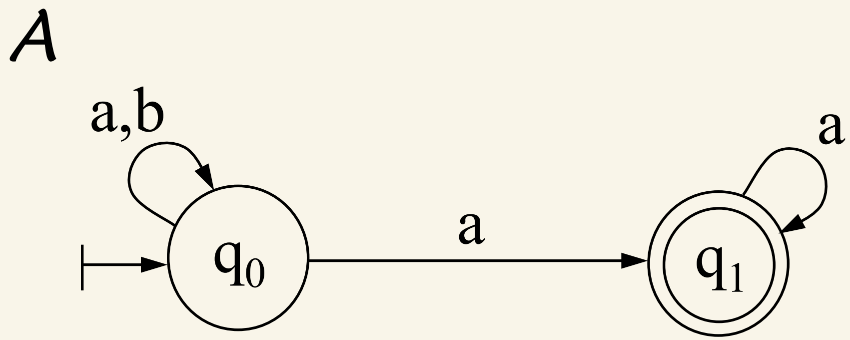
Inherent Size Blowup in ω-Automata

Udi Boker
IDC Herzliya, Israel

DLT *2019*

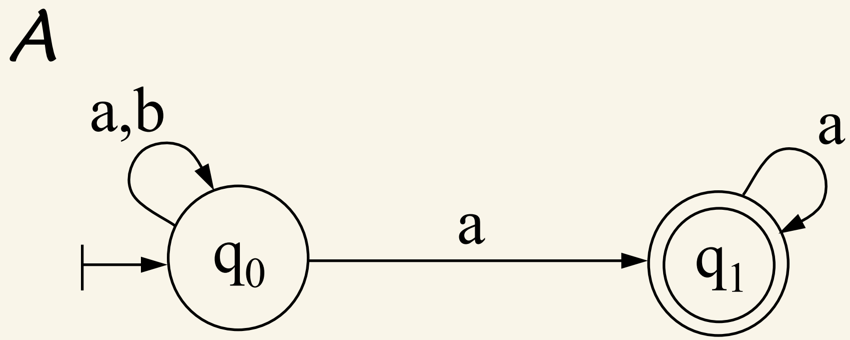
Automata on Finite Words



L(A) = Words that end with an ‘a’

- You all know it, but…
  we will not be speaking of them

Automata on Infinite Words



$L(A) = ?$

- Automata on infinite words were introduced in the 60's, in the course of solving fundamental decision problems in logic.
- Nowadays, they are widely used in formal verification and synthesis of nonterminating (reactive) systems.

But, how do we define the acceptance condition?

Many Automata Types

1962, Büchi

Co-Büchi

1986, Weak

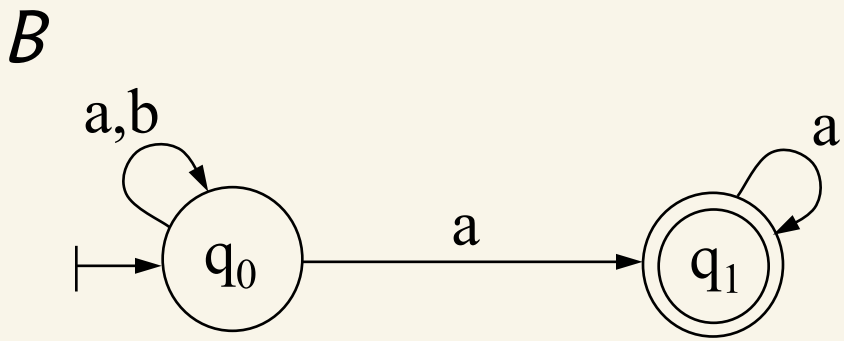
1984, Parity

1969, Rabin

1982, Streett

1963, Muller

Büchi Automata
(By Julius Büchi in 1962)

