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).

Demonic non-termination	Probabilistic termination
while $c = T$ **do** $c :\in \{H, T\}$ **end**	**while** $c = T$ **do** $c := H \oplus_{1/2} T$ **end**

- 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).

[1][MM05] C. Morgan, A. McIver.
Abstraction, Refinement and Proof for Probabilistic Systems. 2005.

Qualitative Reasoning in Event-B

- Introduced 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.

[2][HH07] S. Hallerstede, T. Hoang.
Qualitative Probabilistic Modelling in Event-B. In *iFM 2007*

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.

$$(V_1, V_2) < (V_1', V_2') \quad \Leftrightarrow \quad (V_1 < V_1') \lor (V_1 = V_1' \land V_2 < V_2')$$

Probabilistic Events in Event-B

```
E
  status  probabilistic
  any  t  where
    G(t, v)
  then
    v :| S(t, v, v')
  end
```

- The variant $V(v)$ is bounded above by a constant B.
- The event might decrease the variant $V(v)$.

Probabilistic Lexicographic Variant

Constructing lexicographic variant, e.g. (V_1, V_2):

- Requires refinement.
  - Standard refinement does not preserve almost-certain termination.

```
ae
  status  probabilistic
  any  ...  where
    ...
  then
    v :∈ {good, bad}
  end
```

```
ce
  refines  ae
  status  probabilistic
  any  ...  where
    ...
  then
    v := bad
  end
```

  - 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]

     obs $\widehat{=}$ **when** P **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.

[a] [HKBA09] T.S. Hoang, H. Kuruma, D. Basin and J-R. Abrial. Developing Topology Discovery in Event-B. 2009

Choice Coordination Problem and Rabin's Algorithm

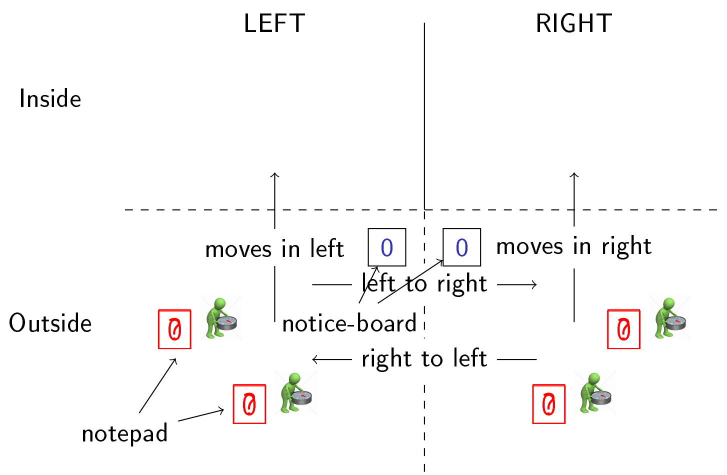
Choice Coordination Problem

- 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



Formal Model. The State

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

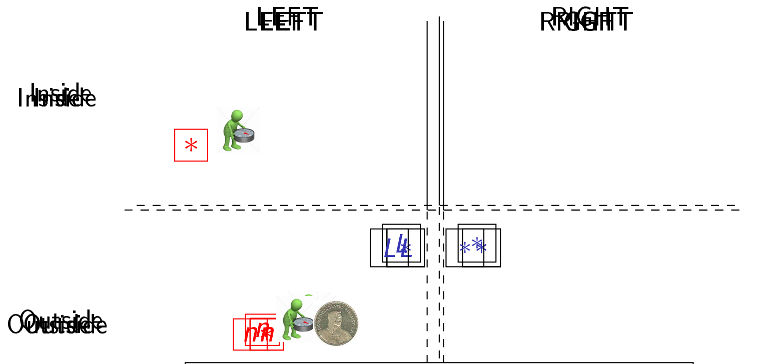
invariants:
- **inv0_3 :** $lin = \varnothing \lor rin = \varnothing$
- **inv1_1 :** $partition(T, lin, rin, lout, rout)$
- **inv2_1 :** $L \in \mathbb{N}$
- **inv2_2 :** $R \in \mathbb{N}$
- **inv2_3 :** $np \in T \to \mathbb{N}$

init
begin
$lin := \varnothing$
$rin := \varnothing$
$lout, rout :| lout' = T \setminus rout'$
$L := 0$
$R := 0$
$np := T \times \{0\}$
end

Algorithm. A Tourist Moves In (First Case)

Algorithm. A Tourist Moves In (Second Case)

Algorithm. A Tourist Alternates (First Case)

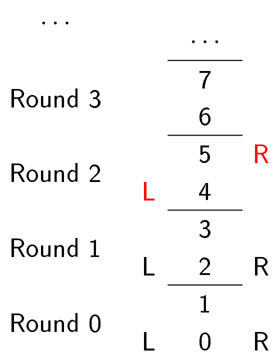


If there is no one inside and $L = n$,
he first tosses a coin and choose a number L'

Algorithm. A Tourist Alternates (Second Case)

Animation with Two Tourists

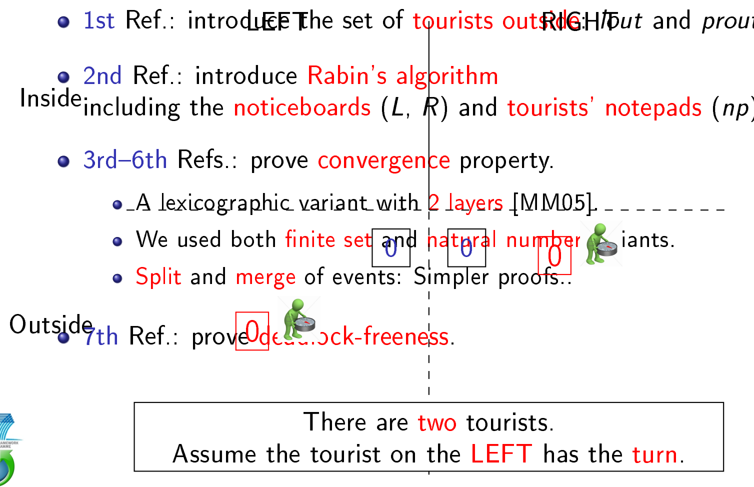
Algorithm Intuition



- 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.

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

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

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.
Modeling in Event-B: System and Software Engineering.
Cambridge University Press, May 2010.

C. Morgan, A. McIver.
Abstraction, Refinement and Proof for Probabilistic Systems.
Springer Verlag, 2005.

S. Hallerstede, T. Hoang.
Qualitative Probabilistic Modelling in Event-B.
In David and Gibbons (eds.), *IFM 2007: Integrated Formal Methods.*
LNCS 4591, pp. 293–312. Springer Verlag, Oxford, U.K., July 2007.

T. Hoang, H. Kuruma, D. Basin, J.-R. Abrial.
Developing topology discovery in Event-B.
Sci. Comput. Program. 74(11-12):879–899, 2009.