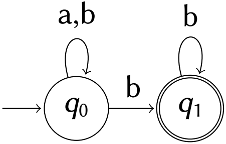
Determinization and Ambiguity of Classical and Probabilistic Büchi Automata

Anton Pirogov

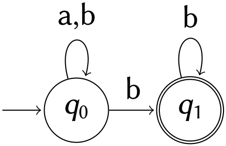
18.02.2021
PhD defense talk

UnRAVeL
RWTH AACHEN UNIVERSITY
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

Büchi automata



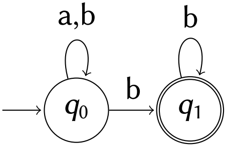
Büchi automata



$$
\begin{aligned}
\Box((call \lor \Diamond open) \to & ((\neg atfloor \lor \neg open)\, \mathcal{U} \\
& (open \lor ((atfloor \land \neg open)\, \mathcal{U} \\
& (open \lor ((\neg atfloor \land \neg open)\, \mathcal{U} \\
& (open \lor ((atfloor \land \neg open)\, \mathcal{U} \\
& (open \lor (\neg atfloor\, \mathcal{U}\, open))))))))))
\end{aligned}
$$

have connections to logic

Büchi automata



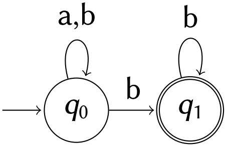

```
// Quellcodebeispiel in C++

#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int alter; // Variable vom Typ Integer

    cout << "Wie alt bist du?";
    cin >> alter;
    cout << "Du bist " << alter << " Jahre alt" << endl;
    getc();
    return 0;
}
```

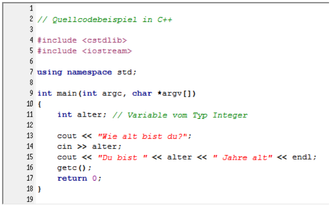
$$\overset{?}{\models}$$

$$\begin{aligned}\Box((call \vee \Diamond open) \rightarrow ((\neg atfloor \vee \neg open)\, \mathcal{U}\, & (open \vee ((atfloor \wedge \neg open)\, \mathcal{U} \\ & (open \vee ((\neg atfloor \wedge \neg open)\, \mathcal{U} \\ & (open \vee ((atfloor \wedge \neg open)\, \mathcal{U} \\ & (open \vee (\neg atfloor\, \mathcal{U}\, open))))))))))\end{aligned}$$

have connections to logic , model checking

Büchi automata





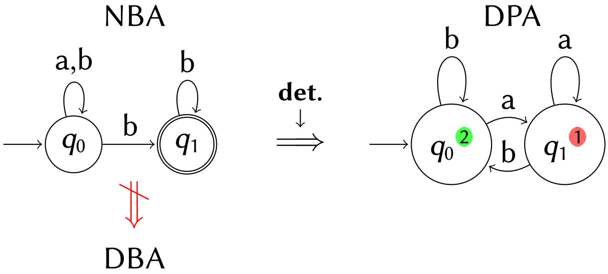
```
// Quellcodebeispiel in C++

#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int alter; // Variable vom Typ Integer

    cout << "Wie alt bist du?";
    cin >> alter;
    cout << "Du bist " << alter << " Jahre alt" << endl;
    getc();
    return 0;
}
```

synthesize $\Longleftarrow$

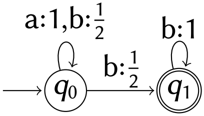
$$\Box((\mathit{call} \vee \Diamond \mathit{open}) \rightarrow ((\neg \mathit{atfloor} \vee \neg \mathit{open})\, \mathcal{U} \\ (\mathit{open} \vee ((\mathit{atfloor} \wedge \neg \mathit{open})\, \mathcal{U} \\ (\mathit{open} \vee ((\neg \mathit{atfloor} \wedge \neg \mathit{open})\, \mathcal{U} \\ (\mathit{open} \vee ((\mathit{atfloor} \wedge \neg \mathit{open})\, \mathcal{U} \\ (\mathit{open} \vee (\neg \mathit{atfloor}\, \mathcal{U}\, \mathit{open}))))))))))$$

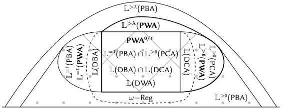
have connections to logic , model checking , synthesis , ...

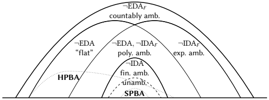
Determinization of nondeterministic Büchi automata



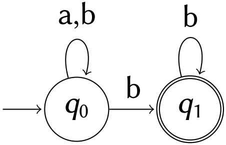
Subclasses of probabilistic Büchi automata



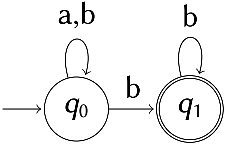




Büchi automata are ω-automata, i.e.
finite automata that read **infinite** words

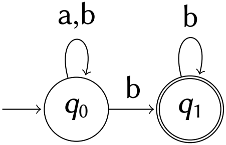


Büchi automata are ω-automata, i.e.
finite automata that read **infinite** words



- $\mathcal{A} = (\Sigma, Q, \Delta, q_0, F) \leftarrow$ just NFA with different semantics
- **Büchi**: some $q \in F$ seen *infinitely often* along a run
- **Co-Büchi**: all $q \in F$ seen *finitely often* along a run

Büchi automata are ω-automata, i.e.
finite automata that read **infinite** words

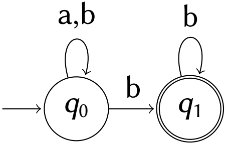


As NFA: $L(\mathcal{A}) = (a+b)^* b$

As NBA: $L(\mathcal{A}) = (a+b)^* b^\omega$

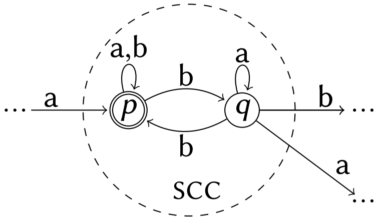
- $\mathcal{A} = (\Sigma, Q, \Delta, q_0, F) \leftarrow$ just NFA with different semantics
- **Büchi**: some $q \in F$ seen *infinitely often* along a run
- **Co-Büchi**: all $q \in F$ seen *finitely often* along a run

Büchi automata are ω-automata, i.e.
finite automata that read **infinite** words

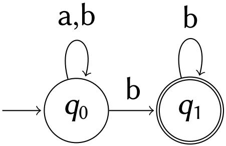


As NFA: $L(\mathcal{A}) = (a+b)^* b$
As NBA: $L(\mathcal{A}) = (a+b)^* b^\omega$

- $\mathcal{A} = (\Sigma, Q, \Delta, q_0, F) \leftarrow$ just NFA with different semantics
- **Büchi**: some $q \in F$ seen *infinitely often* along a run
- **Co-Büchi**: all $q \in F$ seen *finitely often* along a run
- Some **SCC** of $\mathcal{A}$ can be:

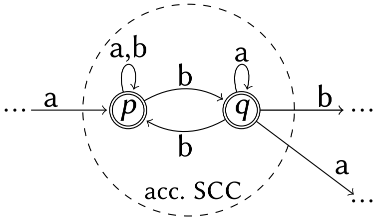


Büchi automata are ω-automata, i.e.
finite automata that read **infinite** words

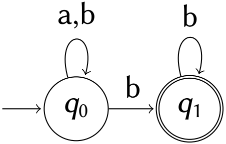


As NFA: $L(\mathcal{A}) = (a+b)^* b$
As NBA: $L(\mathcal{A}) = (a+b)^* b^\omega$

- $\mathcal{A} = (\Sigma, Q, \Delta, q_0, F) \leftarrow$ just NFA with different semantics
- **Büchi**: some $q \in F$ seen *infinitely often* along a run
- **Co-Büchi**: all $q \in F$ seen *finitely often* along a run
- Some **SCC** of $\mathcal{A}$ can be:
  acc.

