Exploiting Different Strategies for the Parallelization of an SMT Solver

Natalia Kalinnik¹, Erika Ábrahám², Tobias Schubert¹, Ralf Wimmer¹ and Bernd Becker¹

¹Institute of Computer Science
Albert-Ludwigs-University of Freiburg
Germany

²Chair of Computer Science II,
RWTH Aachen,
Germany

MBMV 2010
Outline

1 Motivation

2 iSAT – Sequential Interval Constraint Solver

3 Parallelization
   - Search Space Splitting
   - Algorithm Portfolio Design

4 Experimental Results

5 Future Work
Motivation

- Embedded systems are widely used (automotive controllers, air traffic management, medical devices, robots . . . )
- Complexity and safety-critical nature pose a major challenge for verification
- Hybrid systems: discrete + continuous behavior
- Dynamics of hybrid systems may be non-linear (like $x = \sin(y)$)
Satisfiability Modulo Theories (SMT)

- Boolean combinations of non-linear arithmetic constraints

\[(x = \cos(y) \lor y = e^x) \land (x^2 = 2 \cdot y \lor x \geq 0)\]

SMT Formula

- Is the formula satisfiable, i.e. is there an assignment of variables such that the formula evaluates to true?

Problem is undecidable in general
Merges Davis-Putnam-Logemann-Loveland (DPLL) and Interval Constraint Propagation

Manipulation of interval bounds:
\[ x \in [3, 7], \quad y \in [-2, 25] \]

**Deductions:** prune off definite non-solutions

- Unit propagation:
  \[ \cdots \land (x > 8 \lor y = x^2) \land \cdots \]
- Interval Constraint Propagation
Interval Constraint Propagation (ICP)

$x \in [3, 7] \land y \in [-2, 25] \land y = x^2 \quad \Rightarrow \quad x \in [3, 5] \land y \in [9, 25]$
Interval Constraint Solver iSAT

Decisions:
- Split interval in two halves
- Decide in which half to search first
- Propagate resulting information

Learning:
- Deduction can lead to an empty interval $\Rightarrow$ Conflict
- Learn reasons of a conflict in form of conflict clause

Termination:
Stop search when
- All branches lead to conflicts, formula is unsat
- Reasonably small conflict-free box is found
Parallelization based on master/client model and message passing
Algorithmic core of the clients is formed by iSAT
Two variants: Search Space Splitting and Portfolio Approach
PicoSOSSS - Parallelization by the Search Space Splitting

Initial Intervals

\[ x \in [0, 6] \]
\[ y \in [2, 6] \]
\[ z \in [-1, 4] \]

Implication

Decision

\[ x \in (3, 6] \]
\[ x \in [0, 3] \]

Client 1
**Picoso**\textsuperscript{SSS} - Parallelization by the Search Space Splitting

\[ x \in [0, 6] \quad y \in [2, 6] \quad z \in [-1, 4] \]

Initial Intervals

Implication

Decision
\[ x \in (3, 6) \]
\[ x \in [0, 3] \]

Client 2

Client 1
Dynamic work stealing is based on the extended concept of *guiding paths*

A *guiding path* is a sequence of bounds:
- consisting of all decisions and implications and
- a flag, indicating which subproblem still needs to be solved

Each bound with a flag “B” is a candidate for a search space division
Generation of New Subproblems

Initial Intervals

\[ x \in [0, 6] \quad y \in [2, 6] \quad z \in [-1, 4] \]

Implication

\[ \cdots \]

Decision

\[ x \in (3, 6] \quad x \in [0, 3] \]

Client 1

\[ GP_1 = [ (x \geq 0, N), (x \leq 6, N), \\
(x \geq 2, N), (x \leq 6, N), \\
(z \geq -1, N), (z \leq 4, N), \\
(x \leq 3, B) ] \]
Generation of New Subproblems

Initial Intervals

$x \in [0, 6]$

$y \in [2, 6]$

$z \in [-1, 4]$

Implication

Decision

$x \in (3, 6]$

$x \in [0, 3]$

Client 2

Client 1

$GP_1 = \left[ (x \geq 0, \mathbb{N}), (x \leq 6, \mathbb{N}),
(y \geq 2, \mathbb{N}), (y \leq 6, \mathbb{N}),
(z \geq -1, \mathbb{N}), (z \leq 4, \mathbb{N}),
(x \leq 3, \mathbb{N}) \right]$
Generation of New Subproblems

Initial Intervals

$\begin{align*}
\Delta_1 & \Rightarrow \Delta_2 \\
x & \in [0, 6] \\
y & \in [2, 6] \\
z & \in [-1, 4] \\
\end{align*}$

