The Hanoi Omega-Automata Format

Tomáš Babiak[1] František Blahoudek[1]
Alexandre Duret-Lutz[2] Joachim Klein[3] Jan Křetínský[5]
David Müller[3] David Parker[4] Jan Strejček[1]

[1]Faculty of Informatics, Masaryk University, Brno, Czech Republic
[2]LRDE, EPITA, Le Kremlin-Bicêtre, France
[3]Technische Universität Dresden, Germany
[4]University of Birmingham, UK
[5]IST Austria

CAV'15, July 23

Motivation

- there exist many tools that I/O ω-automata with various acceptance conditions (Büchi, Rabin, Streett, Parity...)
- missing an **interchange format for ω-automata**

Motivation

- there exist many tools that I/O ω-automata with various acceptance conditions (Büchi, Rabin, Streett, Parity...)
- missing an **interchange format for ω-automata**

- evidence that generalized acceptance conditions are useful e.g., orders-of-magnitude speed-up for probabilistic LTL model checking using generalized Rabin acceptance
  [Chatterjee et al., CAV'13]
- need support for **generic acceptance conditions**

Motivation

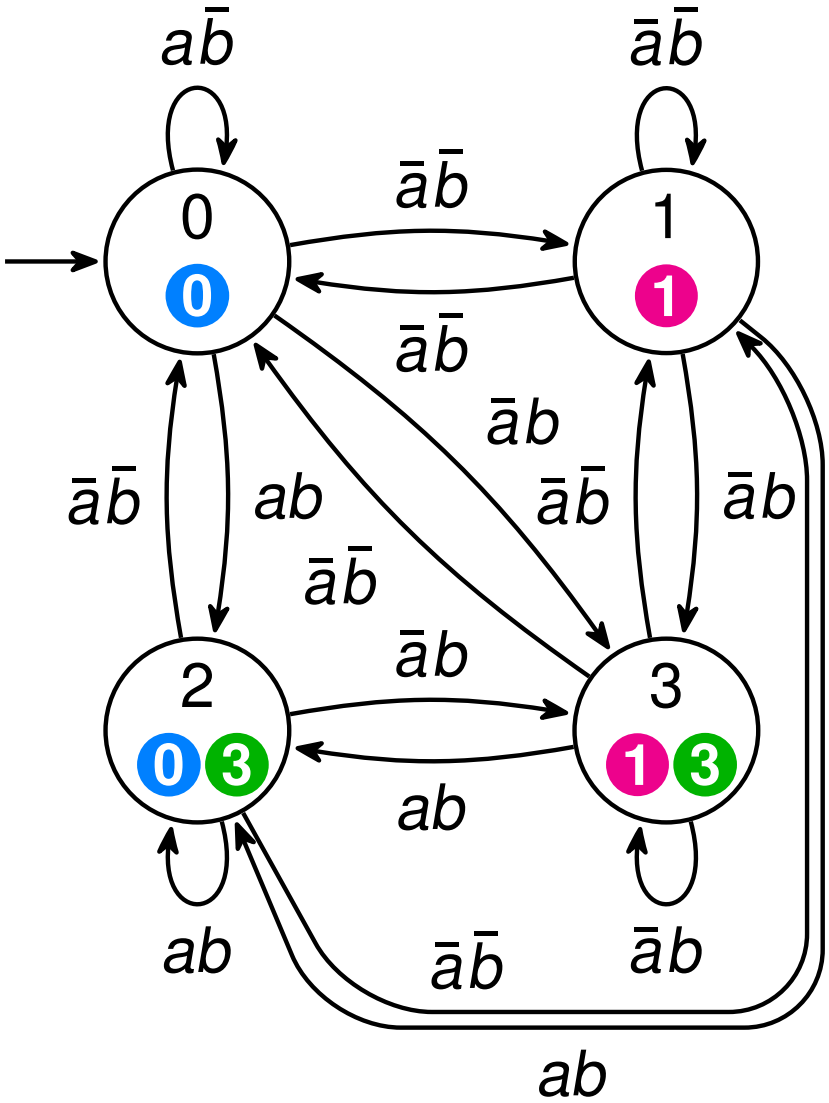
- there exist many tools that I/O ω-automata with various acceptance conditions (Büchi, Rabin, Streett, Parity...)
- missing an **interchange format for ω-automata**

- evidence that generalized acceptance conditions are useful e.g., orders-of-magnitude speed-up for probabilistic LTL model checking using generalized Rabin acceptance
  [Chatterjee et al., CAV'13]
- need support for **generic acceptance conditions**

- support for a wide range of other features (alternation, transition-based acceptance, streaming...)

