SAT-Based Consistency Checking of Automotive Electronic Product Data

Carsten Sinz, Andreas Kaiser and Wolfgang Küchlin
Symbolic Computation Group
University of Tübingen
and Steinbeis Technology Transfer Center OIT

http://www-sr.informatik.uni-tuebingen.de
Introduction

Scenario:
- Electronic configuration of Mercedes car and truck lines
- Rule-based EPDM system already present
- Boolean logic employed to express constraints and to control processing of orders

Problem:
- Complexity of product and documentation induces errors

Goals:
- Computer-based assistance in finding potential errors
- Increasing documentation quality
Customer’s order consists of a set of Boolean variables (codes) describing the model class and additional features.

Order processing performed in three steps:
1. Order completion
2. Constructibility check
3. Parts list generation

All steps controlled by evaluating logical rules.
Step 1: Order completion

- Interpretation of supplementing rules $Cond^S \rightarrow x$:
  Code $x$ is added to a customer’s order $O$, if $x$’s supplementing rule ($Cond^S$) evaluates to true under $O$.

- Notes:
  1. Steering restriction has to be considered
  2. Group controls order in which additional codes are tested
  3. Order of rule application can be relevant

- Example:

<table>
<thead>
<tr>
<th>Code</th>
<th>Steering</th>
<th>Group</th>
<th>Supplementing Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM</td>
<td>-</td>
<td>CAA-1030</td>
<td>M111/M605/M611</td>
</tr>
<tr>
<td>GM</td>
<td>L</td>
<td>CAA-1030</td>
<td>M112</td>
</tr>
<tr>
<td>GA</td>
<td>R</td>
<td>CAA-1030</td>
<td>M113</td>
</tr>
</tbody>
</table>
Step 2: Constructibility check

- Interpretation of constructibility rules: $x \rightarrow Cond^C$

  Code $x$ is constructible (valid) in a customer’s order $O$, if $x$’s constructibility rule ($Cond^C$) evaluates to true under $O$

- Notes:
  1. Additional constructibility rules independent of model class
  2. Rules hierarchically organized in positions and variants
  3. For a valid order all codes have to be constructible

- Example:

<table>
<thead>
<tr>
<th>Code</th>
<th>Steering</th>
<th>Pos.</th>
<th>Var.</th>
<th>Constructibility Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>M111</td>
<td>-</td>
<td>60</td>
<td>1</td>
<td>M18/M20//M23+494+-M001</td>
</tr>
<tr>
<td>M111</td>
<td>L</td>
<td>60</td>
<td>2</td>
<td>M20+M001</td>
</tr>
<tr>
<td>423</td>
<td>-</td>
<td>100</td>
<td>1</td>
<td>M111/M605/M611</td>
</tr>
</tbody>
</table>

21.08.2000 Carsten Sinz, University of Tübingen
Step 3: Parts list generation

- Interpretation of code rules $Cond^p \rightarrow p$:
  Part $p$ is contained in a customer’s order $O$, if $p$’s code rule ($Cond^p$) evaluates to true under $O$

- Notes:
  1. Parts list grouped by modules, positions, variants depending on functional and geometrical aspects
  2. Variants of each position are mutually exclusive

- Example:

<table>
<thead>
<tr>
<th>Part</th>
<th>St.</th>
<th>Module</th>
<th>Pos.</th>
<th>Var.</th>
<th>Code Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>124893</td>
<td>-</td>
<td>040522</td>
<td>200</td>
<td>1</td>
<td>221+292+(500/611)</td>
</tr>
<tr>
<td>242488</td>
<td>L</td>
<td>040522</td>
<td>200</td>
<td>2</td>
<td>(800/801)+-704</td>
</tr>
<tr>
<td>486919</td>
<td>R</td>
<td>012400</td>
<td>100</td>
<td>10</td>
<td>M18/M20+M111</td>
</tr>
</tbody>
</table>
BIS: Consistency of Product Documentation

★ Critical points with DIALOG system:
  – Complexity of rules may cause errors
    • unnecessary rejected orders
    • loss of production
    • wrong scheduling of parts
  – Maintaining the rule base is a demanding job

★ Our solution: BIS (Baubarkeits-Informations-System)
  – Add-on tool to check global consistency criteria of the product data-base as a whole
  – Increase the quality of product documentation
  – Deliver assistance in maintaining the product data-base
Global Consistency Criteria

★ A priori criteria: (not requiring special product knowledge)
   – Necessary Codes
   – Inadmissible Codes
   – Consistency of the order completion process
     • order dependency
     • stability (no valid orders are invalidated)
   – Superfluous parts
   – Ambiguities in the parts list

★ User-specified criteria
SAT Encoding of Consistency Assertions

★ Outline of Encoding

1. Generate Boolean formula $B$ describing all supplemented and checked orders

   $B = Z \land C$

   $Z = (Cond^S_1 \Rightarrow x_1) \land \ldots \land (Cond^S_n \Rightarrow x_n)$

   $C = (x_1 \Rightarrow Cond^C_1) \land \ldots \land (x_m \Rightarrow Cond^C_m)$

2. Specify consistency criteria as side condition $S$

3. Check satisfiability of formula $T = B \land S$

★ Example:

Part $p$ with code rule $Cond^P = 221 + 292 + (500/611)$ is superfluous if formula $T = B \land 221 \land 292 \land (500 \lor 611)$ is unsatisfiable.
First Experience with BIS

- Formula $B$ usually contains 200-2000 variables and 10000-100000 symbols (depending on model class).
- Davis-Putnam style satisfiability checkers solve most of the generated SAT-instances in under a second.
- Push-button technology (no user interaction during proofs required).
- Inconsistencies in the DIALOG system data-base could be found.
Architecture of the BIS System

Clients (Java)  Server-Components (C++)

User 1  User 2  User 3

User Layer  Test Layer

Data Layer

Product Database

CORBA

Prover 1  Prover 2  Prover 3
The BIS System Client

21.08.2000 Carsten Sinz, University of Tübingen
Extensions Based on Experience

- **Additional Functionality**
  - Restricting the set of valid orders
  - Valid additional equipment options
  - Combinations of codes
  - Groups of mutually exclusive codes

- **Satisfiability checking without prior CNF-conversion**

- **Extended Propositional Language**
Extended Propositional Language

Special operators for symmetrically related codes

★ General form: \( R_k : X_1, \ldots, X_n \)
  \[ R \in \{ =, \neq, \leq, \lt, \geq, \gt \}, \text{k a positive number,} \]
  \( X_1, \ldots, X_n \) arbitrary formulae of the extended language

★ Example
  \( \leq 1: A, B, C \) is equivalent to \( \neg (A \land B) \land \neg (A \land C) \land \neg (B \land C) \)

★ Advantages
  – More compact notation for symmetrically related codes
  – Pattern occurs frequently in product configuration (e.g. one country code in each order)
  – Specialized algorithms
Summary & Prospects

★ Summary

- BIS complements existing EPDM system
- Increase in product documentation quality
- Global consistency assertions are
  1. converted to Boolean logic satisfiability problems
  2. solved by Davis-Putnam style prover

★ Prospects

- Adaptation to other EPDM systems
- On-line product configuration (E-commerce) requires high-quality EPDM systems with low error rates