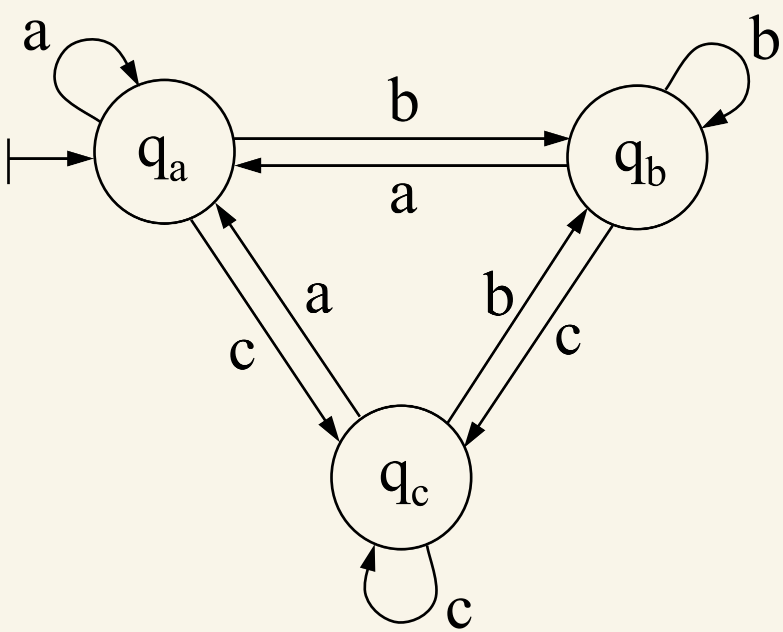


L(*B*) = Finitely many b's

- There is a set of accepting states.
- A run is accepting if it visits some accepting state infinitely often.
- A word is accepted if there is an accepting run on it.

Rabin Automata
(By Michael Rabin in 1969)

R



Acceptance condition:

Bad Good

$\langle \{q_a\}, \{q_b, q_c\} \rangle$;

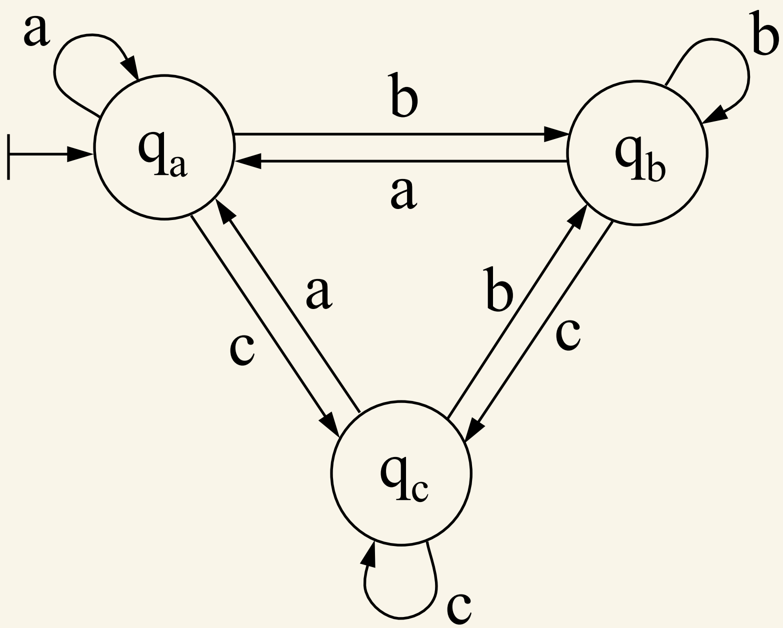
$\langle \{q_b\}, \{q_a\} \rangle$

- A run r is accepting if exists an accepting pair ⟨B,G⟩, such that r visits B finitely often and G infinitely often.

Streett Automata

(By Robert Streett in 1982)

$\mathcal{S}$



Acceptance condition:

Bad Good

$\langle\{q_b,q_c\}, \{q_a\}\rangle$;

$\langle\{q_a\}, \{q_b\}\rangle$

- A run r is accepting if for every pair $\langle B,G\rangle$, r visits B finitely often or G infinitely often.

Why These Automata Types?

Can't we do with one good acceptance condition?

I'm so simple.

Easy emptiness check, intersection, …

Büchi

Rabin

Too many details.

I am sophisticated.

