TRANSITION-BASED ACCEPTANCE VS STATE-BASED ACCEPTANCE

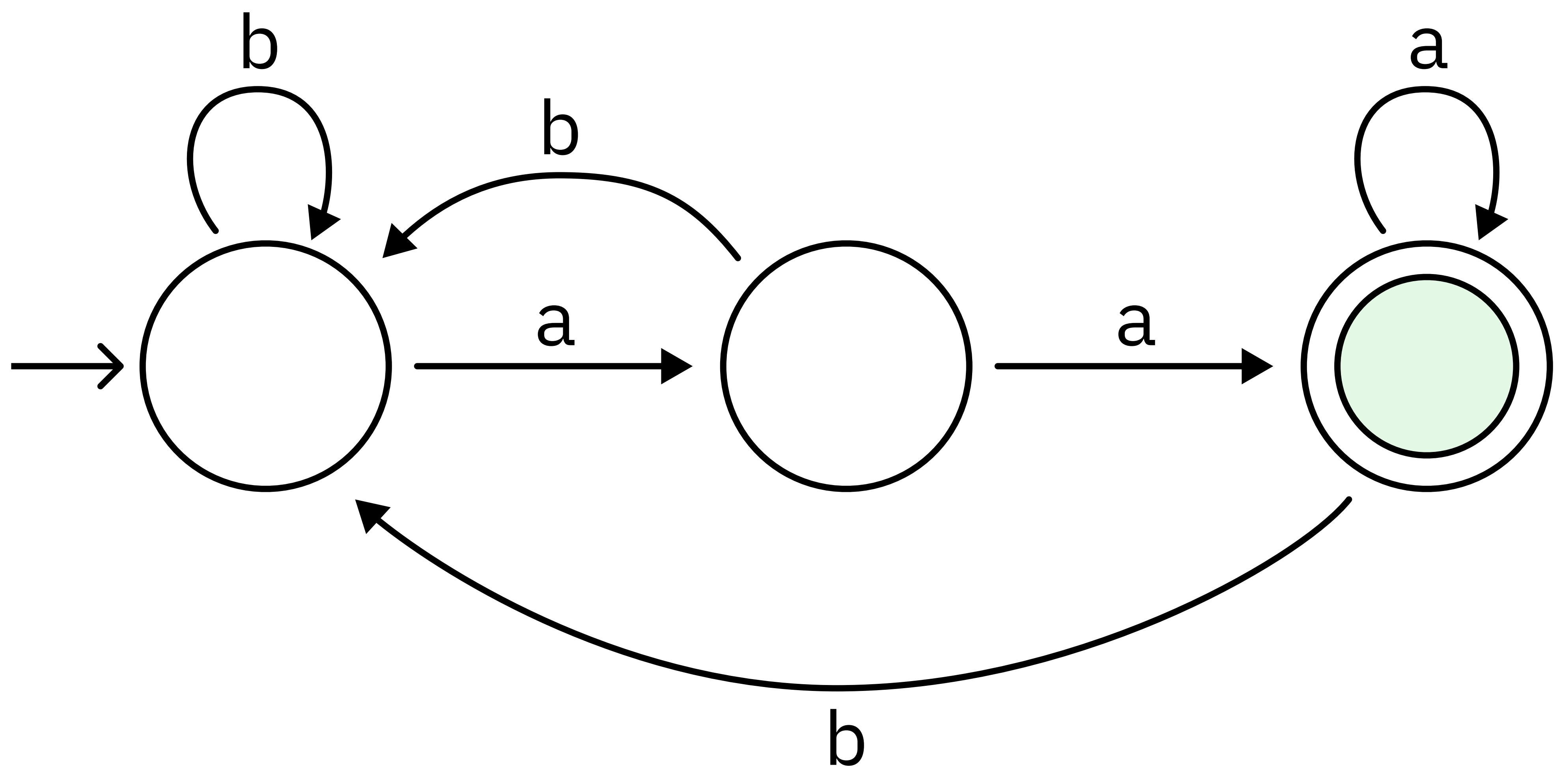
FOR ω-AUTOMATA

Link to the survey

Antonio Casares Santos

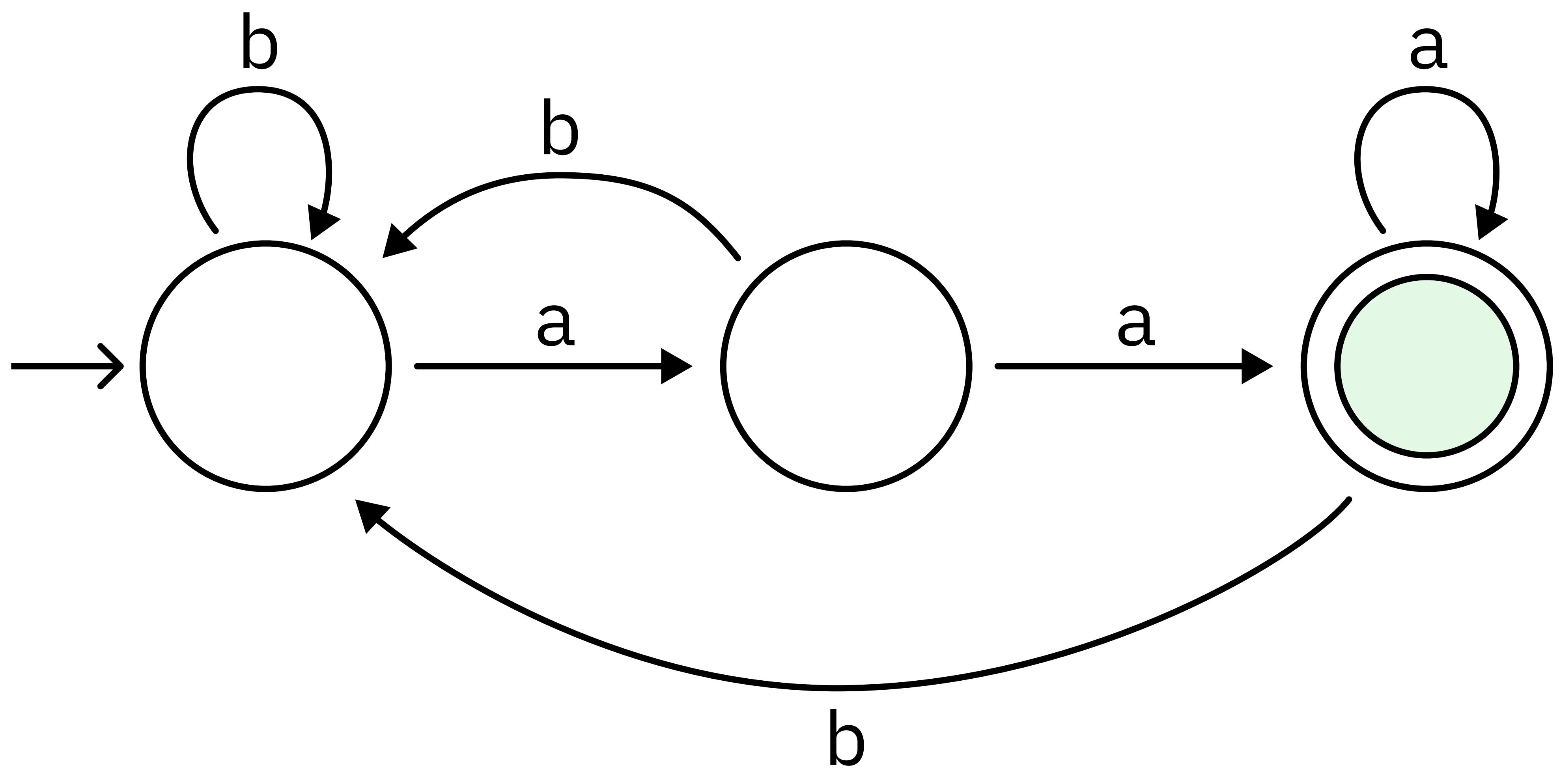
RPTU Kaiserslautern

An automaton



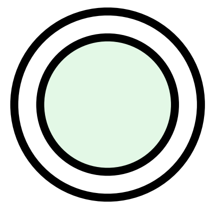
$\mathcal{L}(\mathcal{A}) =$ Words ending in 'aa'

