Development of Rabin’s Choice Coordination Algorithm in Event-B

Emre Yilmaz and Thai Son Hoang

Department of Computer Science
Swiss Federal Institute of Technology Zürich (ETH Zürich)

AVoCS’10, 21st-23rd September, 2010
Düsseldorf, Germany
(part of the work is supported by DEPLOY, an FP7 European Project)
Certain v.s. Almost-Certain Termination

- Consider tossing a fair coin $c$ until it comes up head ($H$).

  ```
  while $c = T$ do
  $c : \in \{H, T\}$
  end
  ```

  Demonic non-termination

  ```
  while $c = T$ do
  $c := H \oplus \frac{1}{2} T$
  end
  ```

  Probabilistic termination

- Technique: loop variant on some well-founded order.

- Certain termination: Every iteration must decrease the loop variant.

- Almost-certain termination ([MM05])\(^1\):
  - Every iteration might decrease the loop variant.
  - The variant is bounded above.
  - The probability needs to be proper (bounded away from 0 and 1).

Introduces in [HH07]².

Introduction of probabilistic events.

Behave (almost) the same as standard non-deterministic events, e.g. invariant preservation proof obligations.

Behave differently for convergence proof obligations.

Our Contribution

Questions

- Probabilistic events and Event-B’s developments with refinement?
- How to construct an probabilistic lexicographic variant?

Contribution

- An approach for developing almost-certain termination systems.
  - Extended Rodin Platform for tool support.
  - Formalised Rabin’s Choice Coordination algorithm.
Background. Event-B

- A modelling notation for discrete transition systems.
- Models (machines) contain variables, invariants and events
- Events contain parameters, guards and actions

```
E
  status ordinary/convergent/anticipated
any t where
  G(t, v)
then
  v :| S(t, v, v')
end
```
Convergence in Event-B

- A variant $V(v)$ is proposed.
- The variant must be a finite set or a natural number.
- Every convergent event must decrease the variant.
- Every anticipated event must not increase the variant.
- Combination with refinement: lexicographic variant.
  - Model $M_0$: $E_1$ is convergent and $E_2$ is anticipated with variant $V_1$.
  - Model $M_1$ refines $M_0$: $E_2$ is convergent with variant $V_2$.
  - $(V_1, V_2)$ is a lexicographic variant with $V_1$ has higher precedence.
    \[(V_1, V_2) < (V'_1, V'_2) \iff (V_1 < V'_1) \lor (V_1 = V'_1 \land V_2 < V'_2)\]
The variant $V(v)$ is **bounded above** by a constant $B$.

The event might **decrease** the variant $V(v)$.
Constructing lexicographic variant, e.g. \((V_1, V_2)\):

- Requires refinement.

  - Standard refinement does not preserve almost-certain termination.

  \[
  \begin{array}{l}
  \text{ae} \\
  \quad \text{status} \ probabilistic \\
  \quad \text{any} \ldots \text{where} \\
  \quad \ldots \\
  \quad \text{then} \\
  \quad \quad v : \in \{\text{good, bad}\} \\
  \text{end}
  \end{array}
  \]

  \[
  \begin{array}{l}
  \text{ce} \\
  \quad \text{refines} \ ae \\
  \quad \text{status} \ probabilistic \\
  \quad \text{any} \ldots \text{where} \\
  \quad \ldots \\
  \quad \text{then} \\
  \quad \quad v := \text{bad} \\
  \text{end}
  \end{array}
  \]

- To restrict refinement.

- \((V_1, V_2)\) needs to be bounded above.

  - All sub-variants need to be bounded above.
    (including the variant for proving standard convergence)
Our Approach

Goal

To prove that condition $P$ holds eventually with probability 1 at the end of a program.

The Approach

1. Establish the model of the program contains:
   - an observer event $^a$
     \[ \text{obs} \equiv \text{when } P \text{ then skip end} \]
   - several anticipated events $E_1, \ldots, E_n$.

2. Prove that eventually only obs is enabled:
   - $E_1, \ldots, E_n$ are convergent (either probabilistic or standard).
   - The system is deadlock-free.