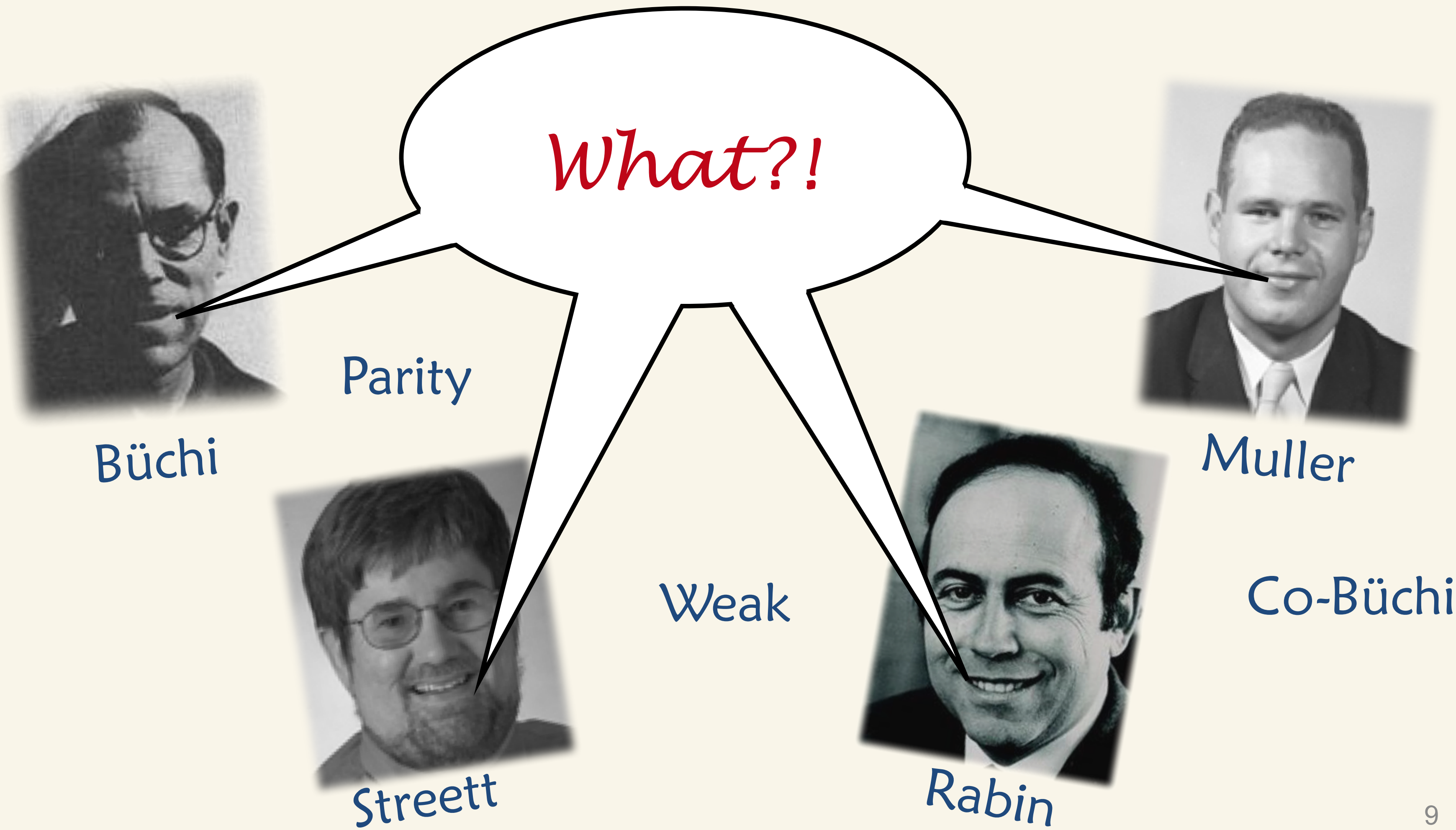
Your deterministic version is not strong enough.

I am the most detailed.