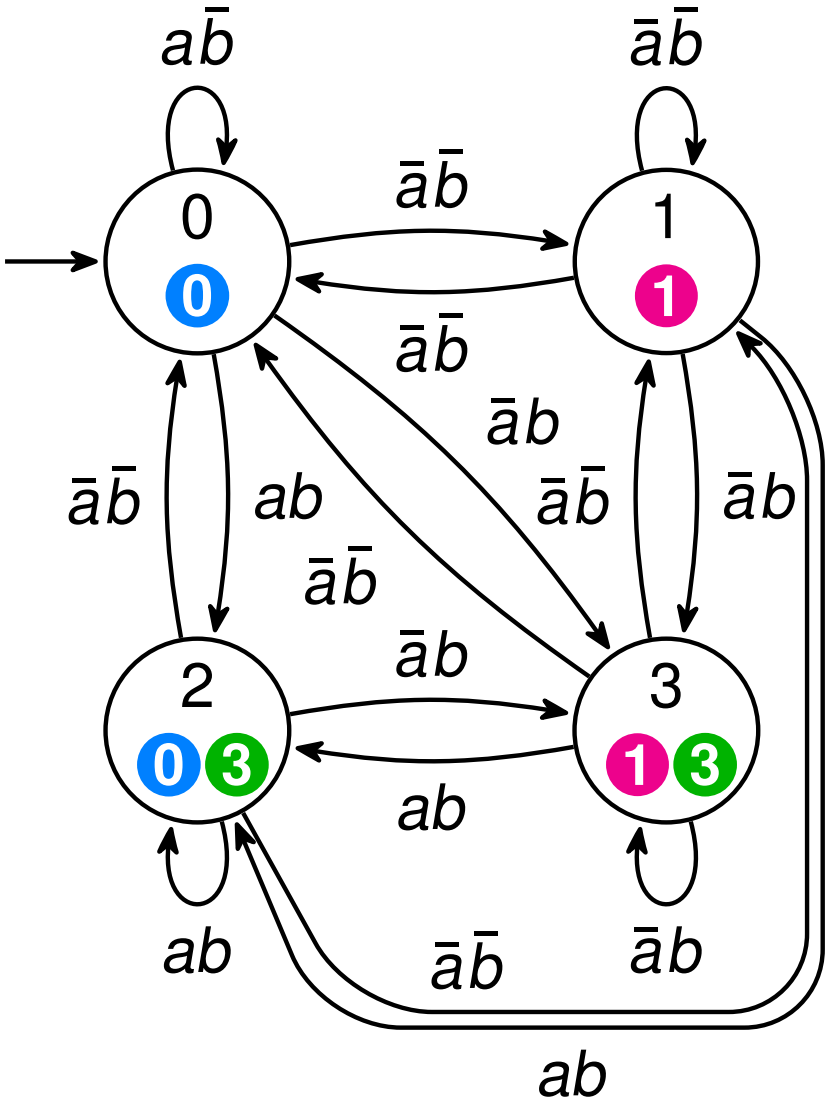
A Rabin automaton for $\mathbf{G}\,\mathbf{F}\,a \rightarrow \mathbf{G}\,\mathbf{F}\,b$

$(\mathrm{Fin}(0) \wedge \mathrm{Inf}(1)) \vee (\mathrm{Fin}(2) \wedge \mathrm{Inf}(3))$



A Rabin automaton for $\mathbf{G}\,\mathbf{F}\,a \rightarrow \mathbf{G}\,\mathbf{F}\,b$