An ω-automaton



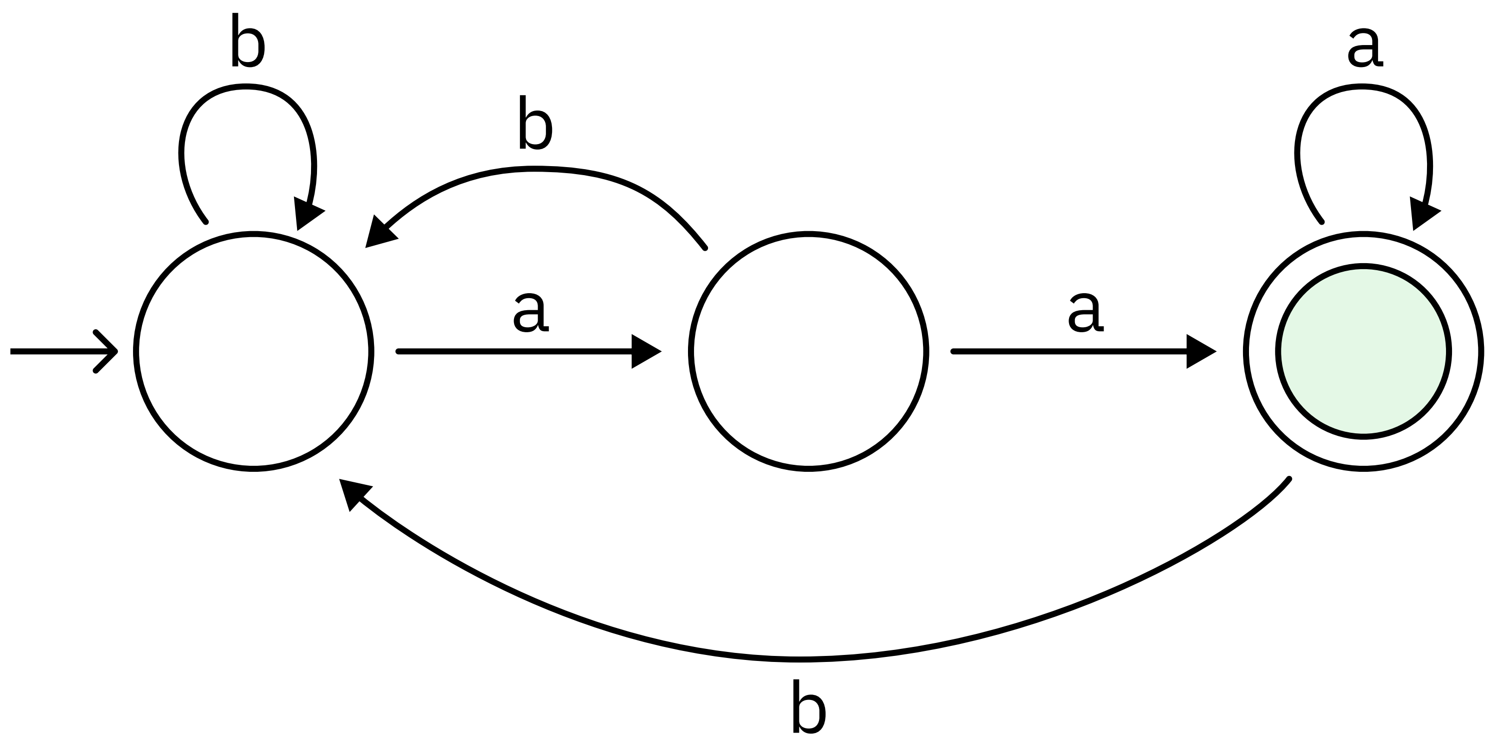
$\mathcal{L}(\mathcal{A}) =$ Words containing '*aa*' infinitely often $\subseteq \Sigma^\omega$

Input: Infinite words $w = abaabbaaa\ldots \in \Sigma^\omega$

Büchi condition: We accept if  visited infinitely often

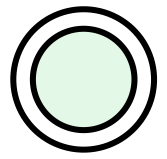
Why should we care?

An ω-automaton



$\mathcal{L}(\mathcal{A}) =$ Words containing '*aa*' infinitely often $\subseteq \Sigma^\omega$

Input: Infinite words $w = abaabbaaa... \in \Sigma^\omega$

Büchi condition: We accept if  visited infinitely often

Why should we care?

Some historical context

1960

1980

2000

2020

Büchi 1962

The monadic second-order (MSO) theory of $(\mathbb{N}, <)$

monadic second-order (MSO) t

- First order logic connectives $\neg\varphi \quad \varphi \vee \psi \quad \varphi \wedge \psi \quad \exists x \quad \forall x$

- Quantification over sets $\exists X \quad \forall X \quad x \in X$

$\exists X \, \forall y \exists x \;\; x \in X \;\wedge\; x > y \quad \longleftrightarrow \quad$ There is an unbounded set

In MSO($\mathbb{N}$, <) we can express divisibility by a given *n*, basic modular arithmetic.

The monadic second-order (MSO) theory of $(\mathbb{N}, <)$

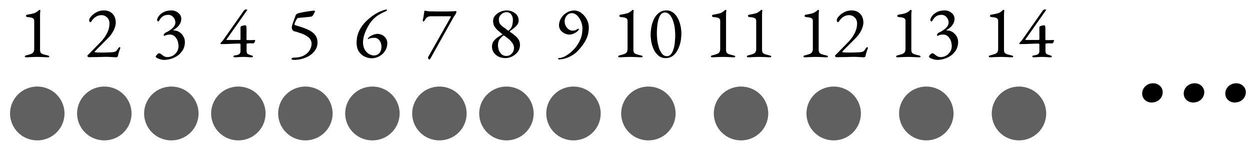
Büchi 1962

- First order logic connectives $\neg\varphi \quad \varphi \vee \psi \quad \varphi \wedge \psi \quad \exists x \quad \forall x$
- Quantification over sets $\exists X \quad \forall X \quad x \in X$

$\exists X\ \forall y \exists x\ \ x \in X\ \wedge\ x > y \quad \longleftrightarrow \quad$ There is an unbounded set

In MSO($\mathbb{N}$, <) we can express divisibility by a given *n*, basic modular arithmetic.

$(\mathbb{N}, <) \quad =$



Infinite word over the unary alphabet {●}

Büchi 1962

The monadic second-order (MSO) theory of $(\mathbb{N}, <)$

- First order logic connectives $\neg\varphi$ $\varphi \vee \psi$ $\varphi \wedge \psi$ $\exists x$ $\forall x$
- Quantification over sets $\exists X$ $\forall X$ $x \in X$

$\exists X\ \forall y \exists x\ \ x \in X\ \wedge\ x > y$ $\longleftrightarrow$ There is an unbounded set

In MSO($\mathbb{N}$, <) we can express divisibility by a given n, basic modular arithmetic.

$(\mathbb{N}, <)$ = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 ●●●●●●●●●●●●●● ...

Infinite word over the unary alphabet {●}

Introduced Büchi automata

- Non-deterministic Büchi automata ≡ MSO logic

$$(\mathbb{N}, <) \models \varphi \iff L(\mathcal{A}_\varphi) \neq \emptyset$$

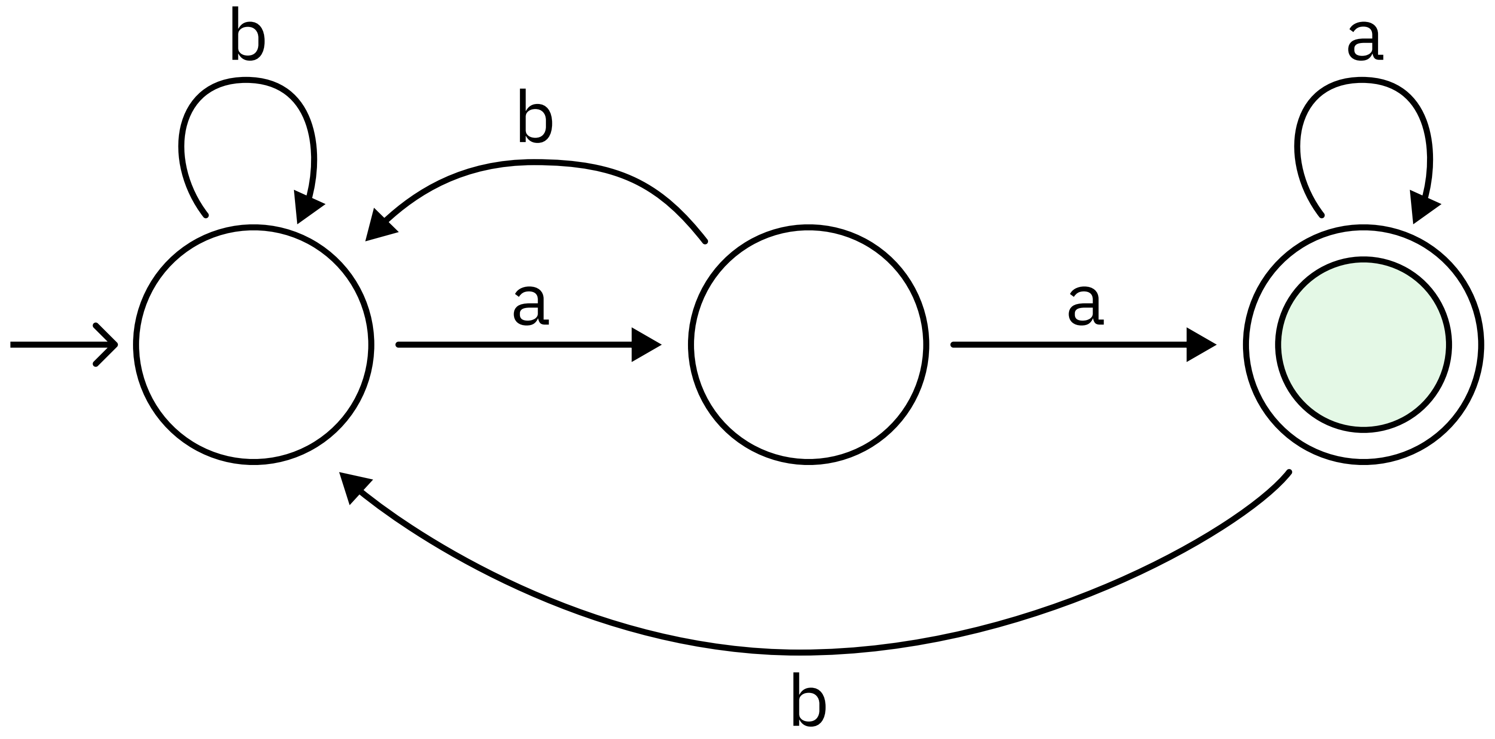
ω-regular languages

- Obtained decidability of MSO$(\mathbb{N}, <)$

Given a MSO formula φ, is φ true in $(\mathbb{N}, <)$?