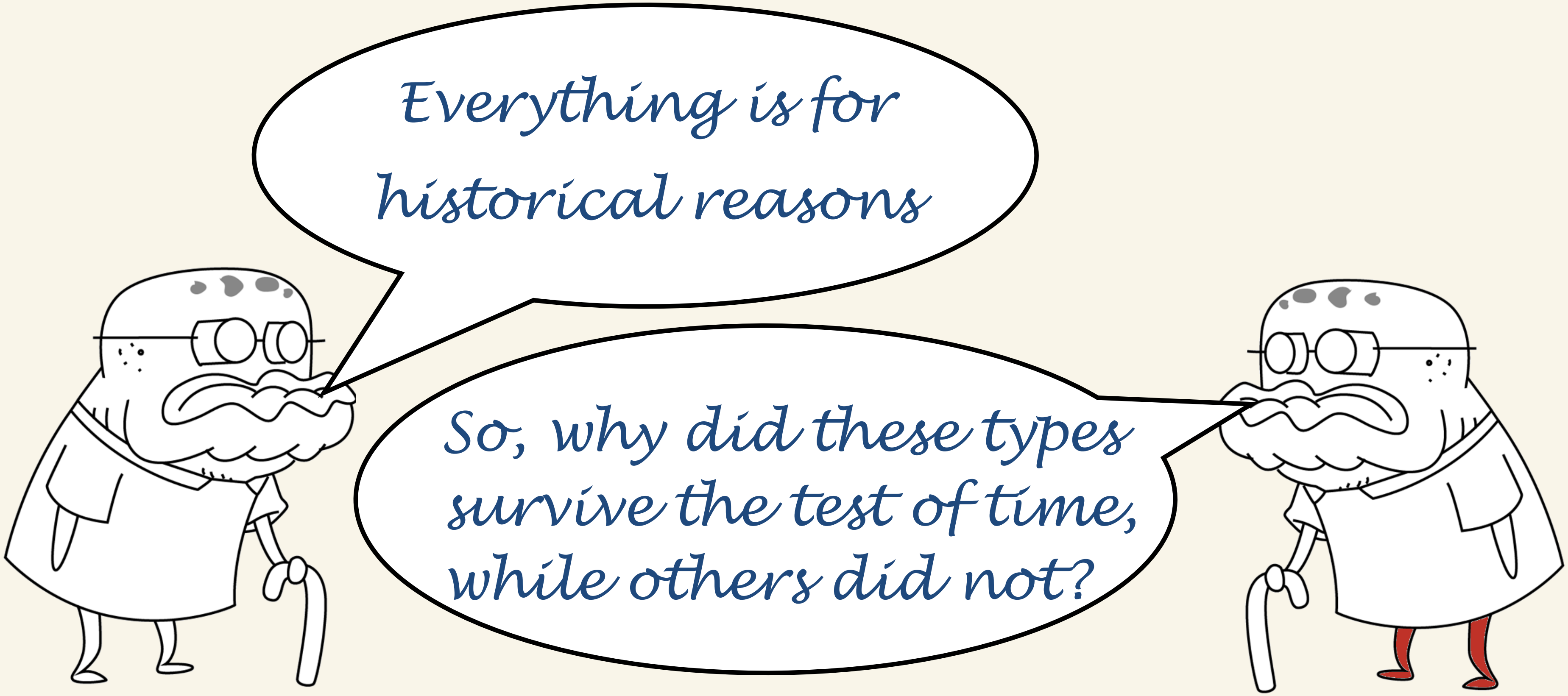
Muller

Why These Automata Types?

Do we need more acceptance conditions?



Why These Automata Types?



[Kurshan, 1994]: "The choice of automaton type to use in connection with formal verification is governed by two issues: *syntactic suitability* and *computational complexity*."

The Plan

We analyze the classic types by their:

- Connection to other formalisms
- Expressiveness
- Succinctness (with respect to each other and to arbitrary types)
- Complexity of decision problems
- Size blowup involved in boolean operations

We answer whether:

- It is justified to use them all
- There is a need for additional types

Spoilers:

Yes

Yes

Classic types: Weak, Büchi, co-Büchi, parity, Rabin, Streett, Muller

Analysis of the Classic Types

Well known	• Connection to other formalisms *All are well connected…*
Well known	• Expressiveness *Clear picture*
Partially known Quite a mess	• Succinctness *Organize and almost complete*
Well known	• Decision problems *Clear picture*
Partially known (as of 2018)	• Boolean operations *We complete the picture*

Connection to Other Formalisms

The different types are related to various aspects of formal verification, to various logics, and other formalisms.

For example:

- Weak: Linear temporal logic, alternation free mu-calculus
- Büchi and Co-Büchi : Fairness
- Parity: Mu-calculus
- Rabin and Streett: Strong fairness, memoryless strategies
- Muller: Wagner hierarchy