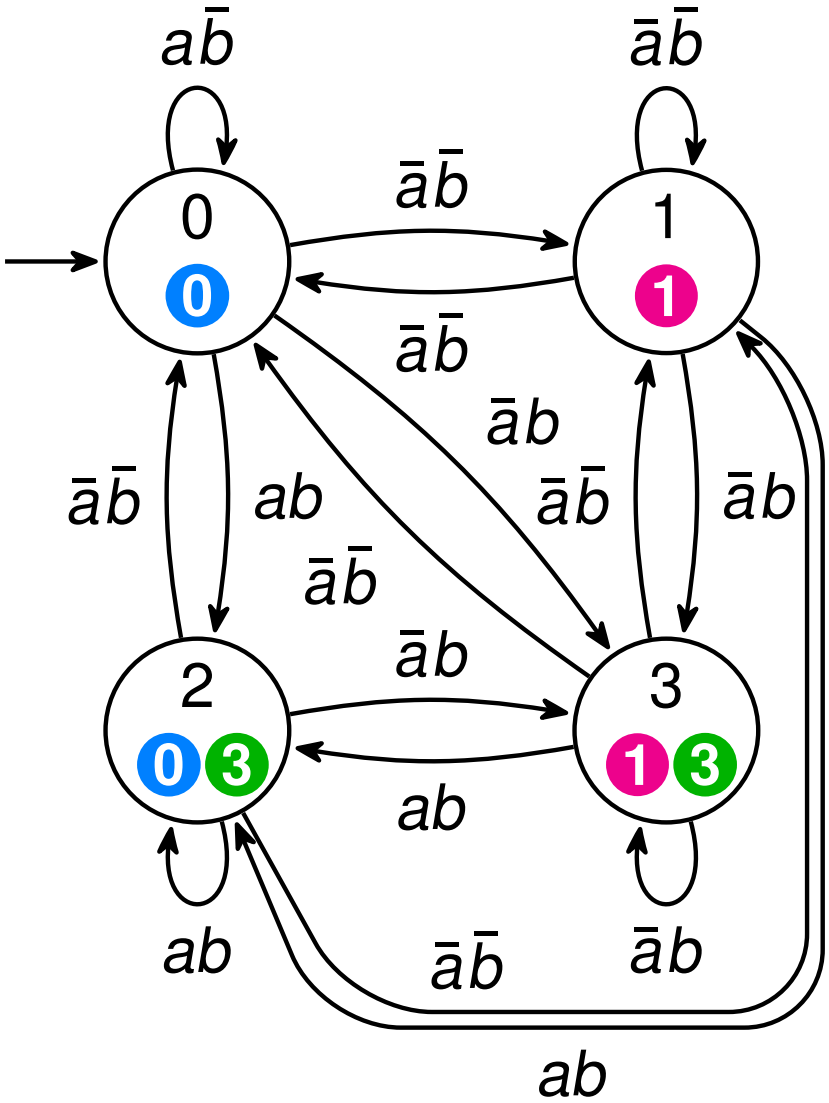
$$(\mathrm{Fin}(0) \wedge \mathrm{Inf}(1)) \vee (\mathrm{Fin}(2) \wedge \mathrm{Inf}(3))$$




```
HOA: v1
States: 4
Start: 0
AP: 2 "a" "b"
acc-name: Rabin 2
Acceptance: 4
  Fin( 0 )&Inf( 1 )|Fin( 2 )&Inf( 3 )
--BODY--
State: 0 { 0 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
State: 1 { 1 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
State: 2 { 0   3 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
State: 3 { 1   3 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
--END--
```

A Rabin automaton for $\mathbf{G}\,\mathbf{F}\,a \rightarrow \mathbf{G}\,\mathbf{F}\,b$