Some historical context

An ω-automaton

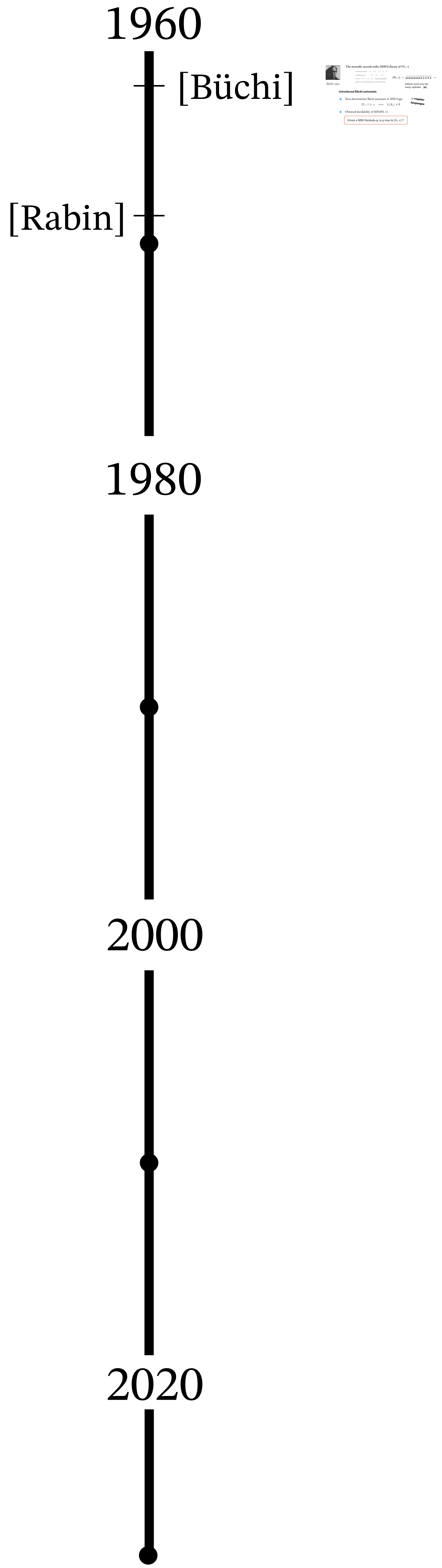


$\mathcal{L}(\mathcal{A}) =$ Words containing '*aa*' infinitely often $\subseteq \Sigma^\omega$

Input: Infinite words $w = abaabbaaa... \in \Sigma^\omega$

Büchi condition: We accept if ◎ visited infinitely often

Why should we care?



Rabin 1969

The MSO theory of $(\mathbb{Q}, <)$ and the full binary tree

Extremely powerful!

Extremely complex proof!!

- Obtained decidability of MSO$(\mathbb{Q}, <)$ and MSO(*infinite binary tree*).

Rabin 1969

The MSO theory of $(\mathbb{Q}, <)$ and the full binary tree

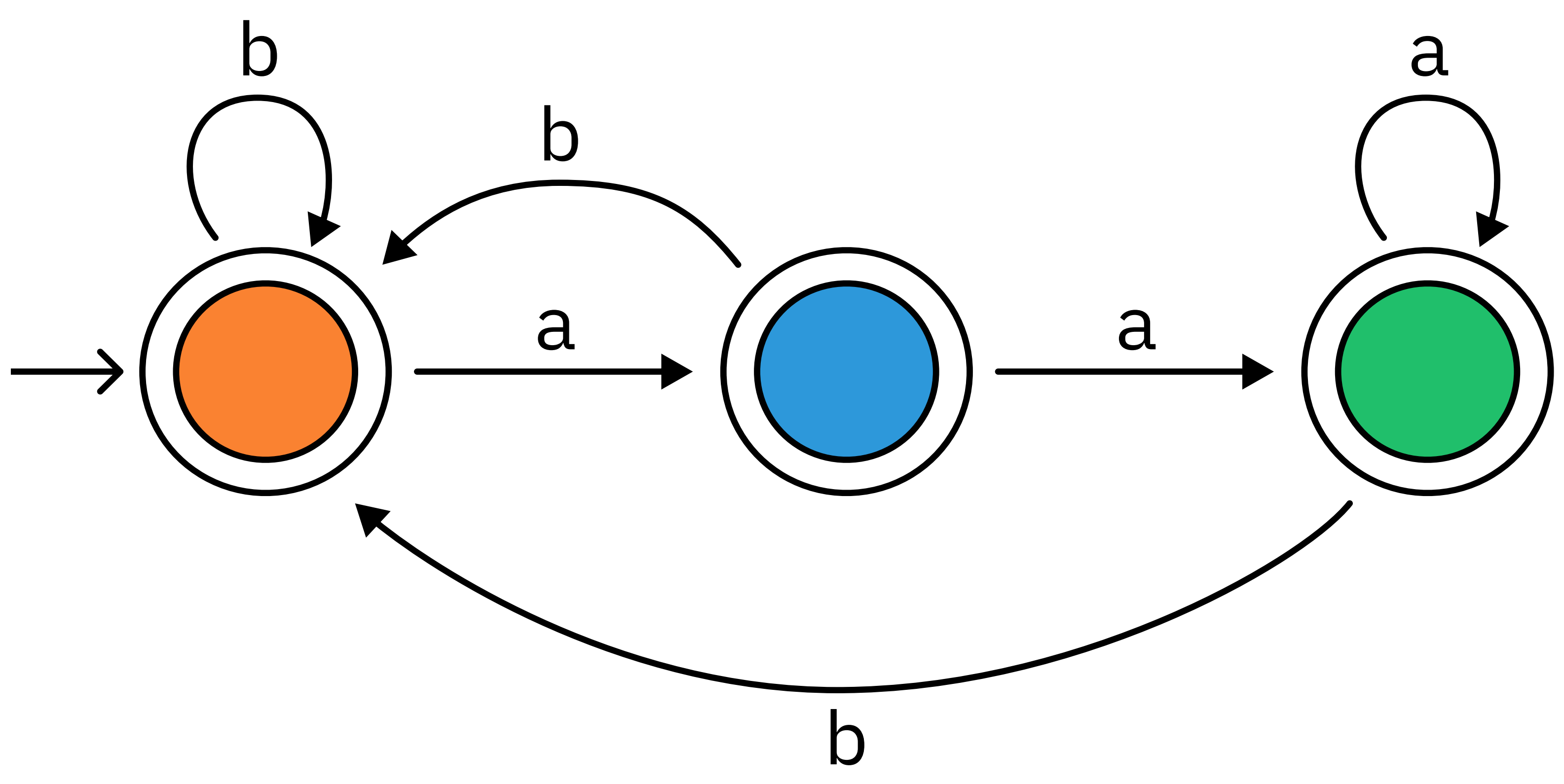
Extremely powerful!

Extremely complex proof!!

- Obtained decidability of MSO$(\mathbb{Q}, <)$ and MSO(*infinite binary tree*).

Introduced automata over infinite trees

Introduced richer acceptance conditions



Boolean combination of states appearing infinitely many times

`Inf.Often(●) or Fin.Often(●)`

(Muller condition)