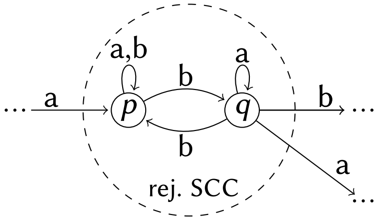


Büchi automata are ω-automata, i.e.
finite automata that read **infinite** words

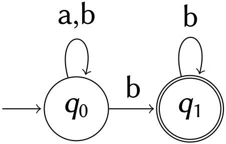


As NFA: $L(\mathcal{A}) = (a+b)^* b$
As NBA: $L(\mathcal{A}) = (a+b)^* b^\omega$

- $\mathcal{A} = (\Sigma, Q, \Delta, q_0, F) \leftarrow$ just NFA with different semantics
- **Büchi**: some $q \in F$ seen *infinitely often* along a run
- **Co-Büchi**: all $q \in F$ seen *finitely often* along a run
- Some **SCC** of $\mathcal{A}$ can be:
  acc. **rej.**

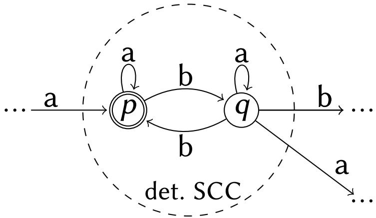


Büchi automata are ω-automata, i.e.
finite automata that read **infinite** words

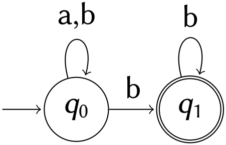


As NFA: $L(\mathcal{A}) = (a+b)^* b$
As NBA: $L(\mathcal{A}) = (a+b)^* b^\omega$

- $\mathcal{A} = (\Sigma, Q, \Delta, q_0, F) \leftarrow$ just NFA with different semantics
- **Büchi**: some $q \in F$ seen *infinitely often* along a run
- **Co-Büchi**: all $q \in F$ seen *finitely often* along a run
- Some **SCC** of $\mathcal{A}$ can be:
  acc. **rej.** **det.** **...**

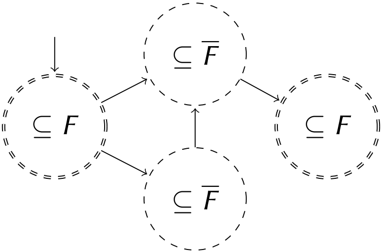


Büchi automata are ω-automata, i.e.
finite automata that read **infinite** words



As NFA: $L(\mathcal{A}) = (a+b)^* b$
As NBA: $L(\mathcal{A}) = (a+b)^* b^\omega$

- $\mathcal{A} = (\Sigma, Q, \Delta, q_0, F) \leftarrow$ just NFA with different semantics
- **Büchi**: some $q \in F$ seen *infinitely often* along a run
- **Co-Büchi**: all $q \in F$ seen *finitely often* along a run
- Some **SCC** of $\mathcal{A}$ can be:
  acc. **rej.** **det.** **...**
- An automaton is **weak**,
  if all SCCs are acc. or rej.



Introduction

NBA Determinization
- Safra Construction
- Muller-Schupp Construction
- Comparison of Safra and Muller-Schupp
- A Sketch of the Unified Construction
- Optimizations and Heuristics

Characterization of PBA subclasses
- Ambiguity of NBA
- Probabilistic Büchi Automata
- Weak PBA
- Ambiguity-restricted PBA
- Complexity results

Summary

- Classical synthesis:

$$\text{LTL} \xrightarrow{\textit{trans.}} \text{NBA} \xrightarrow{\dots} \text{parity game} \xrightarrow{\textit{solve}} \text{FSM}$$

- Classical synthesis:

$$\mathsf{LTL} \xrightarrow{\textit{trans.}} \mathsf{NBA} \xrightarrow{\cdots} \text{parity game} \xrightarrow{\textit{solve}} \mathsf{FSM}$$

- *probabilistic* model checking: e.g. $\mathcal{S}$ is Markov chain.
  A possible question: Is $P(\mathcal{S} \models \varphi) = 1$?

- NBA unsuitable: $\overset{\text{unrestricted}}{\text{nondeterminism}} \rightarrow ↯ \leftarrow$ probabilities

- Classical synthesis:

$$\text{LTL} \xrightarrow{\textit{trans.}} \text{NBA} \xrightarrow{\dots} \text{parity game} \xrightarrow{\textit{solve}} \text{FSM}$$

- *probabilistic* model checking: e.g. $\mathcal{S}$ is Markov chain.
  A possible question: Is $P(\mathcal{S} \models \varphi) = 1$?

- NBA unsuitable: $\overset{\text{unrestricted}}{\text{nondeterminism}} \rightarrow \lightning \leftarrow$ probabilities

- **classical solution**:
  determinize the NBA

- Classical synthesis:

$$\text{LTL} \xrightarrow{\textit{trans.}} \text{NBA} \xrightarrow{\cdots} \text{parity game} \xrightarrow{\textit{solve}} \text{FSM}$$

- *probabilistic* model checking: e.g. $\mathcal{S}$ is Markov chain. A possible question: Is $P(\mathcal{S} \models \varphi) = 1$?

- NBA <u>unsuitable</u>: nondeterminism $\rightarrow \lightning \leftarrow$ probabilities (unrestricted)

- **classical solution**:

  determinize the NBA to ~~DBA~~. impossible in general!

- Classical synthesis:

$$\text{LTL} \xrightarrow{\textit{trans.}} \text{NBA} \xrightarrow{\dots} \text{parity game} \xrightarrow{\textit{solve}} \text{FSM}$$

- *probabilistic* model checking: e.g. $\mathcal{S}$ is Markov chain.
  A possible question: Is $P(\mathcal{S} \models \varphi) = 1$?

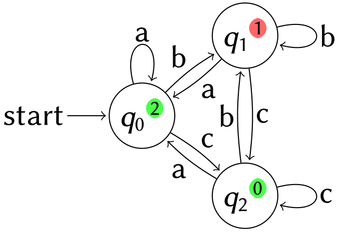
- NBA unsuitable: $\overset{\text{unrestricted}}{\text{nondeterminism}} \rightarrow\lightning\leftarrow$ probabilities

- **classical solution**:
  determinize the NBA to ~~DBA~~. impossible in general!
  a suitable deterministic

  (Muller Streett) Rabin $\overset{\textit{is a}}{\longleftarrow}$ **Parity**

  automaton.

Example DPA: "finitely often b unless infinitely often c"



Parity acceptance:

$w \in L(\mathcal{A}) \Leftrightarrow$ *min.* priority seen *inf.* often along run is *even*

- powerset construction is not enough for NBA!

McNaughton 1966: $2^{2^{\mathcal{O}(n)}}$

- powerset construction is not enough for NBA!

