z
\]

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

\[
x \Rightarrow (y \lor z)
\]

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Non-monotonic Reasoning

- Propositional Circumscription

\[ \phi \models_{\min} \psi \]

- Generalized Closed World Assumption (GCWA)
What about non-boolean configuration?

- For a general set of possibilities, it is hard to help the user.
What about non-boolean configuration?

- For a general set of possibilities, it is hard to help the user.
- But it is possible, if there is a preference on the possibilities.
What about non-boolean configuration?

- For a general set of possibilities, it is hard to help the user.
- But it is possible, if there is a preference on the possibilities.
- We can focus on the most preferred possibilities.
What about non-boolean configuration?

- For a general set of possibilities, it is hard to help the user.
- But it is possible, if there is a preference on the possibilities.
- We can focus on the **most preferred** possibilities.
A value $c$ is **settled** for a variable $v$ *iff* $v$ has the value $c$ in all most preferred possibilities.
A value $c$ is **settled** for a variable $v$ iff $v$ has the value $c$ in all most preferred possibilities.

Settled Values and Dispensable Variables

A variable is dispensable iff False is settled for it under the point-wise preference for False.
A value $c$ is **settled** for a variable $v$ *iff* $v$ has the value $c$ in all most preferred possibilities.

A variable is dispensable *iff* False is settled for it under the point-wise preference for False.
Motivation was to provide **smart completion**.
Summary

- Motivation was to provide **smart completion**.
- In configuration of Boolean constraints it led to defining **dispensable variables**.
Motivation was to provide smart completion.

In configuration of Boolean constraints it led to defining dispensable variables.

Dispensable variables are closely related to CWA.
Summary

- Motivation was to provide smart completion.
- In configuration of Boolean constraints it led to defining dispensable variables.
- Dispensable variables are closely related to CWA.
- In non-Boolean case, smart completion can be provided in the presence of preference.
Summary

- Motivation was to provide *smart completion*.
- In configuration of Boolean constraints it led to defining *dispensable variables*.
- Dispensable variables are closely related to CWA.
- In non-Boolean case, smart completion can be provided in the presence of *preference*.
- Dispensable variables can be seen as a *preference for deselecting*.
- Analogously CWA as a preference for False.
## Experimental Results

<table>
<thead>
<tr>
<th>Name</th>
<th>Features</th>
<th>Clauses</th>
<th>Length</th>
<th>Done</th>
<th>Minimal models</th>
</tr>
</thead>
<tbody>
<tr>
<td>tightvnc</td>
<td>21</td>
<td>22</td>
<td>5.5</td>
<td>5.5</td>
<td>1.0 ± 0.0</td>
</tr>
<tr>
<td>apl</td>
<td>27</td>
<td>41</td>
<td>12.2</td>
<td>11.9</td>
<td>1.0 ± 0.0</td>
</tr>
<tr>
<td>gg4</td>
<td>58</td>
<td>139</td>
<td>10.0</td>
<td>3.8</td>
<td>15.3 ± 22.6</td>
</tr>
<tr>
<td>berkeley</td>
<td>94</td>
<td>183</td>
<td>26.6</td>
<td>17.9</td>
<td>1.7 ± 1.1</td>
</tr>
<tr>
<td>violet</td>
<td>170</td>
<td>341</td>
<td>56.1</td>
<td>47.1</td>
<td>1.6 ± 0.9</td>
</tr>
<tr>
<td>E-shop</td>
<td>287</td>
<td>420</td>
<td>143</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>