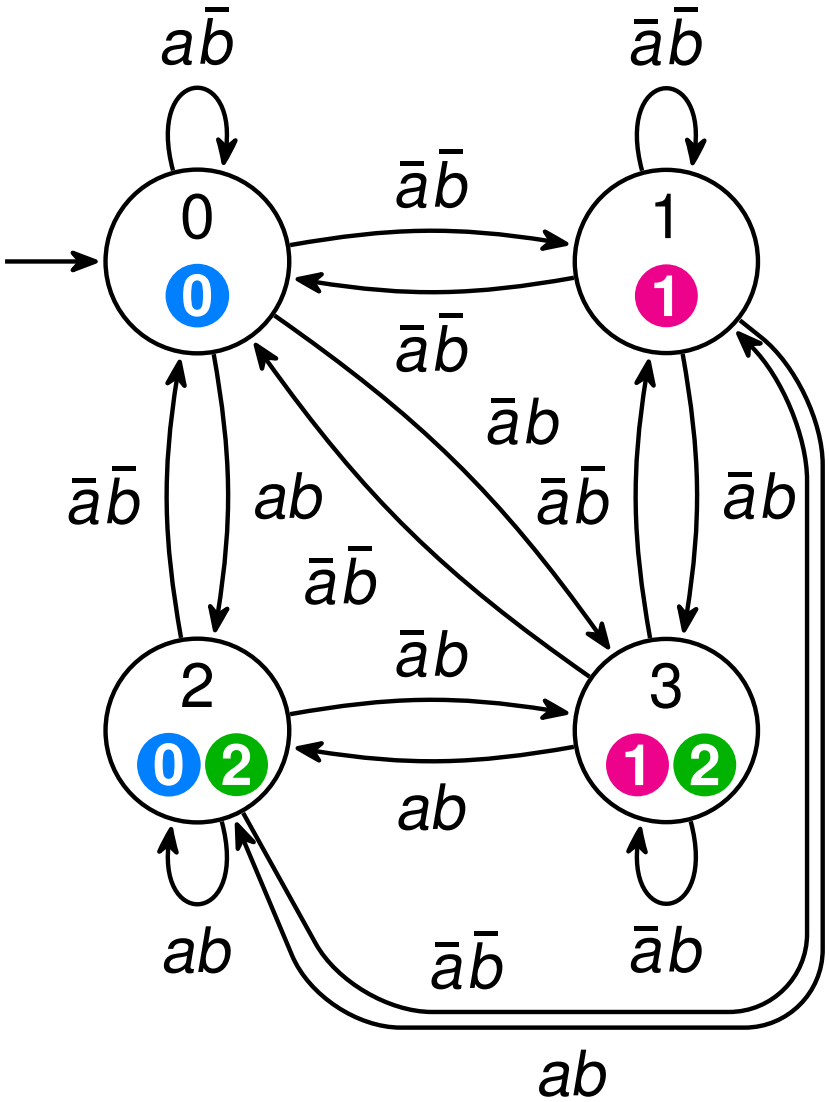
$(\mathrm{Fin}(0) \wedge \mathrm{Inf}(1)) \vee (\mathrm{Fin}(2) \wedge \mathrm{Inf}(3))$




```
HOA: v1
States: 4
Start: 0
AP: 2 "a" "b"
acc-name: Rabin 2
Acceptance: 4
  Fin( 0 )&Inf( 1 )|Fin( 2 )&Inf( 3 )
--BODY--
State: 0 { 0 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
State: 1 { 1 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
State: 2 { 0  3 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
State: 3 { 1  3 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
--END--
```

An ω-automaton for $\mathbf{G}\,\mathbf{F}\,a \rightarrow \mathbf{G}\,\mathbf{F}\,b$