McNaughton 1966: $2^{2^{\mathcal{O}(n)}}$

Safra 1988: $12^n n^{2n} \in 2^{\mathcal{O}(n \log n)}$

Piterman 2007: $2n^n n!$

Schewe 2009: $(n!)^2$

- powerset construction is not enough for NBA!

McNaughton 1966: $2^{2^{\mathcal{O}(n)}}$

???

Safra 1988: $12^n n^{2n} \in 2^{\mathcal{O}(n \log n)}$

Piterman 2007: $2n^n n!$

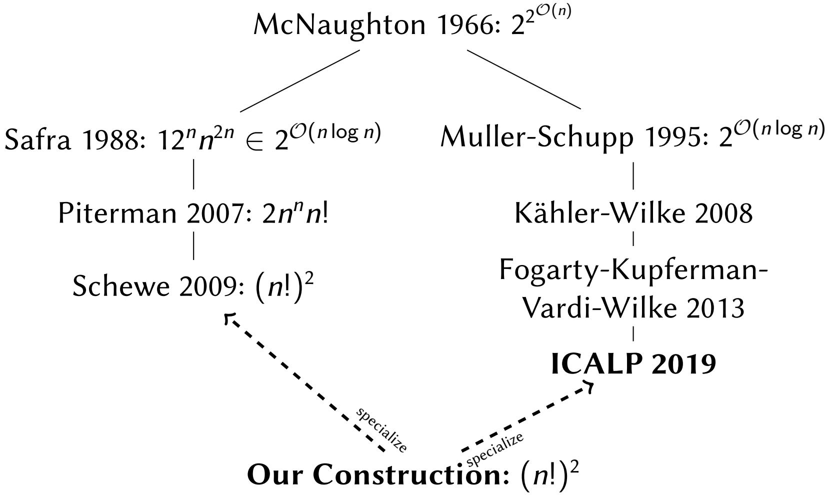
Schewe 2009: $(n!)^2$

Muller-Schupp 1995: $2^{\mathcal{O}(n \log n)}$

Kähler-Wilke 2008

Fogarty-Kupferman-Vardi-Wilke 2013

- powerset construction is not enough for NBA!



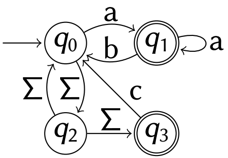
Idea:

- Organize reachable states in a tree (= new states)
- Tree nodes have an “age rank” (introduction order)
- Derive acceptance from green and red node events

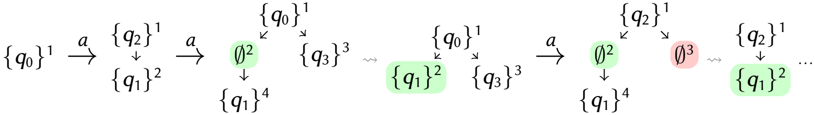
Idea:

- Organize reachable states in a tree (= new states)
- Tree nodes have an "age rank" (introduction order)
- Derive acceptance from green and red node events

NBA $\mathcal{A}$:



$w = a^\omega$

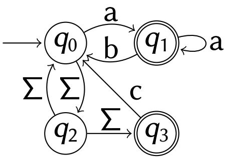


Safra-Piterman construction (quick sketch)

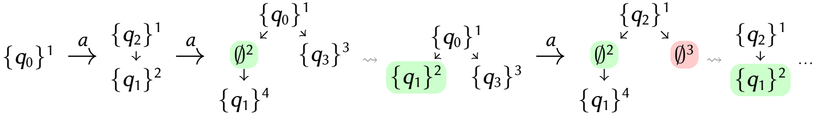
Idea:

- Organize reachable states in a tree (= new states)
- Tree nodes have an "age rank" (introduction order)
- Derive acceptance from green and red node events

NBA $\mathcal{A}$:

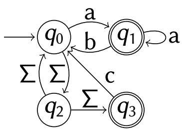


$w = a^\omega$



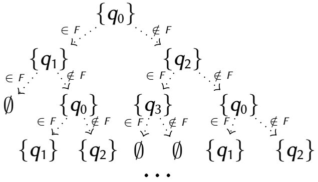
Accept, iff ∃ node rank that is infinitely often green
and finitely often red

$\mathcal{A}$:



$w = (ab)^\omega$

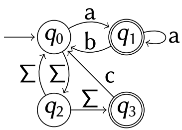
Split-tree:



- each NBA run can be traced
- each acc. run is "left-path"

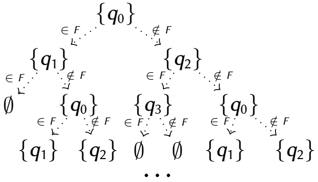
A variant of the Muller-Schupp construction

$\mathcal{A}$:

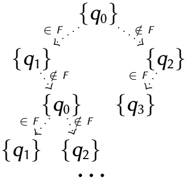


$w = (ab)^\omega$

Split-tree:

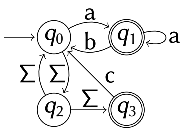


Reduced Split-tree:



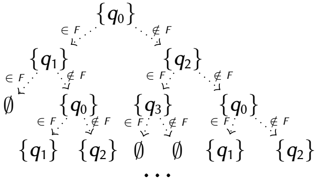
- each NBA run can be traced
- each acc. run is "left-path"

$\mathcal{A}$:

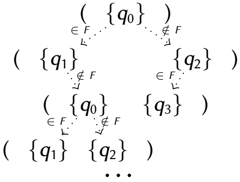


$w = (ab)^\omega$

Split-tree:

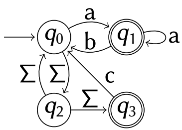


Reduced Split-tree:



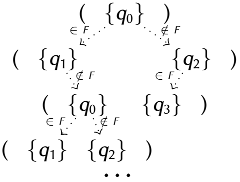
- each NBA run can be traced
- each acc. run is "left-path"
- fin. many different levels
- one can show [KW08]: NBA accepts iff the reduced split-tree has left-path

$\mathcal{A}$:



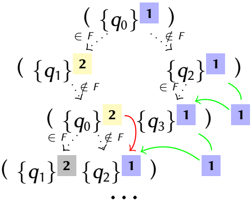
$w = (ab)^\omega$

Reduced Split-tree:



- fin. many different levels
- one can show [KW08]: NBA accepts iff the reduced split-tree has left-path

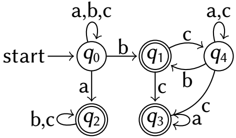
Muller-Schupp:



- add ranked identity tokens
- Tokens move across tree levels
- Accept iff a token has fin. many red and inf. green events

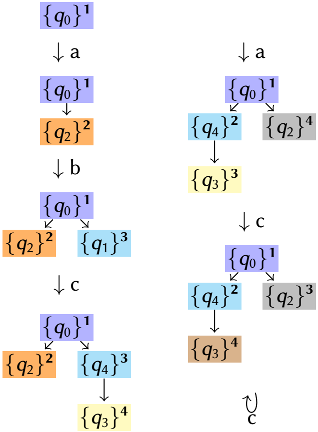
Comparison of Safra and Muller-Schupp

NBA:

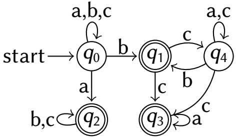


w = abcac$^\omega$

ranked Safra trees on w:

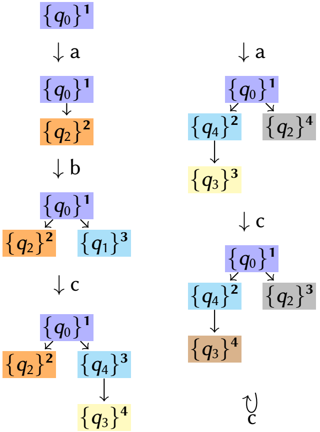


NBA:

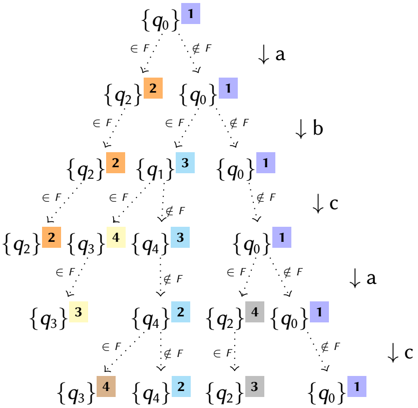


$w = abcac^\omega$

ranked Safra trees on w:



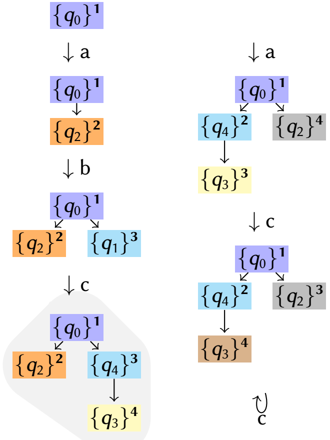
Muller-Schupp tuples on w:



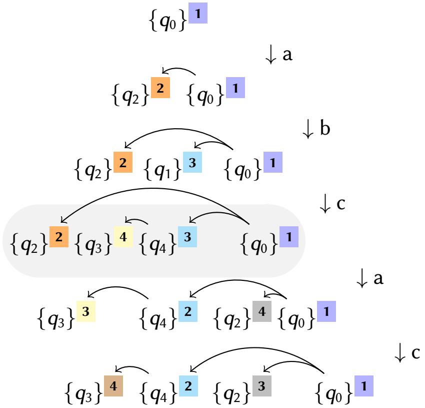
Key observation:

ranked Safra trees $\xleftrightarrow{\text{1-to-1}}$ ranked Muller-Schupp tuples

ranked Safra trees on w:



Muller-Schupp tuples on w:

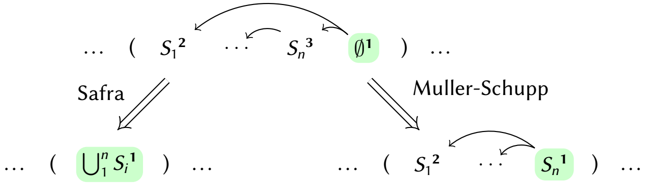


Key observation:

ranked Safra trees $\overset{\text{1-to-1}}{\longleftrightarrow}$ ranked Muller-Schupp tuples

Only difference between the constructions:

handling of nodes with green events



- Formulated on ranked tuples (like Muller-Schupp)
- Transition defined using four simple operations:
  step, **prune**, **merge**, **normalize**

- Formulated on ranked tuples (like Muller-Schupp)
- Transition defined using four simple operations: **step**, **prune**, **merge**, **normalize**
- **merge** is optional and also non-deterministic: “when a rank is red or green, merge sets as you like, while satisfying [*technical condition*]”

- Formulated on ranked tuples (like Muller-Schupp)
- Transition defined using four simple operations:
  step, **prune**, **merge**, **normalize**
- **merge** is optional and also non-deterministic:
  "when a rank is red or green, merge sets
  as you like, while satisfying [*technical condition*]"

- Different instantiations of this merge rule yield:
  - Muller-Schupp-style construction (= no merge)
  - Safra-style construction
  - A new "maximal merge" rule
  - ... and other possibilities

A sketch of the unified construction

$$(\alpha, t) = (\quad S_1^{\alpha(1)} \quad S_2^{\alpha(2)} \quad S_3^{\alpha(3)} \quad \cdots \quad S_n^{\alpha(n)} \quad)$$

A sketch of the unified construction

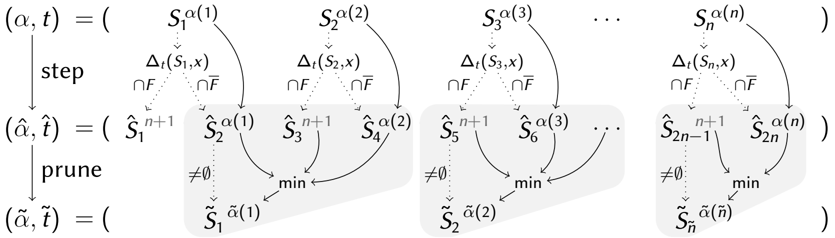
$$(\alpha, t) = \left(S_1^{\alpha(1)} \quad S_2^{\alpha(2)} \quad S_3^{\alpha(3)} \quad \cdots \quad S_n^{\alpha(n)} \right)$$

step

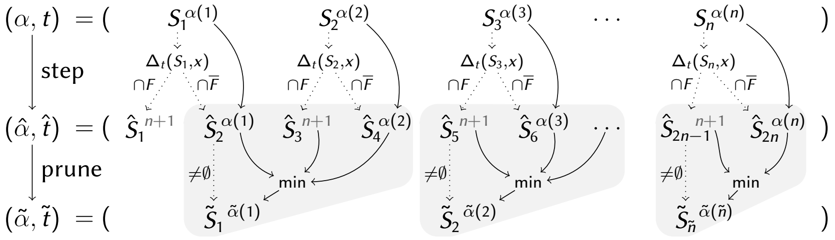
$$\Delta_t(S_1,x) \quad \cap F \quad \cap \overline{F} \qquad \Delta_t(S_2,x) \quad \cap F \quad \cap \overline{F} \qquad \Delta_t(S_3,x) \quad \cap F \quad \cap \overline{F} \qquad \Delta_t(S_n,x) \quad \cap F \quad \cap \overline{F}$$

$$(\hat{\alpha}, \hat{t}) = \left(\hat{S}_1^{\,n+1} \quad \hat{S}_2^{\,\alpha(1)} \quad \hat{S}_3^{\,n+1} \quad \hat{S}_4^{\,\alpha(2)} \quad \hat{S}_5^{\,n+1} \quad \hat{S}_6^{\,\alpha(3)} \quad \cdots \quad \hat{S}_{2n-1}^{\,n+1} \quad \hat{S}_{2n}^{\,\alpha(n)} \right)$$

A sketch of the unified construction

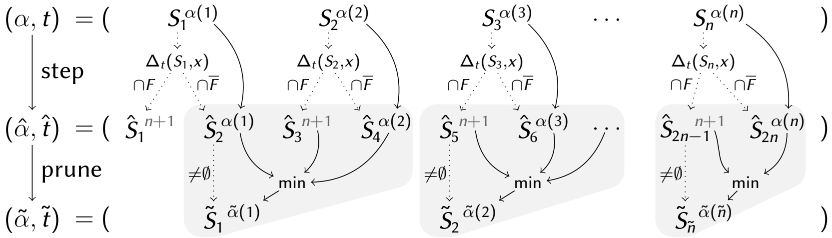


A sketch of the unified construction



(determine active (green + red) ranks and smallest active rank k)