Developing Topology Discovery in Event-B. 2009
Choice Coordination Problem

- Given $n$ processes $P_1, \ldots, P_n$.
- Given $k$ alternatives $A_1, \ldots, A_k$.
- Aim: Processes reach a common choice out of the alternatives.
- Constraints: Processes must not communicate directly.

Rabin’s Algorithm

- The protocol uses $k$ shared variables, one for each alternative.
- A process assume to access and modify a shared variable atomically.
- A simplified version of the algorithm by McIver/Morgan with $k = 2$. 
Algorithm Context

<table>
<thead>
<tr>
<th>LEFT</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inside</td>
<td>Outside</td>
</tr>
</tbody>
</table>

E. Yilmaz, T.S. Hoang (ETH-Zürich)  Rabin Choice Coordination in Event-B  AVoCS’10, 21-23/09/10
Algorithm Context

**Motivation and Contribution**

**Background and Approach**

Rabin’s Choice Coordination Algorithm

**Conclusion and Future Work**

---

**Algorithm Context**

```

<table>
<thead>
<tr>
<th>LEFT</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inside</td>
<td></td>
</tr>
<tr>
<td>Outside</td>
<td></td>
</tr>
<tr>
<td>notepad</td>
<td></td>
</tr>
</tbody>
</table>

0
```

---

E. Yilmaz, T.S. Hoang (ETH-Zürich)
Algorithm Context

- LEFT
  - Inside
  - Outside

- RIGHT
  - notice-board

E. Yilmaz, T.S. Hoang (ETH-Zürich)  Rabin Choice Coordination in Event-B  AVoCS'10, 21-23/09/10
**Formal Model. The State**

**variables:**  
lin, rin,  
lout, rout,  
L, R, np

**invariants:**

- **inv0_3**: \( \text{lin} = \emptyset \lor \text{rin} = \emptyset \)
- **inv1_1**: \( \text{partition}(T, \text{lin}, \text{rin}, \text{lout}, \text{rout}) \)
- **inv2_1**: \( L \in \mathbb{N} \)
- **inv2_2**: \( R \in \mathbb{N} \)
- **inv2_3**: \( np \in T \rightarrow \mathbb{N} \)

**init**

begin

lin := \emptyset  
rin := \emptyset  
\( \text{lout, rout :| lout'} = T \setminus \text{rout'} \)
L := 0  
R := 0  
np := T \times \{0\}

end
Algorithm. A Tourist Moves In (First Case)

If there are some tourists inside, he goes in
Algorithm. A Tourist Moves In (Second Case)

If there is no one inside and \( L < n \), he goes in.
Algorithm. A Tourist Alternates (First Case)

If there is no one inside and \( L > n \), he replaces \( n \) by \( L \) on his notepad.
Algorithm. A Tourist Alternates (Second Case)

If there is no one inside and \( L = n \), he first tosses a coin and choose a number \( L' \)
There are two tourists. Assume the tourist on the LEFT has the turn.
Algorithm Intuition

- Conjugate of an even number $n$ is $n + 1$.
- Conjugate of an odd number $n$ is $n - 1$.
- The algorithm contains several rounds.
- In each round, each notice board is chosen probabilistically in the next pair.
- The algorithm terminates when the values of the notice boards are different in the same round.
Algorithm Intuition

- Conjugate of an even number $n$ is $n + 1$.
- Conjugate of an odd number $n$ is $n - 1$.
- The algorithm contains several rounds.

In each round, each notice board is chosen probabilistically in the next pair.

The algorithm terminates when the values of the notice boards are different in the same round.
Algorithm Intuition

- Conjugate of an even number \( n \) is \( n + 1 \).
- Conjugate of an odd number \( n \) is \( n - 1 \).
- The algorithm contains several rounds.
- In each round, each notice board is chosen probabilistically in the next pair.
- The algorithm terminates when the values of the notice boards are different in the same round.
Algorithm Intuition

- Conjugate of an even number $n$ is $n + 1$.
- Conjugate of an odd number $n$ is $n - 1$.
- The algorithm contains several rounds.
- In each round, each notice board is chosen probabilistically in the next pair.
- The algorithm terminates when the values of the notice boards are different in the same round.