Muller 1963, McNaughton 1966

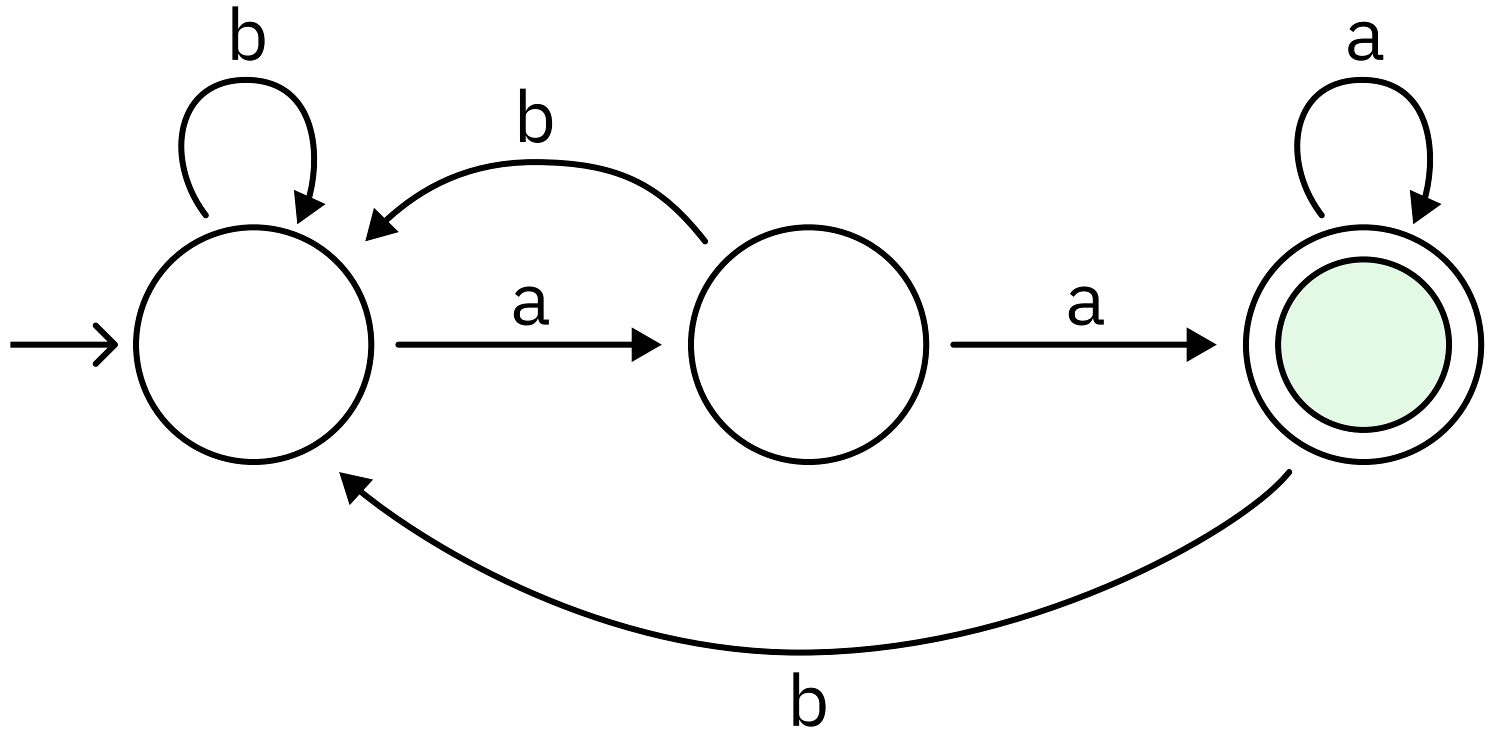
Rabin condition: A sort of simple DNF for these formulas

Necessary for using:

- Deterministic ω-automata
- Automata over infinite trees

Some historical context

An ω-automaton

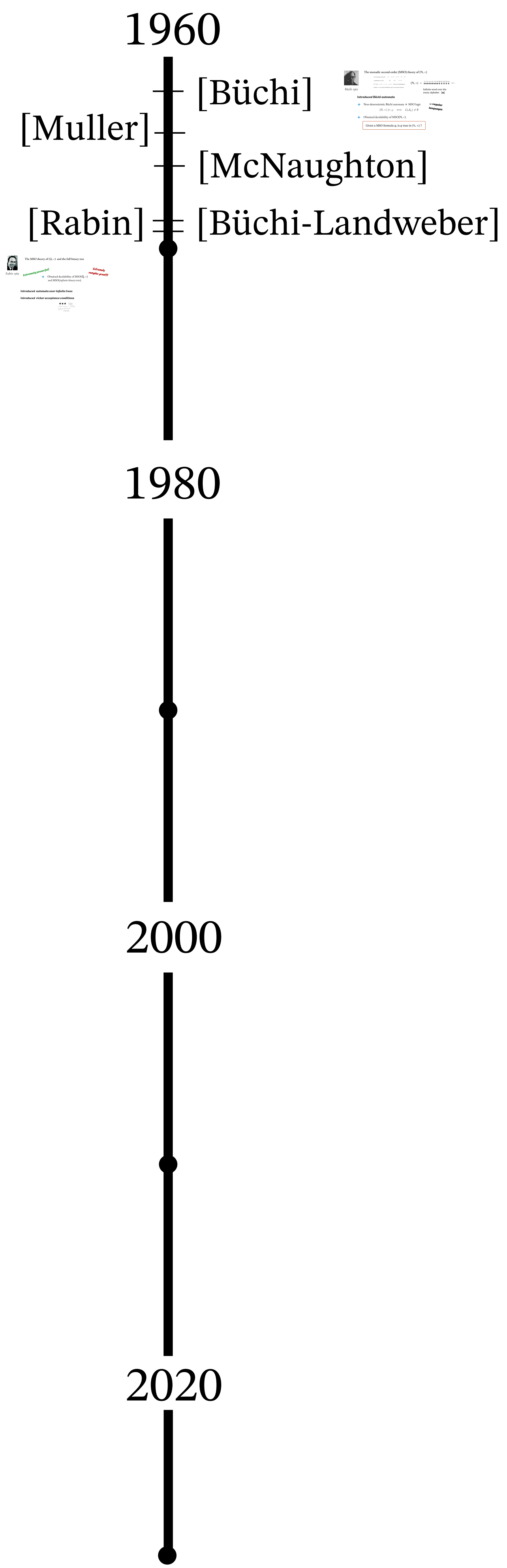


$\mathcal{L}(\mathcal{A}) =$ Words containing '*aa*' infinitely often $\subseteq \Sigma^\omega$

Input: Infinite words $w = abaabbaaa\ldots \in \Sigma^\omega$

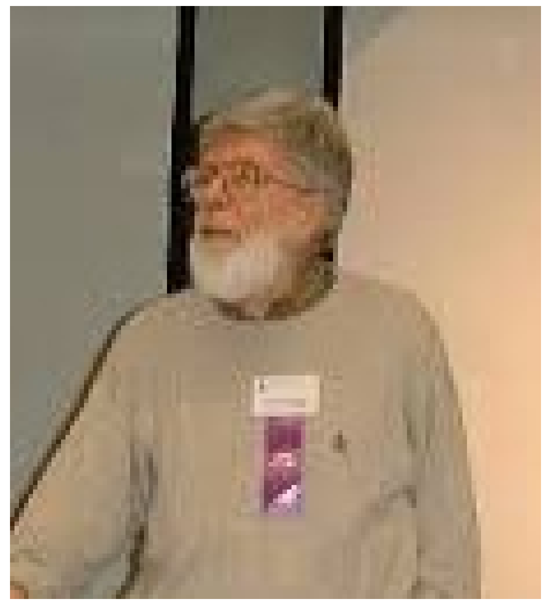
Büchi condition: We accept if ◎ visited infinitely often

Why should we care?



Church's synthesis problem for MSO

Büchi-Landweber 1969



McNaughton 1966

Church's synthesis problem for MSO

I alphabet of input symbols

O alphabet of output symbols

φ a specification over sequences in $(IO)^\omega$

$i_1\ i_2\ i_3 \ldots$ *complete it on-the-fly* $\longrightarrow$ $i_1 o_1 i_2 o_2 \cdots \in (IO)^\omega$

stream of input symbols — satisfying φ

Church synthesis problem

Given φ, decide whether there is a finite-state program (circuit, transducer) producing outputs on-the-fly, ensuring that φ is satisfied.

Church's synthesis problem for MSO

Büchi-Landweber 1969

I alphabet of input symbols

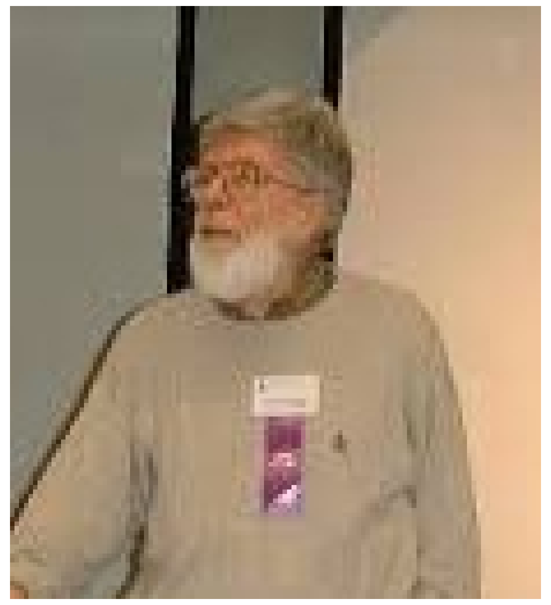
O alphabet of output symbols

φ a specification over sequences in $(IO)^\omega$

$i_1\ i_2\ i_3 \ldots$ stream of input symbols $\xrightarrow{\textit{complete it on-the-fly}}$ $i_1 o_1 i_2 o_2 \cdots \in (IO)^\omega$ satisfying φ

Church synthesis problem

Given φ, decide whether there is a finite-state program (circuit, transducer) producing outputs on-the-fly, ensuring that φ is satisfied.