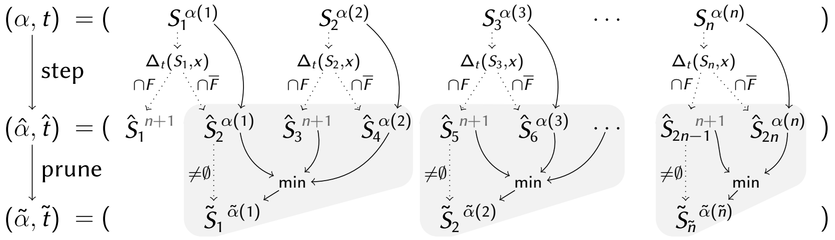
A sketch of the unified construction



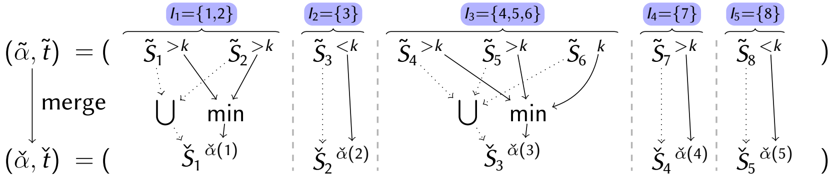
(determine active (green + red) ranks and smallest active rank k)

$$(\tilde{\alpha}, \tilde{t}) = (\overbrace{\tilde{S}_1^{\,>k} \quad \tilde{S}_2^{\,>k}}^{I_1=\{1,2\}} \quad \overbrace{\tilde{S}_3^{\,<k}}^{I_2=\{3\}} \quad \overbrace{\tilde{S}_4^{\,>k} \quad \tilde{S}_5^{\,>k} \quad \tilde{S}_6^{\ k}}^{I_3=\{4,5,6\}} \quad \overbrace{\tilde{S}_7^{\,>k}}^{I_4=\{7\}} \quad \overbrace{\tilde{S}_8^{\,<k}}^{I_5=\{8\}})$$

A sketch of the unified construction



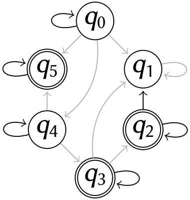
(determine active (green + red) ranks and smallest active rank k)



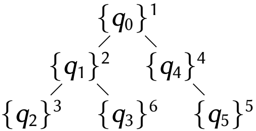
$\xrightarrow{\text{normalize}} (\alpha', t')$

The flexibility of the merge operation

NBA $\mathcal{A}$:

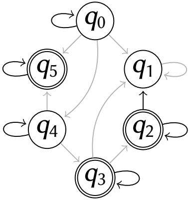


$(\alpha, t) =$

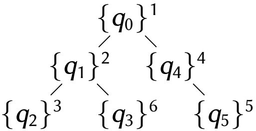


The flexibility of the merge operation

NBA $\mathcal{A}$:

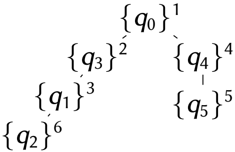


$(\alpha, t) =$

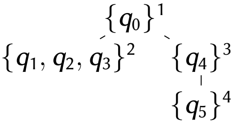


depending on performed **merge** we get different successors:

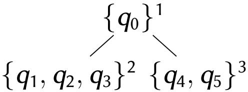
Muller-Schupp:



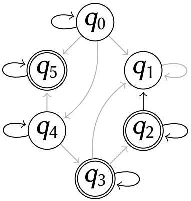
Safra:



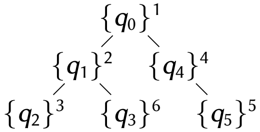
Maximal merge:



NBA $\mathcal{A}$:

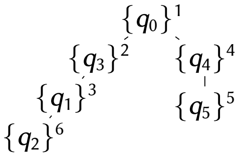


$(\alpha, t) =$

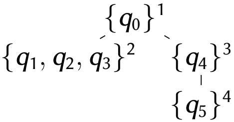


depending on performed **merge** we get different successors:

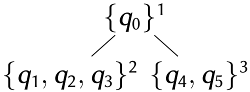
Muller-Schupp:

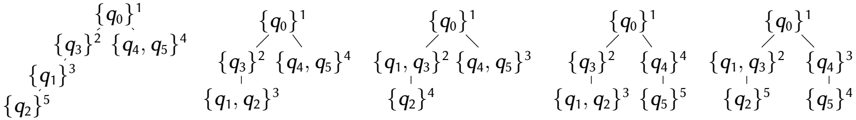


Safra:



Maximal merge:





Introduction

NBA Determinization
Safra Construction
Muller-Schupp Construction
Comparison of Safra and Muller-Schupp
A Sketch of the Unified Construction
Optimizations and Heuristics

Characterization of PBA subclasses
Ambiguity of NBA
Probabilistic Büchi Automata
Weak PBA
Ambiguity-restricted PBA
Complexity results

Summary

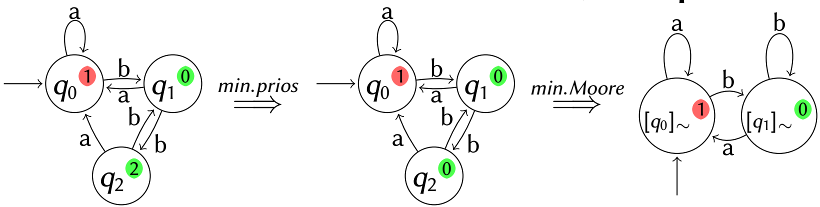
Optimization using preprocessing:
Use known *language inclusions* between NBA states (e.g. from simulation techniques) to simplify DPA states.
Example: (assuming $L(q) \supseteq L(p)$)

$$(\{\mathbf{q}, r\}^2, \{s\}^3, \{\mathbf{p}, t\}^1) \overset{opt.}{\rightsquigarrow} (\{\mathbf{q}, r\}^2, \{s\}^3, \{t\}^1)$$

Optimization using preprocessing:
Use known *language inclusions* between NBA states
(e.g. from simulation techniques) to simplify DPA states.
Example: (assuming $L(q) \supseteq L(p)$)

$$(\{\mathbf{q}, r\}^2, \{s\}^3, \{\mathbf{p}, t\}^1) \overset{opt.}{\rightsquigarrow} (\{\mathbf{q}, r\}^2, \{s\}^3, \{t\}^1)$$

Postprocessing optimization:
first minimize priorities (⇒ hope for more equiv. traces)
then "minimize" DPA like Moore automaton, **Example:**



Use modularized determinization construction:

- use sep. tuples for each NBA SCC, run them in parallel
- used for heuristics targeting acc./rej./det. SCCs of NBA

DPA state of original version: $\left(S_1^{\alpha(1)}, S_2^{\alpha(2)}, \ldots, S_n^{\alpha(n)}\right)$

DPA state of modularized version:

$$\underbrace{(\{q_1, q_3\}^5, \{q_2\}^3)}_{tracked\ SCC\ 1} \mid \underbrace{(\{q_4\}^1, \{q_6\}^6, \{q_5\}^4)}_{tracked\ SCC\ 2} \mid \underbrace{(\{q_7, q_8, q_9\}^2)}_{tracked\ SCC\ 3} \mid\mid \underbrace{\{q_{10}\}}_{buffer}$$

Use modularized determinization construction:

- use sep. tuples for each NBA SCC, run them in parallel
- used for heuristics targeting acc./rej./det. SCCs of NBA