<table>
<thead>
<tr>
<th>Round 0</th>
<th>Round 1</th>
<th>Round 2</th>
<th>Round 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>L 2 R</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Conjugate of an even number $n$ is $n + 1$.
Conjugate of an odd number $n$ is $n - 1$.

The algorithm contains several rounds.
In each round, each notice board is chosen probabilistically in the next pair.
The algorithm terminates when the values of the notice boards are different in the same round.
Algorithm Intuition

- Conjugate of an even number $n$ is $n + 1$.
- Conjugate of an odd number $n$ is $n - 1$.
- The algorithm contains several rounds.
- In each round, each notice board is chosen probabilistically in the next pair.
- The algorithm terminates when the values of the notice boards are different in the same round.

| Round 0 | 0 |
| Round 1 | 1 |
| Round 2 | L 4 3 |
| Round 3 | 7 6 5 R |

...
Refinement Strategy

- **Initial model:** introduce the set of *tourists inside*: \( \text{lin} \) and \( \text{rin} \).

- **1st Ref.:** introduce the set of *tourists outside*: \( \text{lout} \) and \( \text{prout} \).

- **2nd Ref.:** introduce **Rabin’s algorithm**
  including the noticeboards \( (L, R) \) and tourists’ notepads \( (np) \).

- **3rd–6th Refs.:** prove convergence property.
  - A lexicographic variant with 2 layers [MM05].
  - We used both finite set and natural number variants.
  - Split and merge of events: Simpler proofs.

- **7th Ref.:** prove **deadlock-freeness**.
Refinement Strategy

- **Initial** model: introduce the set of **tourists inside**: $lin$ and $rin$.

- **1st** Ref.: introduce the set of **tourists outside**: $lout$ and $prout$.

- **2nd** Ref.: introduce **Rabin’s algorithm** including the **noticeboards** ($L$, $R$) and tourists’ **notepads** ($np$).

- **3rd–6th** Refs.: prove **convergence** property.
  - A lexicographic variant with 2 layers [MM05].
  - We used both **finite set** and **natural number** variants.
  - **Split** and **merge** of events: Simpler proofs..

- **7th** Ref.: prove **deadlock-freeness**.
Proof Statistics

<table>
<thead>
<tr>
<th>Model</th>
<th>Total</th>
<th>Auto. (%)</th>
<th>Man. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial model</td>
<td>6</td>
<td>6(100%)</td>
<td>0(N/A)</td>
</tr>
<tr>
<td>1st Refinement</td>
<td>8</td>
<td>7(88%)</td>
<td>1(12%)</td>
</tr>
<tr>
<td>2nd Refinement</td>
<td>63</td>
<td>49(78%)</td>
<td>14(23%)</td>
</tr>
<tr>
<td>Outer variant</td>
<td>54</td>
<td>29(54%)</td>
<td>25(46%)</td>
</tr>
<tr>
<td>Inner variant</td>
<td>11</td>
<td>8(73%)</td>
<td>3(27%)</td>
</tr>
<tr>
<td>Deadlock freedom</td>
<td>4</td>
<td>0(0%)</td>
<td>4(100%)</td>
</tr>
<tr>
<td>Total</td>
<td>146</td>
<td>99(68%)</td>
<td>47(32%)</td>
</tr>
</tbody>
</table>
Conclusion

- An approach for developing almost-certain termination programs.
  - probabilistic lexicographic variant.
  - Practical tool support.

Future work

- Improve tool support.
- Verify other examples, e.g. IEEE1394 protocol.
- Elaborate refinement while preserving probabilistic convergence.
For Further Reading I

J.-R. Abrial. 

C. Morgan, A. McIver. 

S. Hallerstede, T. Hoang. 
Qualitative Probabilistic Modelling in Event-B. 

Developing topology discovery in Event-B. 