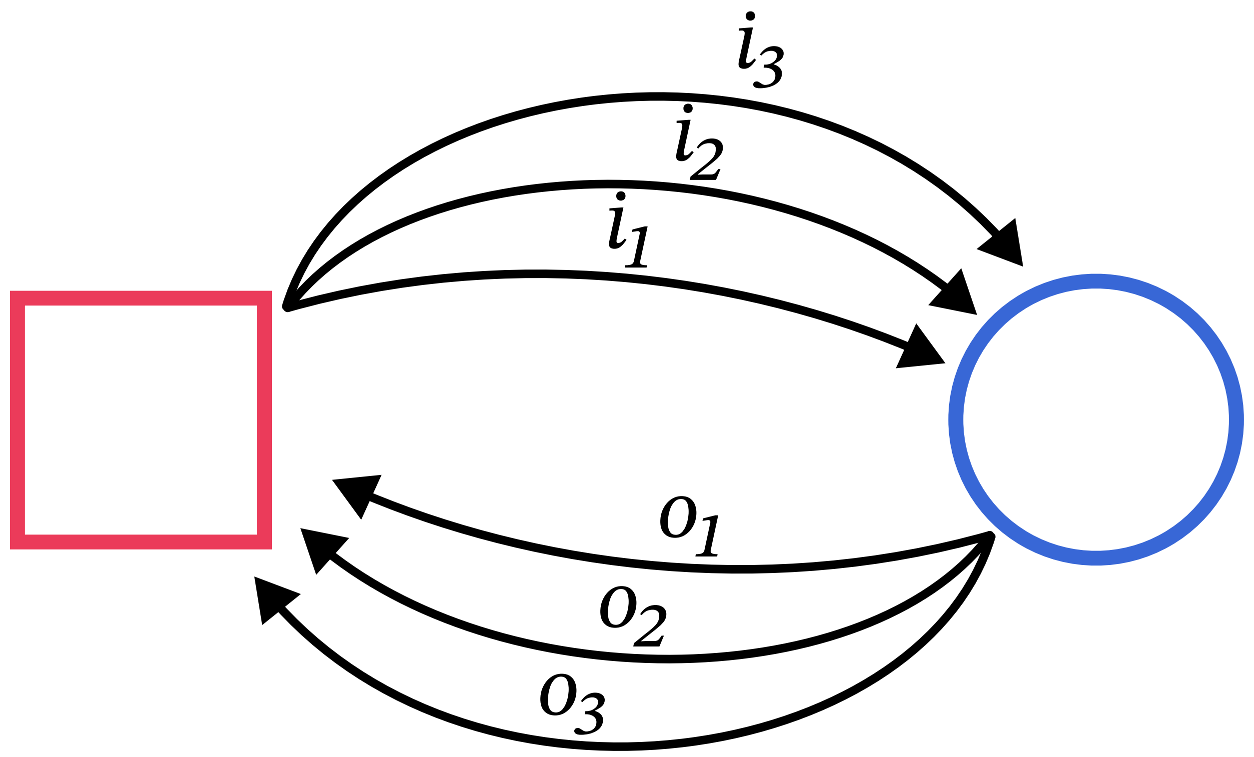


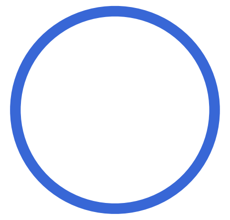
McNaughton 1966

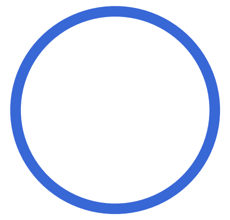
- Decidability of the synthesis problem for specifications in MSO

Using games on graphs

games on graphs

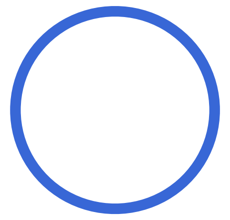


Player  wins if the final output satisfies φ

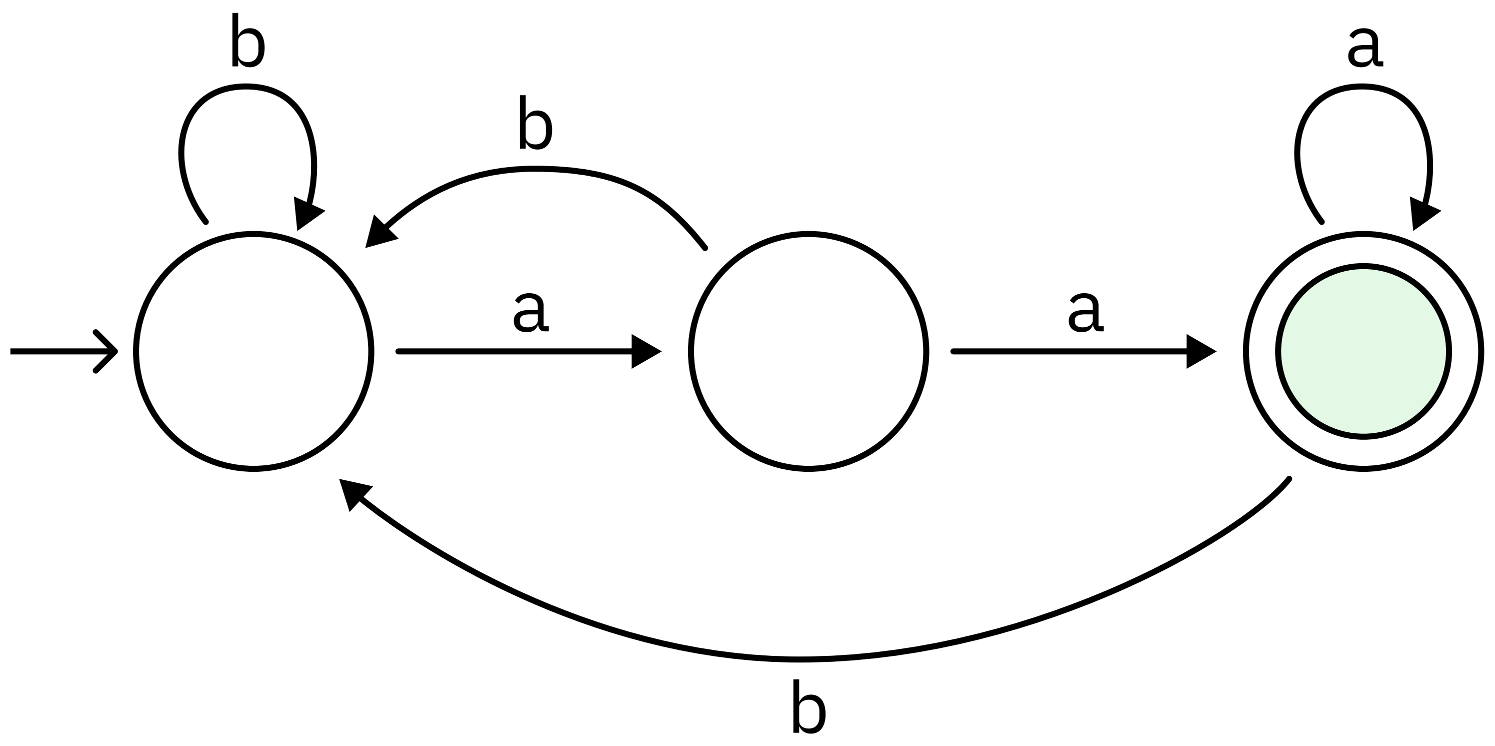
Winning strategy for  $\longleftrightarrow$ Program for Church's problem

Theorem

If $\varphi \in$ MSO (i.e., ω-regular), these games are determined and the winner has a strategy given by a finite automaton.

It is decidable if  can win.

An ω-automaton



$\mathcal{L}(\mathcal{A}) =$ Words containing '*aa*' infinitely often $\subseteq \Sigma^\omega$

Input: Infinite words $w = abaabbaaa... \in \Sigma^\omega$

Büchi condition: We accept if ◎ visited infinitely often

Why should we care?

Some historical context

*Extremely partial account of works

1960
- [Büchi]
- [Muller]
- [McNaughton]
- [Rabin]
- [Büchi-Landweber]
- [Staiger-Wagner]
- [Wagner]
- [Ehrenfeucht-Mycielski]

1980
- [Gurevich-Harrington]
- [Mostowski]
- [Emerson-Jutla]