DPA state of original version: $\left(S_1{}^{\alpha(1)}, S_2{}^{\alpha(2)}, \ldots, S_n^{\alpha(n)}\right)$

DPA state of modularized version:

$$\underbrace{(\{q_1, q_3\}^5, \{q_2\}^3)}_{tracked\ SCC\ 1} \mid \underbrace{(\{q_4\}^1, \{q_6\}^6, \{q_5\}^4)}_{tracked\ SCC\ 2} \mid \underbrace{(\{q_7, q_8, q_9\}^2)}_{tracked\ SCC\ 3} \mid\mid \underbrace{\{q_{10}\}}_{buffer}$$

Use degrees of freedom of the construction:

- manage existing states of DPA in a trie for quick lookup
- on each transition, try to reuse suitable existing states

Use modularized determinization construction:

- use sep. tuples for each NBA SCC, run them in parallel
- used for heuristics targeting acc./rej./det. SCCs of NBA

DPA state of original version: $\left(S_1^{\alpha(1)}, S_2^{\alpha(2)}, \ldots, S_n^{\alpha(n)}\right)$

DPA state of modularized version:

$$\underbrace{(\{q_1, q_3\}^5, \{q_2\}^3)}_{tracked\ SCC\ 1} \mid \underbrace{(\{q_4\}^1, \{q_6\}^6, \{q_5\}^4)}_{tracked\ SCC\ 2} \mid \underbrace{(\{q_7, q_8, q_9\}^2)}_{tracked\ SCC\ 3} \mid\mid \underbrace{\{q_{10}\}}_{buffer}$$

Use degrees of freedom of the construction:

- manage existing states of DPA in a trie for quick lookup
- on each transition, try to reuse suitable existing states

Use information in naive "powerset automaton" (PSA):

- determinize each SCC of PSA separately
- per PSA SCC, keep one bottom SCC of partial DPA

Introduction

NBA Determinization
- Safra Construction
- Muller-Schupp Construction
- Comparison of Safra and Muller-Schupp
- A Sketch of the Unified Construction
- Optimizations and Heuristics

Characterization of PBA subclasses
- Ambiguity of NBA
- Probabilistic Büchi Automata
- Weak PBA
- Ambiguity-restricted PBA
- Complexity results

Summary

- Ambiguity on word w: # of different accepting runs on w
- Ambiguity class of $\mathcal{A}$: amb. upper bound on any word

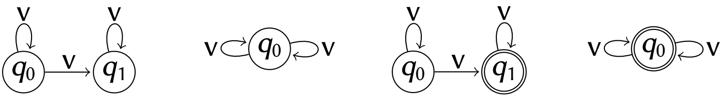
Brief detour: Patterns characterizing NBA ambiguity (DLT 2018)

- Ambiguity on word *w*: # of different accepting runs on *w*
- Ambiguity class of $\mathcal{A}$: amb. upper bound on any word

---

Existence of the pattern

IDA **EDA** **IDA$_F$** **EDA$_F$**

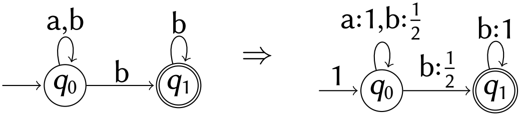


for some $v \in \Sigma^*$ implies at least

polynomial **exponential** *strict*— **countable** $(= \aleph_0\text{-amb.})$ **uncountable** $(= 2^{\aleph_0}\text{-amb.})$

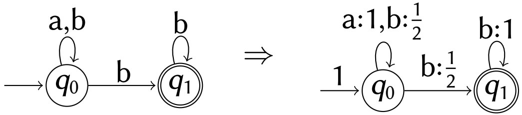
ambiguity of the automaton (none present ⇒ **finite**)

Consider probabilistic instead of nondeterministic choice:



- replace Q_0 with initial distribution $\mu_0 : Q \to [0, 1]$
- replace Δ with probabilistic distributions $\delta(q, x)$
  i.e. $\sum_{q' \in Q} \delta(q, x, q') = 1$ for all $q \in Q, x \in \Sigma$

Consider probabilistic instead of nondeterministic choice:



- replace Q_0 with initial distribution $\mu_0 : Q \to [0, 1]$
- replace Δ with probabilistic distributions $\delta(q, x)$ i.e. $\sum_{q' \in Q} \delta(q, x, q') = 1$ for all $q \in Q, x \in \Sigma$
- choose some semantics to define languages:
  - almost sure acceptance $L^{=1}(\mathcal{A})$
  - positive acceptance $L^{>0}(\mathcal{A})$
  - threshold acceptance $L^{>\lambda}(\mathcal{A}), \lambda \in \mathbb{R}$

Unlike PFA, which are still regular under > 0 and $= 1$ semantics, the following holds for PBA:

$$
\begin{array}{lccccl}
 & \mathbb{L}(DBA) & \subset & \mathbb{L}^{=1}(PBA) & \\
 & \cap & & \cap & \\
\text{regular} & \mathbb{L}(NBA) & \subset & \mathbb{L}^{>0}(PBA) & \text{non-regular} \\
 & = \omega\text{-Reg} & & \cap & \\
 & & & \mathbb{L}^{>\lambda}(PBA) &
\end{array}
$$

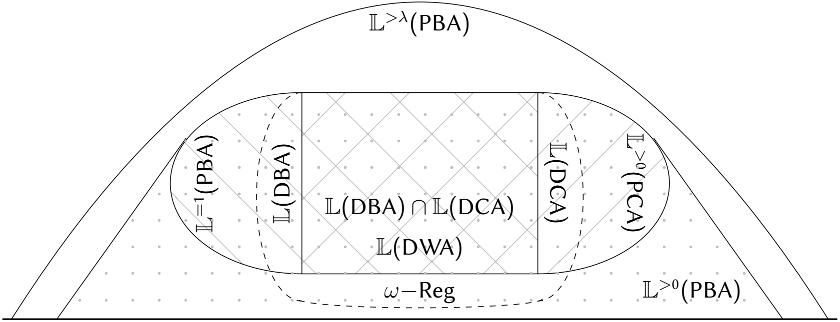
Unlike PFA, which are still regular under >0 and $=1$ semantics, the following holds for PBA:

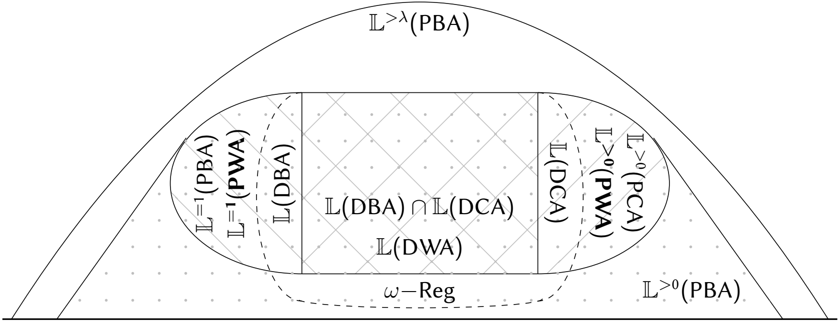
$$\begin{array}{lccc l} & \mathbb{L}(DBA) & \subset & \mathbb{L}^{=1}(PBA) & \\ & \cap & & \cap & \\ \text{regular} & \mathbb{L}(NBA) & \subset & \mathbb{L}^{>0}(PBA) & \text{non-regular} \\ & = \omega\text{-Reg} & & \cap & \\ & & & \mathbb{L}^{>\lambda}(PBA) & \end{array}$$