$(\mathrm{Fin}(0) \wedge \mathrm{Inf}(1)) \vee \mathrm{Inf}(2)$

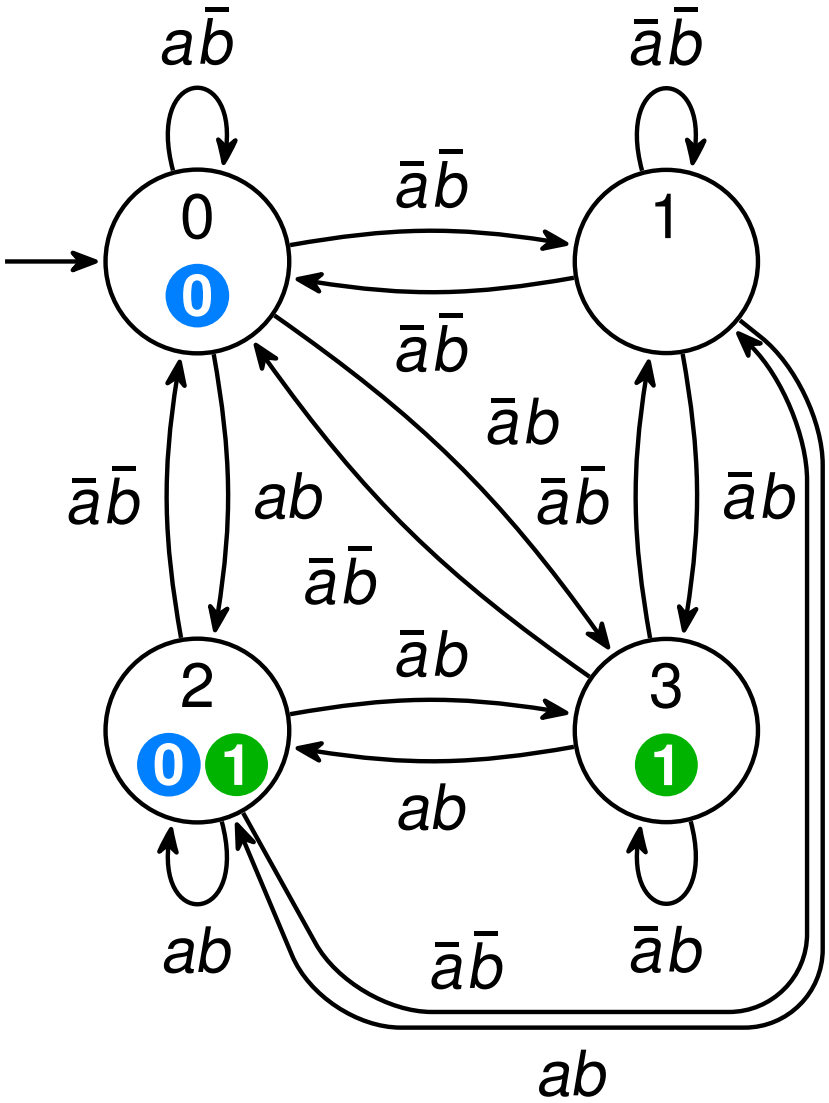



```
HOA: v1
States: 4
Start: 0
AP: 2 "a" "b"

Acceptance: 3
  Fin( 0 )&Inf( 1 )|Inf( 2 )
--BODY--
State: 0 { 0 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
State: 1 { 1 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
State: 2 { 0  2 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
State: 3 { 1  2 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
--END--
```

A Streett automaton for $\mathbf{G}\,\mathbf{F}\,a \rightarrow \mathbf{G}\,\mathbf{F}\,b$

Fin(0)∨Inf(1)




```
HOA: v1
States: 4
Start: 0
AP: 2 "a" "b"
acc-name: Streett 1
Acceptance: 2
  Fin( 0 )|Inf( 1 )
--BODY--
State: 0 { 0 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
State: 1
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
State: 2 { 0  1 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
State: 3 { 1 }
[!0&!1] 1 [0&!1] 0 [!0&1] 3 [0&1] 2
--END--
```

Tool Support

`http://adl.github.io/hoaf/support.html`

`ltl2dstar 0.5.3` input BA, output DRA or DSA

`ltl3ba 1.1.2` output BA, TGBA, or VWAA

`ltl3dra 0.2.2` output DRA, TGDRA or MMAA

`Rabinizer 3.1` output DRA, TDRA, GDRA, or TGDRA

`PRISM 4.3`
: input deterministic automata for probabilistic model checking; (generalized) Rabin for MDP; any acceptance for CTMC/DTMC

`Spot 1.99.2` (tool suite)
: can input/output anything that is not alternating; can convert from never claims or LBTT; has several transformations

`jhoafparser` and `cpphoafparser`
: two parsers with pretty printers, and convenient transformations