2000

2020

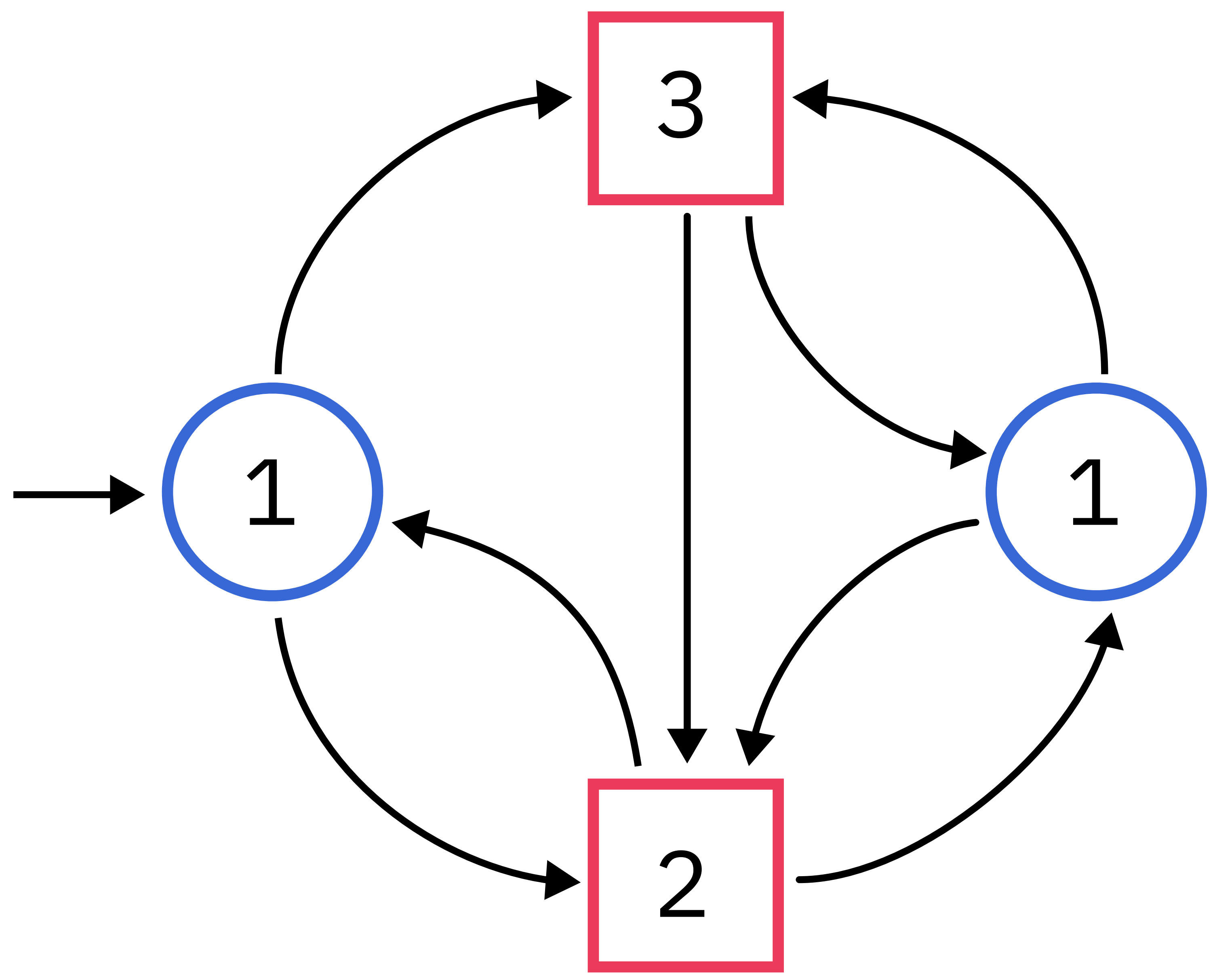
Gurevich-Harrington 1982

Mostowski 1984

Emerson-Jutla 1991

- Simpler proof of Rabin's theorem using game-theoretic ideas
- Parity condition

Parity condition



Numbers in states/vertices

○ wins if the maximal number appearing infinitely often is even

✦ "Normal form" for Rabin conditions

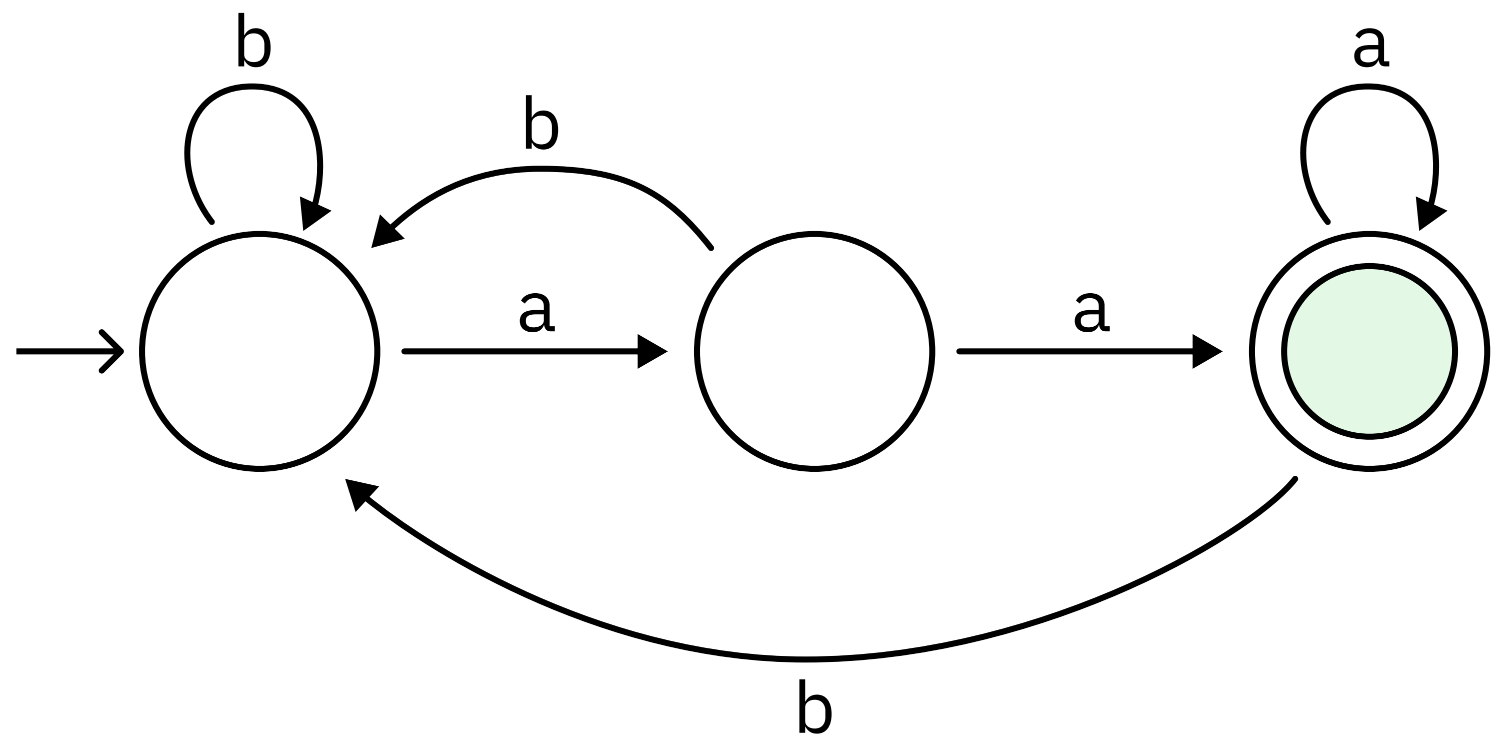
Simplest condition for recognizing all ω-reg. languages using deterministic automata

THEOREM (positional determinacy of parity games)

In a parity game, the winner has a *positional* strategy

strat: Vertices → Edges

An ω-automaton



$\mathcal{L}(\mathcal{A}) =$ Words containing 'aa' infinitely often $\subseteq \Sigma^\omega$

Input: Infinite words $w = abaabbaaa... \in \Sigma^\omega$

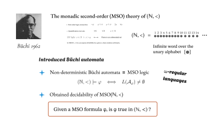
Büchi condition: We accept if ◎ visited infinitely often

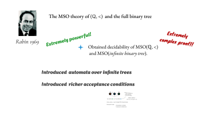
Why should we care?

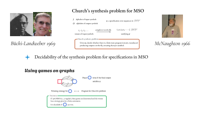
Some historical context

*Extremely partial account of works

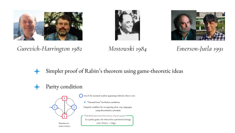
1960
- [Büchi]
- [Muller]
- [McNaughton]
- [Rabin]
- [Büchi-Landweber]
- [Staiger-Wagner]
- [Wagner]
- [Ehrenfeucht-Mycielski]

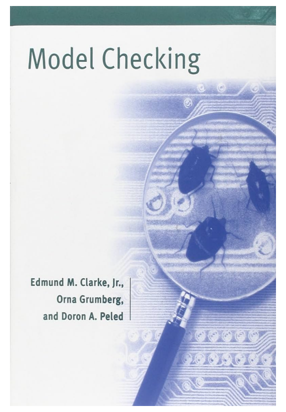






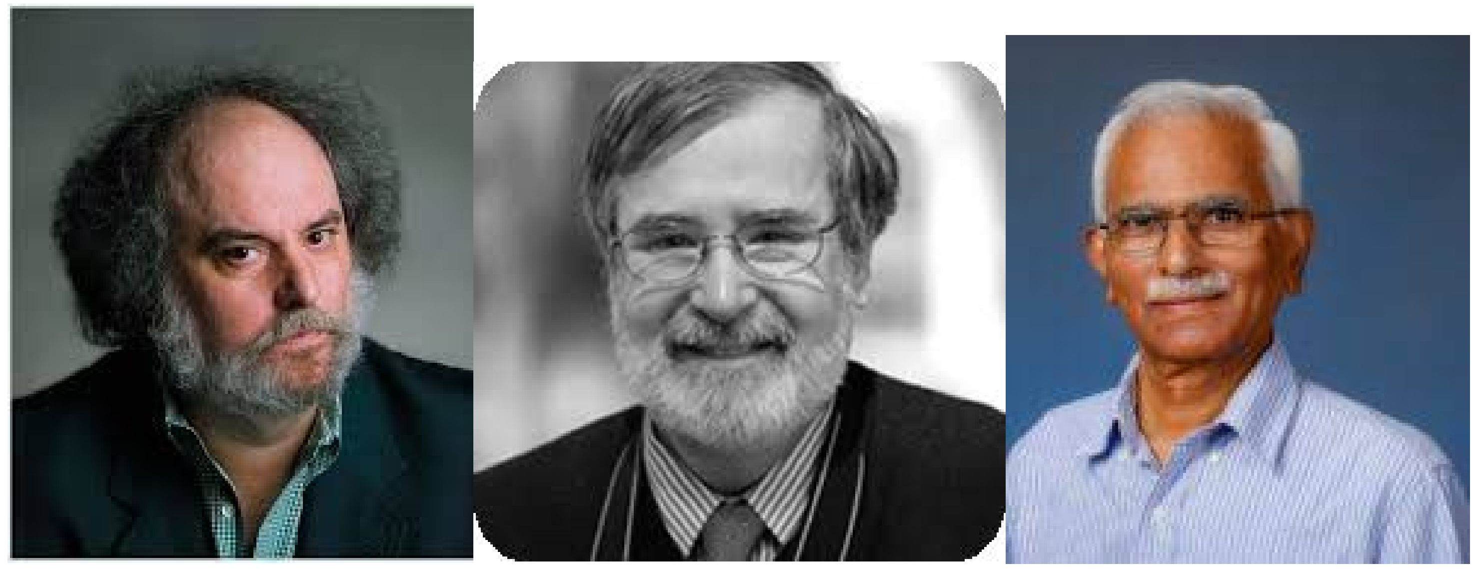
1980
- [Gurevich-Harrington]
- [Clarke-Emerson-Sistla]
- [Mostowski]
- [Streett-Emerson]
- [Vardi-Wolper]
- [Safra]
- [Pnueli-Rosner]
- [Emerson-Jutla]