Furthermore,

- checking $L^{=1}(\mathcal{A})=\emptyset$ or $L^{=1}(\mathcal{A})=\Sigma^\omega$ is PSPACE complete,
- checking $L^{>0}(\mathcal{A})=\emptyset$ or $L^{>0}(\mathcal{A})=\Sigma^\omega$ is undecidable.
- **Goal:** find more “tame” subclasses with good properties

Weak probabilistic Büchi automata

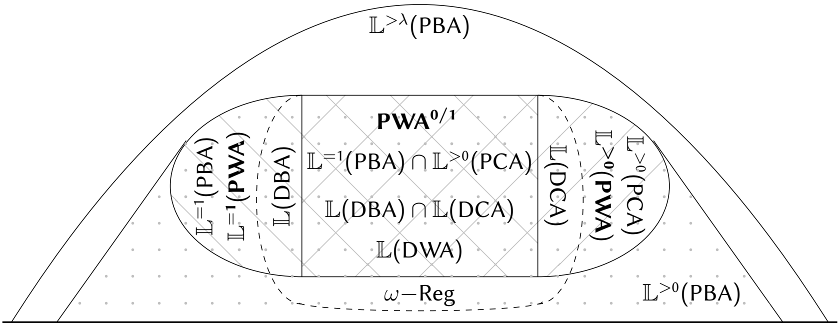




We characterized the expressivity of weak PBA:

Theorem (FOSSACS 2020)

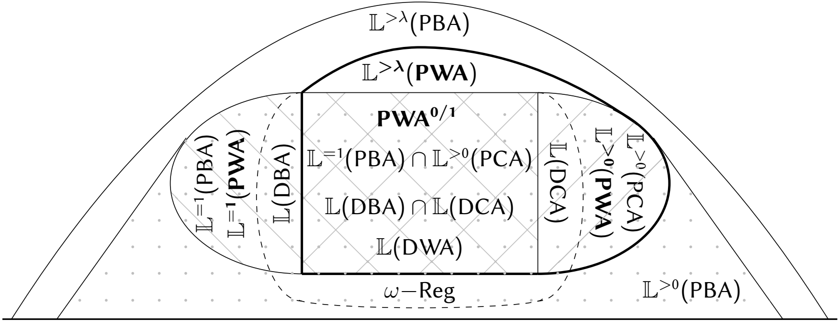
- $\mathbb{L}^{>0}(\text{PWA}) = \mathbb{L}^{>0}(\text{PCA})$ *and* $\mathbb{L}^{=1}(\text{PWA}) = \mathbb{L}^{=1}(\text{PBA})$



We characterized the expressivity of weak PBA:

Theorem (FOSSACS 2020)

- $\mathbb{L}^{>0}(\mathrm{PWA}) = \mathbb{L}^{>0}(\mathrm{PCA})$ *and* $\mathbb{L}^{=1}(\mathrm{PWA}) = \mathbb{L}^{=1}(\mathrm{PBA})$
- $\mathbb{L}^{>0}(\mathrm{PWA}) \cap \mathbb{L}^{=1}(\mathrm{PWA}) = \mathbb{L}(\mathrm{DWA}) = \mathbb{L}(\mathrm{PWA}^{0/1})$



We characterized the expressivity of weak PBA:

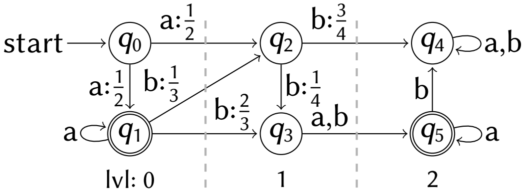
Theorem (FOSSACS 2020)

- $\mathbb{L}^{>0}(\mathrm{PWA}) = \mathbb{L}^{>0}(\mathrm{PCA})$ *and* $\mathbb{L}^{=1}(\mathrm{PWA}) = \mathbb{L}^{=1}(\mathrm{PBA})$
- $\mathbb{L}^{>0}(\mathrm{PWA}) \cap \mathbb{L}^{=1}(\mathrm{PWA}) = \mathbb{L}(\mathrm{DWA}) = \mathbb{L}(\mathrm{PWA}^{0/1})$
- $\mathbb{L}^{>0}(\mathrm{PWA}) \subset \mathbb{L}^{>\lambda}(\mathrm{PWA}) \subset \mathbb{L}^{>\lambda}(\mathrm{PBA})$

Two PBA restrictions identified in [Chadha et al. '09/'11]:

- *Hierarchical PBA* have the following property:
- There exists a function lvl : $Q \to \mathbb{N}$ s.t. $\forall q \in Q, a \in \Sigma$
  - no transition leads to a state q' with lower level
  - there is *at most one* trans. to some q' with the same level

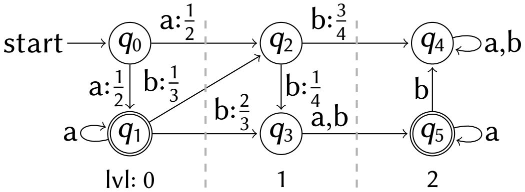
HPBA:



Two PBA restrictions identified in [Chadha et al. ’09/’11]:

- *Hierarchical PBA* have the following property:
- There exists a function lvl : $Q \to \mathbb{N}$ s.t. $\forall q \in Q, a \in \Sigma$
  - no transition leads to a state q' with lower level
  - there is *at most one* trans. to some q' with the same level

HPBA:

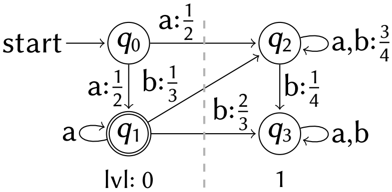


- $\mathbb{L}^{>0}(HPBA) = \mathbb{L}(NBA), \quad \mathbb{L}^{=1}(HPBA) = \mathbb{L}(DBA)$

Two PBA restrictions identified in [Chadha et al. '09/'11]:

- *Hierarchical PBA* have the following property:
- There exists a function lvl : $Q \to \mathbb{N}$ s.t. $\forall q \in Q, a \in \Sigma$
  - no transition leads to a state q' with lower level
  - there is *at most one* trans. to some q' with the same level

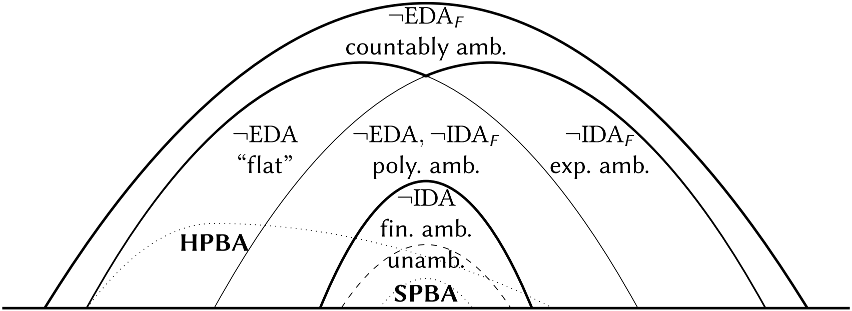
SPBA:



- $\mathbb{L}^{>0}(HPBA) = \mathbb{L}(NBA), \quad \mathbb{L}^{=1}(HPBA) = \mathbb{L}(DBA)$