Expressiveness

Nondeterministic

Weak co-Büchi Büchi parity Rabin Streett Muller

weaker ω-regular complete

Deterministic

Weak co-Büchi Büchi parity Rabin Streett Muller

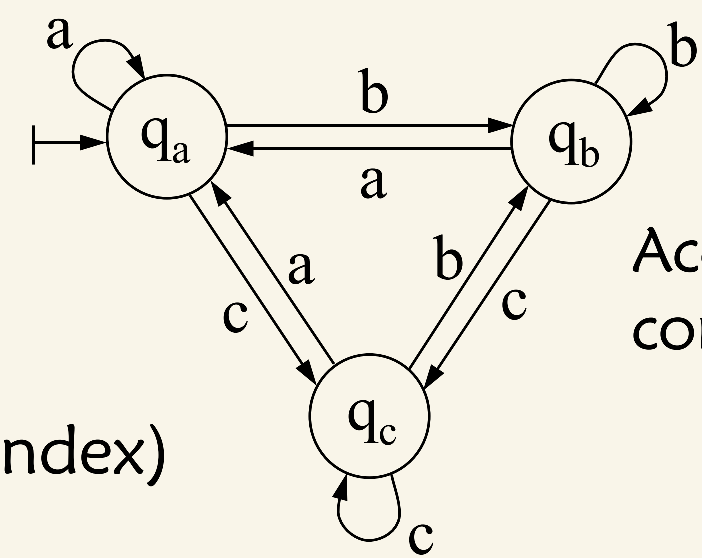
Succinctness – Between Them

- Due to the advantages and disadvantages of each type, there is a rich literature on the translations between them.
  - Starting in the 60s and continuing until these days.
- But, the literature is inconsistent and incomplete.
  - There is inconsistency in the view of the automaton *size*.
  - There are "only" 175 non-self translations between them.
    - A problem to find the data.
    - Some translation-bounds are still missing.

Automaton Elements

An automaton has:

- Alphabet
- States
- Transitions
- Acceptance condition (Index)



Acceptance condition:

$\left.\begin{matrix} \\ \ldots \\ \\ \end{matrix}\right\}$ Index

- All results on the translations between automata, when speaking of size, take into account the number of states.
- Yet, some ignore the alphabet, and others ignore the index.

Automaton Size

We define:

Size = max(alphabet, states, transitions, index)

The source and target automaton share the same alphabet.

Ideally:

- Upper bounds over arbitrary alphabets
- Lower bounds over a fixed alphabet (i.e., of a constant size)

It indeed works, eliminating the alphabet influence.

Automaton Size

We define:

$$\boxed{\text{Size} = \max(_{\text{alphabet},}\ \text{states, transitions, index})}$$

- For deterministic automata, the number of transitions is bounded by States × Alphabet.
- For nondeterministic automata, the number of transitions is bounded by States × States × Alphabet.

Furthermore, it turns out that the transition blowup goes hand in hand with the state blowup.

Automaton Size

We define:

$$\text{Size} = \max(\text{alphabet}, \text{states}, \text{transitions}, \text{index})$$

We demonstrate next that the number of states and the index are both important and moreover – interconnected.

Streett ➠ Rabin

n states
Index n

2^n states; Index 1

or

n^2 states
Index 2^n

(From FSTTCS 2017)

- Inevitable State Blowup: Quadratic
- Inevitable Index Blowup: Constant

Yet,

- Inevitable Size Blowup: Exponential

Or something in between them

Automaton Size

We define:

$$\boxed{\text{Size} = \max(\text{alphabet, states, transitions, index})}$$

Moral:

- The state blowup is important

  but

- The size blowup is also important

Succinctness – Between Them

A web site with all results:

http://www.faculty.idc.ac.il/udiboker/automata

State blowup and Size blowup

Succinctness – W.r.t. Arbitrary Types

- Proposition: For every ω-regular nondet. automaton of size n, there is an equivalent nondet. Büchi automaton of size $2^{O(n)}$ and an equivalent det. parity automaton of size $2^{2^{O(n)}}$.
- [Safra & Vardi, 89]: There is a family A_n of ω-regular automata of size n, such that every ω-regular automaton for the complement of A_n has at least 2^{2^n} states.
- Conclusion: The classic types (except for Muller) provide a reasonable tradeoff between succinctness and blowup of determinization and complementation, having all of them singly exponential.

Decision Problems

<table>
<tr><th></th><th>Nonemptiness</th><th>Universality</th></tr>
<tr><td>Weak</td><td rowspan="3">Linear time</td><td rowspan="7">PSPACE-Complete</td></tr>
<tr><td>Co-Büchi</td></tr>
<tr><td>Büchi</td></tr>
<tr><td>Parity</td><td rowspan="4">PTIME</td></tr>
<tr><td>Rabin</td></tr>
<tr><td>Streett</td></tr>
<tr><td>Muller</td></tr>
</table>

Boolean Operations

Nondeterministic automata

<table>
<tr><th></th><th>Union</th><th>Intersect.</th><th>Complement.</th></tr>
<tr><td>Weak</td><td rowspan="7">Linear</td><td rowspan="6">Polynomial</td><td rowspan="6">Exponential</td></tr>
<tr><td>Co-Büchi</td></tr>
<tr><td>Büchi</td></tr>
<tr><td>Parity</td></tr>
<tr><td>Rabin</td></tr>
<tr><td>Streett</td></tr>
<tr><td>Muller</td><td>?</td><td>?</td></tr>
</table>