2000

2020



Emerson-Clarke-Sistla 1983

Sifakis 1982

Vardi-Wolper 1986

Model checking

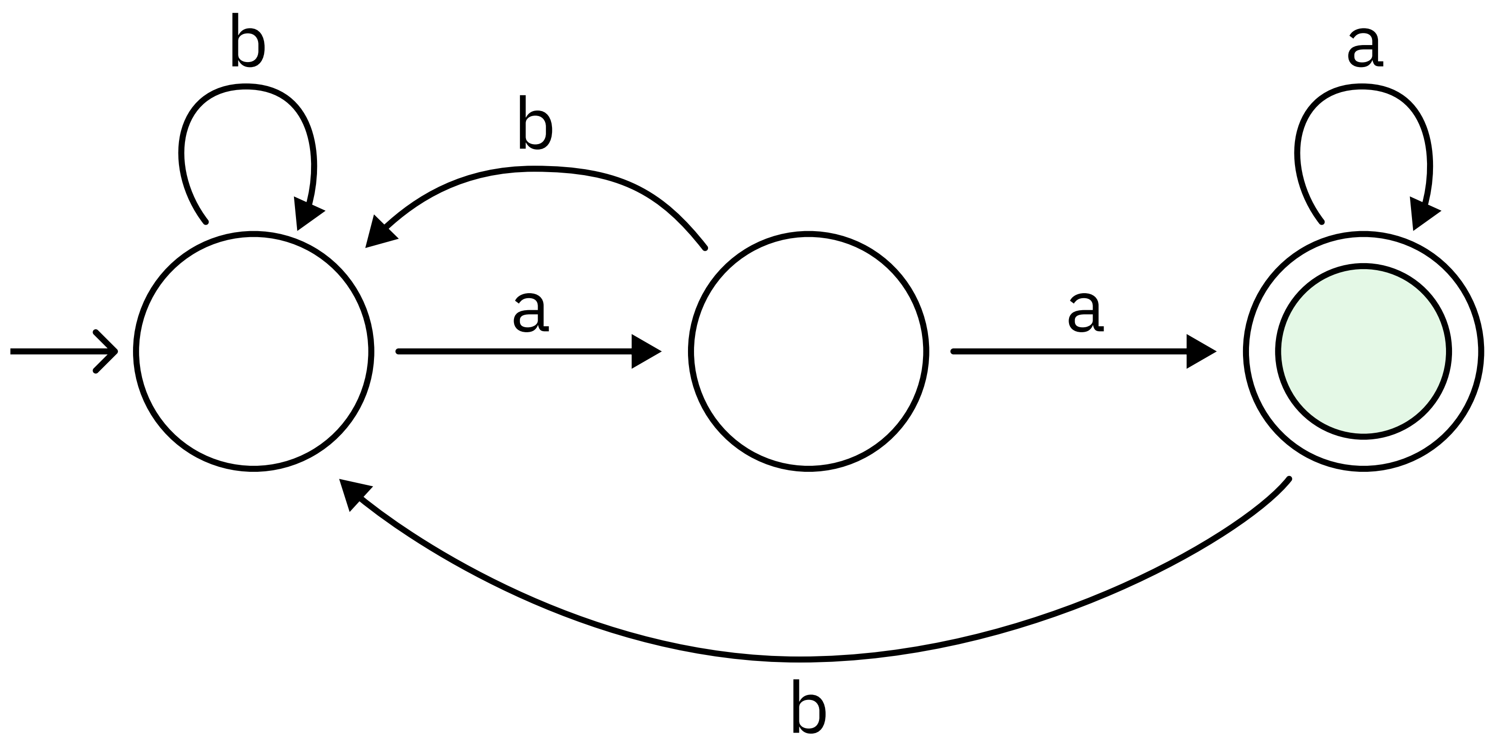
Does a program satisfy a given specification?

Pnueli-Rosner 1989

Efficient *Linear Temporal Logic* synthesis

Given a specification, build a program that satisfies it.

An ω-automaton



$\mathcal{L}(\mathcal{A})$ = Words containing '*aa*' infinitely often $\subseteq \Sigma^\omega$

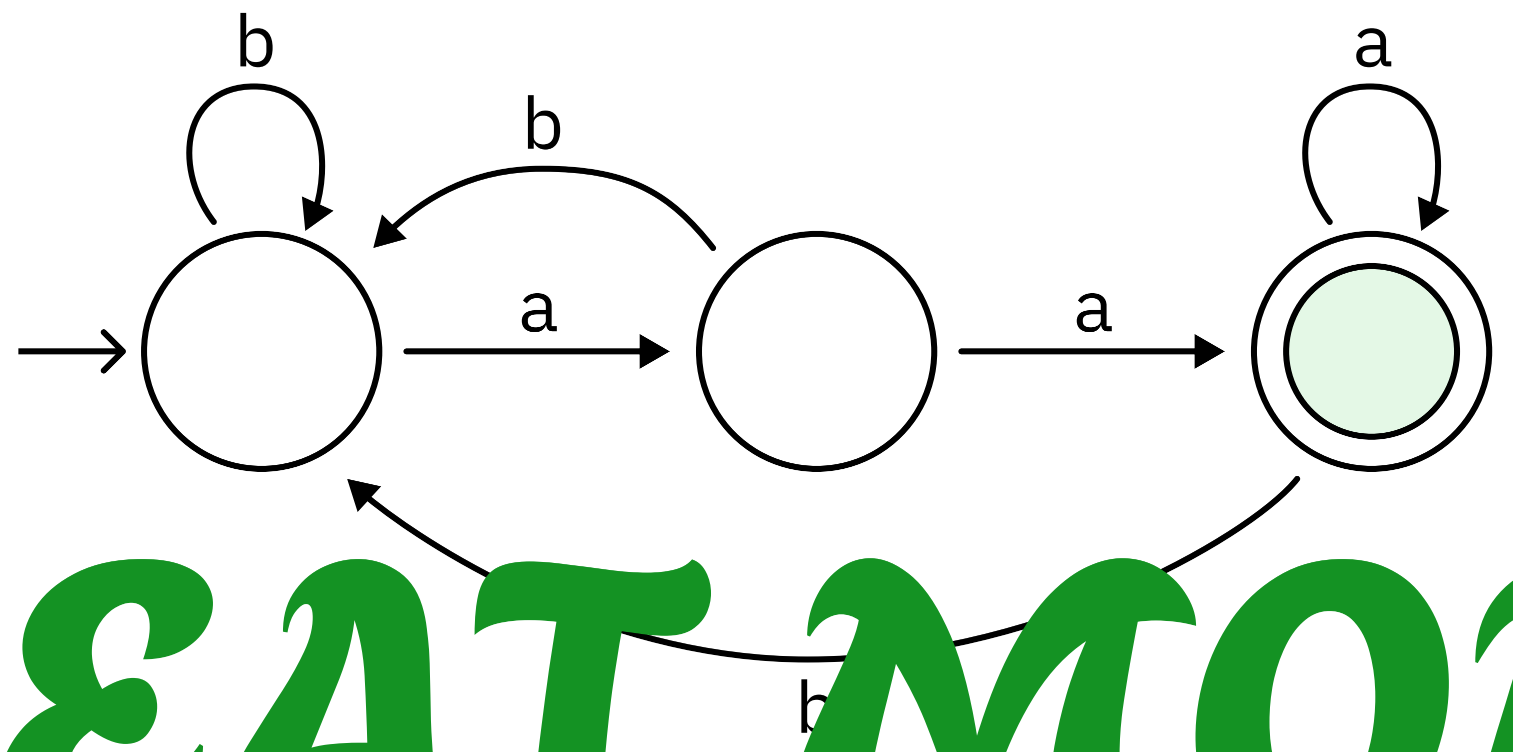
Input: Infinite words $w = abaabbaaa... \in \Sigma^\omega$

Büchi condition: We accept if ◯ visited infinitely often

Why should we care?