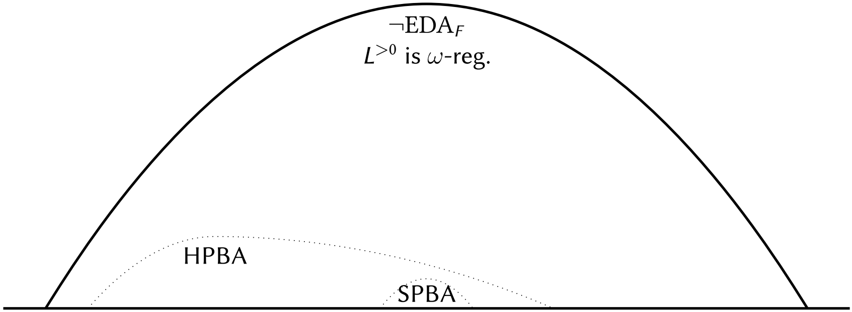
- *Simple PBA* = 2-level HPBA with all acc. states on level 0
- $\mathbb{L}^{>0}(SPBA) = \mathbb{L}^{=1}(SPBA) = \mathbb{L}^{>\lambda}(SPBA) = \mathbb{L}(DBA)$

By simple observations wrt. ambiguity patterns we have:



It seems *natural* to also check the expressivity of PBA without certain ambiguity patterns!

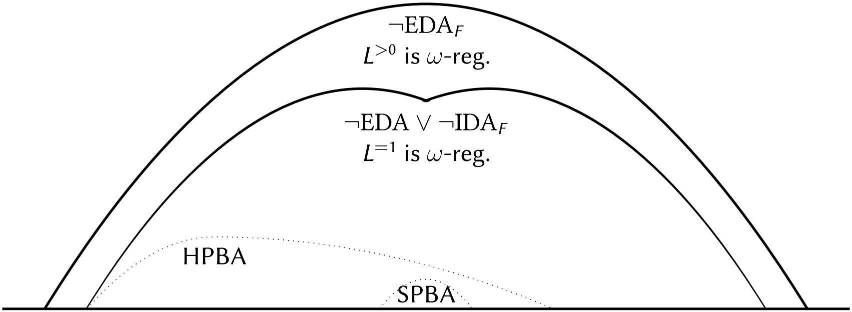
full classification of the syntactic amb. classes wrt. regularity under $>0/=1/>\lambda$ semantics, generalizing SPBA/HPBA:



Theorem (FOSSACS 2020)

- $\mathbb{L}^{>0}(\textit{fin. amb.}\ \text{PBA}) = \mathbb{L}^{>0}(\aleph_0\textit{-amb.}\ \text{PBA}) = \mathbb{L}(\text{NBA})$

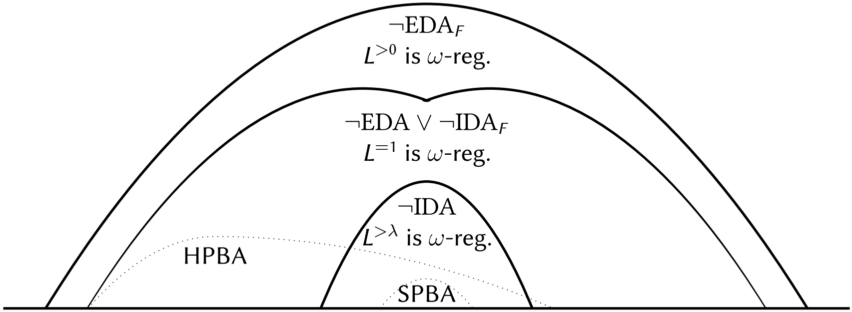
full classification of the syntactic amb. classes wrt. regularity under $> 0/= 1/> \lambda$ semantics, generalizing SPBA/HPBA:



Theorem (FOSSACS 2020)

- $\mathbb{L}^{>0}$(*fin. amb.* PBA) $= \mathbb{L}^{>0}$($\aleph_0$*-amb.* PBA) $= \mathbb{L}$(NBA)
- $\mathbb{L}^{=1}$($\leq$ *exp. amb.* PBA) $= \mathbb{L}^{=1}$(*flat* PBA) $= \mathbb{L}$(DBA) $\subset \mathbb{L}^{=1}$($\aleph_0$*-amb.* PBA)

full classification of the syntactic amb. classes wrt. regularity under $> 0/= 1/> \lambda$ semantics, generalizing SPBA/HPBA:



Theorem (FOSSACS 2020)

- $\mathbb{L}^{>0}$(*fin. amb.* PBA) $=$ $\mathbb{L}^{>0}$($\aleph_0$-*amb.* PBA) $=$ $\mathbb{L}$(NBA)
- $\mathbb{L}^{=1}$($\leq$ *exp. amb.* PBA) $=$ $\mathbb{L}^{=1}$(*flat* PBA) $=$ $\mathbb{L}$(DBA) $\subset$ $\mathbb{L}^{=1}$($\aleph_0$-*amb.* PBA)
- $\mathbb{L}^{>\lambda}$(*fin. amb.* PBA) $=$ $\mathbb{L}$(NBA) $\subset$ $\mathbb{L}^{>\lambda}$(*poly. amb.* PBA)

Summary of results on ambiguity-restricted PBA (FOSSACS 2020)

PBA Type	ω-regular?			Emptiness		Universality	
	> 0	$= 1$	$> \lambda$	> 0	$= 1$	> 0	$= 1$
fin. amb.							
pol. amb.				$\in$ NL	$\in$ PSPACE	$\in$ PSPACE	$\in$ NL
exp. amb.							
flat				$\in$ NL c.	$\in$ PSPACE c.	$\in$ PSPACE c.	$\in$ NL c.
$\aleph_0$ amb.							$\in$ PSPACE

Summary of results on ambiguity-restricted PBA (FOSSACS 2020)

PBA Type	ω-regular?			Emptiness		Universality	
	> 0	$= 1$	$> \lambda$	> 0	$= 1$	> 0	$= 1$
fin. amb.							
pol. amb.				$\in$ NL	$\in$ PSPACE	$\in$ PSPACE	$\in$ NL
exp. amb.							
flat				$\in$ NL c.	$\in$ PSPACE c.	$\in$ PSPACE c.	$\in$ NL c.
$\aleph_0$ amb.							$\in$ PSPACE

- In contrast to the classical setting, restricting ambiguity influences the expressivity

PBA Type	ω-regular?			Emptiness		Universality	
	> 0	$= 1$	$> \lambda$	> 0	$= 1$	> 0	$= 1$
fin. amb.				$\in$ NL	$\in$ PSPACE	$\in$ PSPACE	$\in$ NL
pol. amb.							
exp. amb.							
flat				$\in$ NL c.	$\in$ PSPACE c.	$\in$ PSPACE c.	$\in$ NL c.
$\aleph_0$ amb.							$\in$ PSPACE

- In contrast to the classical setting, restricting ambiguity influences the expressivity
- The identified ambiguity-restricted PBA classes
  - subsume Hierarchical and Simple PBA
  - have similar complexity results

In joint work with Christof Löding, we obtained the following main results, presented in the PhD thesis:

- a new unified determinization construction for NBA **(ICALP 2019)**
- multiple new optimizations that are based on it **(ATVA 2019)**
- a full characterization of ambiguity in Büchi automata **(DLT 2018)**
- new regular subclasses of probabilistic Büchi automata **(FOSSACS 2020)**