- The question marks stand for *open as of 2018.*

Boolean Operations

Nondeterministic automata

	Union	Intersect.	Complement.
Weak	Linear	Polynomial	Exponential
Co-Büchi	Linear	Polynomial	Exponential
Büchi	Linear	Polynomial	Exponential
Parity	Linear	Polynomial	Exponential
Rabin	Linear	Polynomial	Exponential
Streett	Linear	Polynomial	Exponential
Muller	Linear	Exponential	Double-Exp

- Muller automata are great for theoretical purposes
- Not suitable for practical implementation

Boolean Operations

Deterministic automata

Not ω-regular complete

	Union	Intersect.	Complement.
Weak			
Co-Büchi			
Büchi			
Parity			
Rabin			
Streett			
Muller			

Boolean Operations

Deterministic automata

Not
ω-regular
complete

	Union	Intersect.	Complement.
Weak			No blowup
Co-Büchi	Quadratic		No blowup (if possible)
Büchi			
Parity			
Rabin			
Streett			
Muller			

Boolean Operations

Deterministic automata

Not
ω-regular
complete

	Union	Intersect.	Complement.
Weak	Quadratic		No blowup
Co-Büchi			No blowup (if possible)
Büchi			
Parity			No blowup
Rabin			Exponential
Streett			
Muller			

Boolean Operations

Deterministic automata

Not ω-regular complete

<table>
<tr><th></th><th>Union</th><th>Intersect.</th><th>Complement.</th></tr>
<tr><td>Weak</td><td colspan="2" rowspan="3">Quadratic</td><td>No blowup</td></tr>
<tr><td>Co-Büchi</td><td rowspan="2">No blowup (if possible)</td></tr>
<tr><td>Büchi</td></tr>
<tr><td>Parity</td><td>?</td><td>?</td><td>No blowup</td></tr>
<tr><td>Rabin</td><td>Quadratic</td><td>?</td><td rowspan="3">Exponential</td></tr>
<tr><td>Streett</td><td>?</td><td>Quadratic</td></tr>
<tr><td>Muller</td><td>?</td><td>?</td></tr>
</table>

- Deterministic automata are not necessary for model checking
- Required in synthesis and probabilistic model checking

Boolean Operations

Deterministic automata

Not ω-regular complete

<table>
<tr><th></th><th>Union</th><th>Intersect.</th><th>Complement.</th></tr>
<tr><td>Weak</td><td colspan="2" rowspan="3">Quadratic</td><td>No blowup</td></tr>
<tr><td>Co-Büchi</td><td rowspan="2">No blowup (if possible)</td></tr>
<tr><td>Büchi</td></tr>
<tr><td>Parity</td><td colspan="2">Exponential</td><td>No blowup</td></tr>
<tr><td>Rabin</td><td>Quadratic</td><td>Exponential</td><td rowspan="3">Exponential</td></tr>
<tr><td>Streett</td><td>Exponential</td><td>Quadratic</td></tr>
<tr><td>Muller</td><td colspan="2">Exponential</td></tr>
</table>

Boolean Operations

Deterministic automata

Not ω-regular complete

	Union	Intersect.	Complement.
[illegible]	[illegible]	[illegible]	No blowup
Co-[illegible]	[illegible]	[illegible]	No blowup (if possible)
Büchi	[illegible]	[illegible]	No blowup (if possible)
Parity	Exponential	Exponential	No blowup
Rabin	Quadratic	Exponen	Exponential
Streett	Exponential	Quadra	Exponential
Muller	Exponential	Exponential	Exponential

Positive Boolean Operations

ω-regular-complete deterministic automata

	Union	Intersect.
Parity	Exponential	
Rabin	Quadratic	Exponential
Streett	Exponential	Quadratic
Muller	Exponential	

- Deterministic automata are required nowadays
- Union and intersection are very useful for compound systems
- They involve an exponential size blowup in all the ω-regular-complete classic automata

Is an exponential blowup inevitable?

Positive Boolean Operations

ω-regular-complete deterministic automata

Is an exponential blowup inevitable?

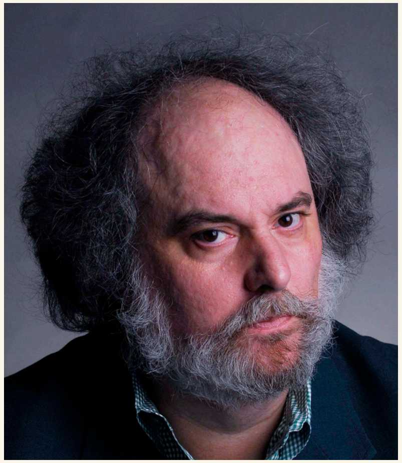
No

How?

