A Rabin automaton for $\mathbf{GF} a \rightarrow \mathbf{GF} b$

An alternating co-Büchi automaton for $\mathbf{GF} a \rightarrow \mathbf{GF} b$

A transition-based Streett automaton for $\mathbf{GF} a \rightarrow \mathbf{GF} b$

The Hanoi Omega-Automata Format

Tomáš Babiak¹, František Blahoudek¹, Alexandre Duret-Lutz², Joachim Klein³, Jan Křetínský⁴, David Müller⁵, David Parker¹, and Jan Strejček¹

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

---

**Generic acceptance**

Acceptance: $n$. acc specifies the acceptance condition using the following grammar:

$$
\text{acc} ::= f \mid t \mid \text{Inf}(s) \mid \text{Inf}(s) \mid \text{Fin}(s) \mid \text{Fin}(s) \mid \text{acc} \& \text{acc} \mid \text{acc} \mid \text{acc} \mid \text{acc} \\
$$

Where $s$ is an accepting set number smaller than $n$, $\neg s$ denotes the complement of that set. $\text{Fin}(\text{Inf})$ is satisfied when the set is visited finitely (resp. infinitely) often by a run.

For alternating automata all branches of a run-tree have to satisfy the condition.

**Known acceptance conditions** can be named with the optional acc-name: header.

Of course acceptance conditions can be created as needed, they do not require a name.

---

**Batch processing**

The --END-- marker allows multiple automata to be chained and be batch-processed by a pipe of several commands.

---

**Trivia**

Work on this format started during the ATVA'13 conference in Hanoi (Vietnam). Hence the name.

---

**Open development**

The format is developed on GitHub at https://github.com/adl/hoaf.

Feel free to make suggestions or report bugs on the issue tracker.

---

**Tool support**

**ltl2dstar 0.5.3**: creates deterministic automata from LTL or Büchi automata inputs BA, outputs DRA or DSA.

**ltl2ba 1.1.2**: creates automata from LTL inputs BA, TGBA, or VWAA.

**ltl3dra 0.2.2**: creates deterministic automata from (a subset of) LTL inputs DRA, TGMDA, or MMDA.

Rabinizer 3: creates deterministic automata from LTL outputs DRA, TGMDA, or MMDA.

PRISM 4.3: probabilistic LTL model checking using deterministic HOA automata; (generalized) Rabin for MDP, any acceptance for CTMC/DTMC; scripts for interfacing with the tools above.

Spot 1.99.1: tool suite for LTL/PILA and automata manipulation can input/output anything that is not alternating; translates between formats (like never claim or LBTT); has several automata transformations; to easily develop new consumer tools; jhoafparser/cpphoafparser: Java and C++ parser libraries with pretty printers, translators, and forwarding, to easily choreograph between formats.

If you implement HOA support, tell us so we can list your tool there.

---

**Acc types**

- **Buchi**
- **Rabin**
- **Streett**
- **Generalized-Buchi**
- **Generalized-Rabin**
- **Parity**
- **Monitor**
- **Automatic**

---

**Known acceptance conditions**

- **Buchi**
- **Rabin**
- **Streett**
- **Generalized-Buchi**
- **Generalized-Rabin**
- **Parity**
- **Monitor**
- **Automatic**

---

**References**

Chatterjee et al. (CAV'13) observed order of magnitude speedups replacing Rabin acceptance by generalized Rabin for probabilistic model checking with PRISM.

---

**Contact**

For questions, comments, or contributions, please contact us at babiak@fi.muni.cz, f.blahoudek@epita.fr, duret-lutz@irif.fr, joachim.klein@tu-dresden.de, david.muller@tu-dresden.de, david.parker@masaryk-university.cz, jan.strejcek@masaryk-university.cz.