Implication

Decision

$\begin{align*}
x & \in (3, 6] \\
x & \in [0, 3] \\
\end{align*}$

GP$_1 = \begin{cases} 
(x \geq 0, N), (x \leq 6, N), \\
(y \geq 2, N), (y \leq 6, N), \\
(z \geq -1, N), (z \leq 4, N), \\
(x \leq 3, N) \end{cases}$

GP$_2 = \begin{cases} 
(x \geq 0, N), (x \leq 6, N), \\
(y \geq 2, N), (y \leq 6, N), \\
(z \geq -1, N), (z \leq 4, N), \\
(x > 3, N) \end{cases}$
Generation of New Subproblems

Initial Intervals

\[ x \in [0, 6], \quad y \in [2, 6], \quad z \in [-1, 4] \]

Implication

\[ \ldots \]

Decision

\[ x \in (3, 6] \]

Client 2

GP_1 = [ \( (x \geq 0, \mathbb{N}) \), \( (x \leq 6, \mathbb{N}) \), 
\( (y \geq 2, \mathbb{N}) \), \( (y \leq 6, \mathbb{N}) \), 
\( (z \geq -1, \mathbb{N}) \), \( (z \leq 4, \mathbb{N}) \), 
\( (x \leq 3, \mathbb{N}) \) ]

\[ \]

\[ \]

GP_2 = [ \( (x \geq 0, \mathbb{N}) \), \( (x \leq 6, \mathbb{N}) \), 
\( (y \geq 2, \mathbb{N}) \), \( (y \leq 6, \mathbb{N}) \), 
\( (z \geq -1, \mathbb{N}) \), \( (z \leq 4, \mathbb{N}) \), 
\( (x > 3, \mathbb{N}) \) ]

\[ \]

GP'_2 = [ \( (x \leq 6, \mathbb{N}) \), 
\( (y \geq 2, \mathbb{N}) \), \( (y \leq 6, \mathbb{N}) \), 
\( (z \geq -1, \mathbb{N}) \), \( (z \leq 4, \mathbb{N}) \), 
\( (x > 3, \mathbb{N}) \) ]
To speed up the search, information in form of conflict clauses is exchanged.

Small modifications of the heuristic may lead to large changes in computation time.

Examine disjoint parts of the search space with different heuristics.
Solvers with different heuristics are combined in a single portfolio
No search space splitting is performed
The client that finished the search process first terminates the whole portfolio
Conflict clause sharing
Experimental Results

- Four 2.3 GHz AMD processor cores
- One master process and 4 clients
- Clients are allowed to share conflict $\leq 6$
- Computation times for Picoso are the arithmetical mean over 3 runs
Comparing iSAT, *Picoso<sup>PF</sup>* and *Picoso<sup>SSS</sup>*

<table>
<thead>
<tr>
<th></th>
<th>(iSAT_H1)</th>
<th>(iSAT_H2)</th>
<th>(iSAT_H3)</th>
<th>(iSAT_H4)</th>
<th><em>Picoso&lt;sup&gt;SSS&lt;/sup&gt;</em></th>
<th><em>Picoso&lt;sup&gt;PF&lt;/sup&gt;</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>#Solved</td>
<td>49</td>
<td>38</td>
<td>46</td>
<td>55</td>
<td>63</td>
<td>64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solver</th>
<th>(iSAT_H1)</th>
<th>(iSAT_H2)</th>
<th>(iSAT_H3)</th>
<th>(iSAT_H4)</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Picoso&lt;sup&gt;SSS&lt;/sup&gt;</em></td>
<td>8.02</td>
<td>2.89</td>
<td>4.47</td>
<td>7.36</td>
</tr>
<tr>
<td><em>Picoso&lt;sup&gt;PF&lt;/sup&gt;</em></td>
<td>9.39</td>
<td>3.39</td>
<td>5.24</td>
<td>8.62</td>
</tr>
</tbody>
</table>

![Graph showing time (sec) vs. instances solved for *Picoso<sup>SSS</sup>* and *Picoso<sup>PF</sup>*](image-url)
Future Work

- Decision heuristics and parameter settings for portfolio
- Heuristics for conflict clause sharing
- Integration of shared memory concepts
- Combination of Search Space Splitting and Portfolio Approaches
Parallel SMT-solver for the first-order theory of the reals extended with transcendental functions

Two variants: Search Space Splitting and Portfolio Approach

Use of different heuristics and lemma exchange