Using stronger acceptance conditions

Does it worth it?

Seems so



Emerson-Lei Automata

- Introduced in 1985 by Allen Emerson and Chin-Laung Lei
- Some popularity shortly after; Less popular afterwards
- Regained popularity in the past seven years
- Acceptance condition: a boolean formula over *Fin*(*S*), *Inf*(*S*).
  - *S* is an arbitrary set of states.
  - *Fin*(*S*) / *Inf*(*S*) means that *S* is visited finitely/infinitely often
- Generalizes the other conditions
  - Büchi: $Inf(F)$
  - Rabin: $\bigvee_{i=1}^{n} Fin(B_i) \wedge Inf(G_i)$

Emerson-Lei Automata

Pros

- A very flexible acceptance condition
- Boolean operations on deterministic automata are trivial

Cons

- For nondeterministic automata: Doubly-exponential complementation and determinization
- In general: Nonemptiness check is NP-complete

Bottom line

- Interesting in some settings
- Often too much of a price for the extra flexibility

Some other (re)New(ed) Types

- Generalized-Rabin [J. Kretínský and J. Esparza, 2012] :

$$\bigvee_{i=1}^{n} Fin(B_i) \wedge Inf(G_{i1}) \wedge Inf(G_{i2}) \wedge \cdots \wedge Inf\left(G_{i\_ki}\right)$$

- Hyper-Rabin ("canonical form" in [Emerson&Lei, 1987]):

$$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} Fin(B_{i,j}) \vee Inf(G_{i,j})$$

- Generalized-Streett [F. Blahoudek, 2012]:

$$\bigwedge_{i=1}^{n} Inf(G_i) \vee Fin(B_{i1}) \vee Fin(B_{i2}) \vee \cdots \vee Fin(B_{i\_ki})$$

- Hyper-Streett:

$$\bigwedge_{i=1}^{n} \bigvee_{j=1}^{m} Fin\left(B_{i,j}\right) \wedge Inf\left(G_{i,j}\right)$$

The Hyper- further generalize the Generalized-.

Thm. Hyper-Rabin/Streett generalize all the classic conditions; There is an exponential size blowup in the other direction, for deterministic automata.

Connection to Verification

- The generalized- and hyper-Rabin conditions naturally occur in the translation of various fragments of LTL into automata.
  - [K. Chatterjee, A. Gaiser, J. Kretínský, 2012]; [J. Esparza, J. Kretínský, Sickert, 2016]

- The generalized- and hyper-Streett conditions naturally occur in n-player omega-regular games.
  - [E. Filiot, R. Gentilini, J. F. Raskin, 2018]

Costs

Deterministic automata

	Union	Intersect.	Complement.	Nonemptiness
Hyper-Rabin				
Hyper-Streett				
Emerson-Lei	Quadratic		No blowup	NP-Complete

- Generalized-Rabin/Streett have the same costs as Hyper-Rabin/Streett.

Costs

Deterministic automata

	Union	Intersect.	Complement.	Nonemptiness
Hyper-Rabin	Quadratic			
Hyper-Streett				
Emerson-Lei			No blowup	NP-Complete

- Generalized-Rabin/Streett have the same costs as Hyper-Rabin/Streett.

Costs

Deterministic automata

<table>
<tr><th></th><th>Union</th><th>Intersect.</th><th>Complement.</th><th>Nonemptiness</th></tr>
<tr><td>Hyper-Rabin</td><td colspan="2" rowspan="3">Quadratic</td><td rowspan="2">Exponential</td><td></td></tr>
<tr><td>Hyper-Streett</td><td></td></tr>
<tr><td>Emerson-Lei</td><td>No blowup</td><td>NP-Complete</td></tr>
</table>

- Generalized-Rabin/Streett have the same costs as Hyper-Rabin/Streett.

Costs

Deterministic automata

	Union	Intersect.	Complement.	Nonemptiness
Hyper-Rabin	Quadratic		Exponential	PTIME
Hyper-Streett				NP-Complete
Emerson-Lei			No blowup	

- Generalized-Rabin/Streett have the same costs as Hyper-Rabin/Streett.

So: Hyper-Rabin has great potential for compound systems.

