Abstraction/Refinement
Verification
Algorithms with Backtracking
and Layering

Sharon Barner

12.9.02
Model Checking

- The model is given by a Kripke Structure.
  - Each state is an assignment to all state variables.
- The property is given in Temporal logic.
  - In our case Sugar.

![Diagram of Model Checking](image)
The main problem of model checking is its high memory requirements. We refer to this problem as "the state explosion problem".

A possible approach to overcome the state explosion problem is by using abstraction of the model.

In this work we present an "Abstraction/Refinement" algorithm for always(p) formulas (where p is a boolean expression).
Model Before Abstraction
Abstraction

q and b are abstracted
The model has more behaviors.
The abstract model has less details.
Abstraction

- A model $M'$ is **bigger** than a model $M$ if it contains more behaviors than $M$ ($M < M'$).
- $M'$ is an abstraction of $M$ if it is bigger than $M$ but it is "less detailed".
- Less memory is required in order to represent the abstracted model $M'$.
- If an ACTL formula is true on $M'$ it is also true on $M$ 
  - (Contains all the on-the-fly subset of sugar)
Localization reduction

- Automatic Abstraction/Refinement algorithm.
- Based on the fact that most properties are local
  - Influenced only by the closed environment of the signals in the property.
- Good for verification of large designs.
Reconstruction

Initial states

\[ \rightarrow \rightarrow \rightarrow \rightarrow !p \]
Reconstruction

Initial states
Reconstruction

Initial states
Reconstruction

Initial states
Concrete Counter Example

Initial states

- p
There are cases where there is no concrete counter example which is consistent with the abstract counter example.
Refinement

Split an abstract state in a way that removes the bogus counter example.
Model $M$ has a finite set of variables $V = \{v_1, \ldots, v_n\}$ called state variables.

- A vector of values $(v_1, \ldots, v_n)$ is a state of $M$.

**Projection** of $V' \subseteq V$ of a set $A$ of states, denoted $\text{project}(A, V')$, means the removal of all coordinates not in $V'$ from elements in $A$.

\[
\begin{align*}
V &= \{v_1, v_2, v_3\} & V' &= \{v_1, v_2\} \\
A &= \{ (0,1,0), (0,1,1), (1,1,1) \} & \text{project}(A, V') &= \{ (0,1), (1,1) \}
\end{align*}
\]
Our model abstraction is limited to projection on a subset $V'$ of the state variables.

- the abstract model state variables becomes $V'$ and the variables which influence $V'$ directly.
- the initial state set is projected onto this set, the transition relation is projected, etc.
- The next state functions of the variables that influence directly the signals in $V'$ are left nondeterministic.
Example - Abstraction by projection

\[ M' = \text{project}(M, \text{blue vars}) \]
We choose one abstract counter example.
If we can extend it to counter example in the original model the property failed.
Otherwise we refine the abstraction with the variables which have wrong behavior in the abstract counter example (their behavior is not allowed in the original model).
Example - Abstraction by projection

$M' = \text{project}(M, \text{blue vars})$
Example - Abstraction by projection

Formula

\[ M' = \text{project}(M, \text{blue vars}) \]
Example - Final abstraction

\[ M' = \text{project}(M, \text{blue vars}) \]

No bogus counter example is found - the formula passes
Gradual reconstruction
Gradual reconstruction - layer 1
Gradual reconstruction - layer 2

We found that the counter example is bogus without using the original model.
Example - Abstraction by projection

\[ M' = \text{project}(M, \text{blue vars}) \]
M' = \text{project}(M, \text{blue vars})
$M' = \text{project}(M, \text{blue vars})$
Formula

\[ M' = \text{project}(M, \text{blue vars}) \]
M' = project(M, blue vars)
Final abstraction

Formula

\[ M' = \text{project}(M, \text{blue vars}) \]

No bogus counter example is found - the formula passes
Reconstruction with Abstract State Replacement

Initial states
Reconstruction with Abstract State Replacement

Initial states
Reconstruction with Abstract State Replacement

Initial states
Reconstruction with Abstract State Replacement

Initial states

Different only in the values given to the variables in the free cut. Changing their values does not contradict their behavior in the abstract model.
Reconstruction with Abstract State Replacement

Initial states
Reconstruction with Abstract State Replacement

Initial states
Reconstruction with Backtracking

Initial states
Reconstruction with Backtracking

Only subset of the initial states are selected according to BDD criteria.
Reconstruction with Backtracking

Initial states
Reconstruction with Backtracking

Initial states
Reconstruction with Backtracking

Initial states

There is no edge to the next abstract state.
Reconstruction with Backtracking

Initial states
Reconstruction with Backtracking

Initial states
Reconstruction with Backtracking

Initial states
Reconstruction with Backtracking
Reconstruction with Backtracking

Initial states

Counter example is found
## Results

<table>
<thead>
<tr>
<th>Design Unit</th>
<th>Result</th>
<th>Number of State Vars</th>
<th>No Local</th>
<th>Clarke et al.</th>
<th>Layer+ Alg1</th>
<th>Layer+Alg2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infiniband 1</td>
<td>passed</td>
<td>396(93)</td>
<td>Mem</td>
<td>Mem</td>
<td>54s/43M</td>
<td>137s/47M</td>
</tr>
<tr>
<td>Infiniband 2</td>
<td>passed</td>
<td>377(7)</td>
<td>Mem</td>
<td>* 0.95s/33M</td>
<td>4.32s/33M</td>
<td>4.19s/33M</td>
</tr>
<tr>
<td>Ethernet 1</td>
<td>passed</td>
<td>96(79)</td>
<td>1601s/87M</td>
<td>657s/189M</td>
<td>243s/88M</td>
<td>287s/88M</td>
</tr>
<tr>
<td>Ethernet 2</td>
<td>passed</td>
<td>156(36)</td>
<td>599s/92M</td>
<td>Mem</td>
<td>85s/99M</td>
<td>335s/93M</td>
</tr>
<tr>
<td>Queue CRM</td>
<td>passed</td>
<td>79(70)</td>
<td>148s/45M</td>
<td>75s/42M</td>
<td>34s/41M</td>
<td>28s/41M</td>
</tr>
<tr>
<td>CPU 1</td>
<td>passed</td>
<td>123(65)</td>
<td>Mem</td>
<td>14211s/185M</td>
<td>Mem</td>
<td>9.4s/31M</td>
</tr>
<tr>
<td>CPU 2</td>
<td>failed</td>
<td>105(35)</td>
<td>595s/62M</td>
<td>N/A</td>
<td>405s/50M</td>
<td>229s/50M</td>
</tr>
<tr>
<td>CPU 3</td>
<td>failed</td>
<td>167(66)</td>
<td>11943s/96M</td>
<td>Mem</td>
<td>28963s/192M</td>
<td>1096s/103M</td>
</tr>
</tbody>
</table>

* no refinement was needed