  - Already partially fulfilled with generalized-Rabin [K. Chatterjee, A. Gaiser, J. Kretínský, 2012]; [J. Esparza, J. Kretínský, Sickert, 2016]

Costs

Deterministic automata

<table>
<tr><th></th><th>Union</th><th>Intersect.</th><th>Complement.</th><th>Nonemptiness</th></tr>
<tr><td>Hyper-Rabin</td><td colspan="2" rowspan="3">Quadratic</td><td rowspan="2">Exponential</td><td>PTIME</td></tr>
<tr><td>Hyper-Streett</td><td rowspan="2">NP-Complete</td></tr>
<tr><td>Emerson-Lei</td><td>No blowup</td></tr>
</table>

But, can we hope for more?

Costs

Deterministic automata

<table>
<tr><th></th><th>Union</th><th>Intersect.</th><th>Complement.</th><th>Nonemptiness</th></tr>
<tr><td>Hyper-Rabin</td><td colspan="2" rowspan="3">Quadratic</td><td rowspan="2">Exponential</td><td>PTIME</td></tr>
<tr><td>Hyper-Streett</td><td rowspan="2">NP-Complete</td></tr>
<tr><td>Emerson-Lei</td><td>No blowup</td></tr>
<tr><td>?</td><td colspan="3">Polynomial</td><td>PTIME</td></tr>
</table>

But, can we hope for more?

A deterministic automaton type that allows for polynomial boolean operations, <u>including complementation</u>, and PTIME decision procedures (including universality)?

Costs

Deterministic automata

<table>
<tr><th></th><th>Union</th><th>Intersect.</th><th>Complement.</th><th>Nonemptiness</th></tr>
<tr><td>Hyper-Rabin</td><td colspan="2" rowspan="3">Quadratic</td><td rowspan="2">Exponential</td><td>PTIME</td></tr>
<tr><td>Hyper-Streett</td><td rowspan="2">NP-Complete</td></tr>
<tr><td>Emerson-Lei</td><td>No blowup</td></tr>
<tr><td>Hyper-dual</td><td colspan="3">Polynomial</td><td>PTIME</td></tr>
</table>

But, can we hope for more?

A deterministic automaton type that allows for polynomial boolean operations, <u>including complementation</u>, and PTIME decision procedures (including universality)? YES!

Hyper-dual

- At first, it might seem to be just a dirty trick.
- At a second look, it may be very interesting…

Hyper-dual = A pair of hyper-Rabin and hyper-Streett automata for the same language.

Hyper-Rabin	Hyper-Streett

For deterministic automata, it is the same as having a pair of Hyper-Rabin automata for both L and its complement L^c.

Are You Kidding?

Maintaining automata for both L and L^c is pure redundancy!

- Use a single hyper-Rabin and complement it on demand.

Yes: It indeed reveals an interesting property of Hyper-Rabin.

And No:

- The complementation procedure is exponential, and does not guarantee the smallest possible hyper-Streett automaton.
- Often, automata generation is iterative, and with hyper-dual automaton we avoid complementation: We may start with two equal copies of a Rabin automaton.
- Targeting a hyper-dual automaton, we may use properties that are succinctly expressed by it.

A General Trick?

So, is it in general beneficial to maintain two copies of deterministic automata over dual acceptance conditions?

- For example, Rabin and Streett automata?

No.

- It obviously allows for "free" complementation, yet it might have a price in union and intersection.
- A pair of deterministic Rabin and Streett automata has an exponential size blowup on both union and intersection.
- Hyper-dual is strong enough to prevent this price, and not too strong for preserving decision problems in PTIME.

Hyper-dual Costs

The promising properties:

- Succinctness: Not more than (twice) Rabin and Street, and sometimes exponentially less than them.
- Boolean operations: Complementation with no blowup; union and intersection with quadratic blowup.
- Decision problems: All in PTIME – nonemptiness, universality, equivalence, and containment.

Containment of Det. Hyper-Dual

- Consider deterministic hyper-dual automata $C=(A, B)$ and $C' =(A', B')$.
- Then $L(C) \subseteq L(C')$ iff $L(A) \cap L(B')^C = \emptyset$.
- Observe that $L(B')^C = L(B'^C)$ and that B'^C is a hyper-Rabin automaton.
- Thus, we intersect two hyper-Rabin automata in quadratic time, and check emptiness in PTIME.

Conclusions

- Automata on infinite words deserve many types.
  - Each is related to other interesting formalisms, and has its pros and cons.
- An organized picture: http://www.faculty.idc.ac.il/udiboker/automata
- There is still place to look into additional new types.
- Hyper-Rabin/Streett/dual automata look interesting for further exploration.