Some historical context

An ω-automaton



GREAT MODEL

$\mathcal{L}(\mathcal{A})$ = Words containing aa infinitely often $\subseteq \Sigma^\omega$

Input: Infinite words $w = abaabbaaa\ldots \in \Sigma^\omega$

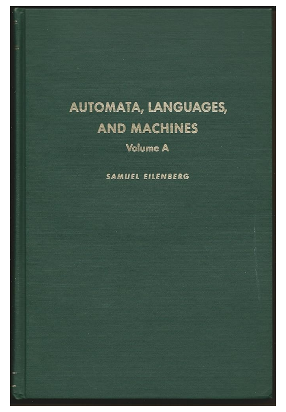
Büchi condition: We accept if ◎ visited infinitely often

Why should we care?

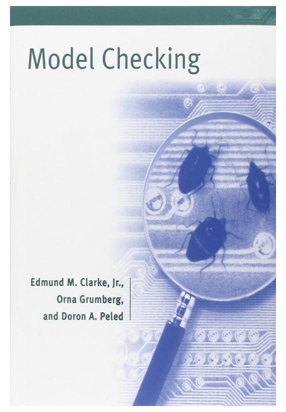
Some historical context

*Extremely partial account of works

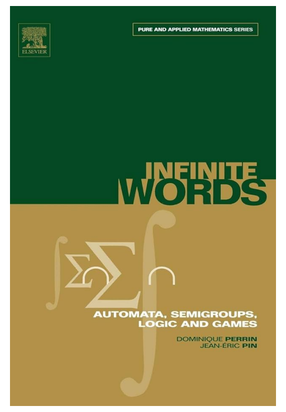
1960
[Büchi]
[Muller]
[McNaughton]
[Rabin] [Büchi-Landweber]
[Staiger-Wagner]
[Wagner] [Ehrenfeucht-Mycielski]

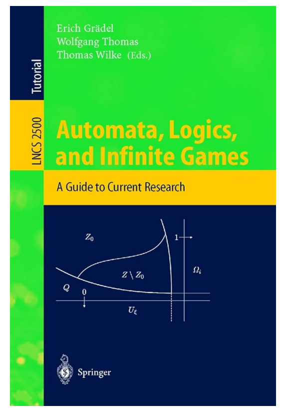


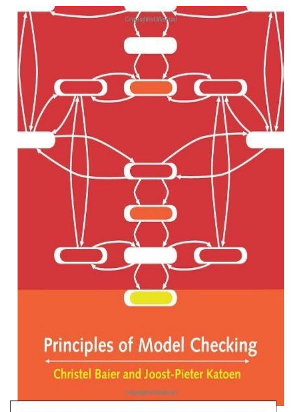
1980
[Gurevich-Harrington] [Clarke-Emerson-Sistla]
[Mostowski] [Streett-Emerson]
[Vardi-Wolper]
[Safra]
[Pnueli-Rosner]
[Emerson-Jutla] [Wilke]
[Puri] [Walukiewicz]
[Zwick-Paterson] [Jurdziński]
[Niwinski-Walukiewicz] [Zielonka]

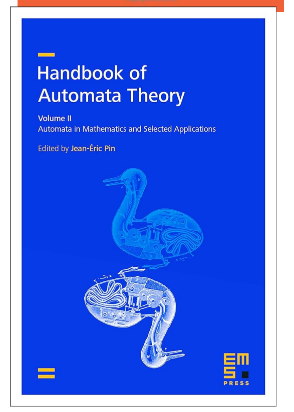


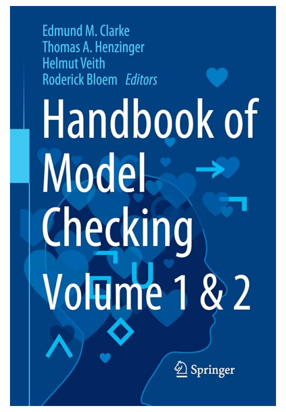
2000
[Löding]
[Kupferman-Vardi]
[Henzinger-Piterman]





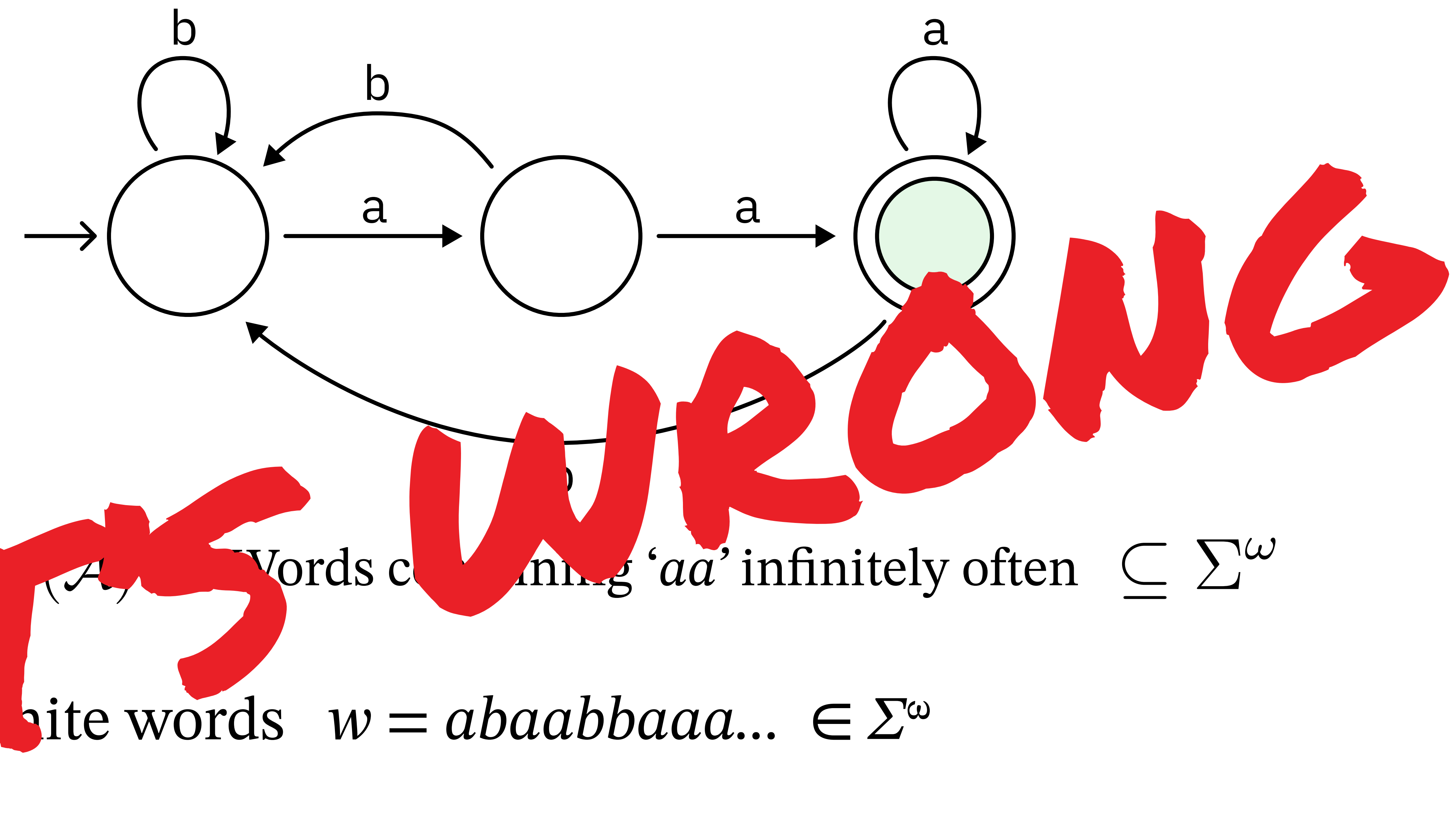






[Many, many, works]
2020

An ω-automaton



(A) Words c… ining 'aa' infinitely often $\subseteq \Sigma^\omega$

Inpu Ini ite words $w = abaabbaaa\ldots \in \Sigma^\omega$

Büchi condition: We accept if ◎ visited infinitely often

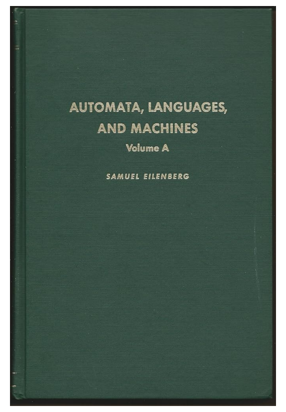
Why should we care?

Some historical context

*Extremely partial account of works

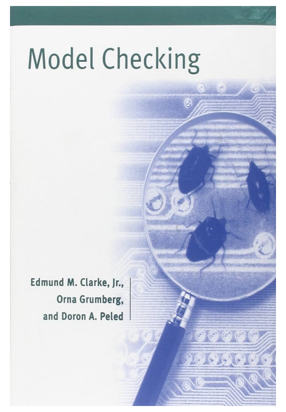
1960

[Büchi]
[Muller]
[McNaughton]
[Rabin] [Büchi-Landweber]
[Staiger-Wagner]
[Wagner] [Ehrenfeucht-Mycielski]



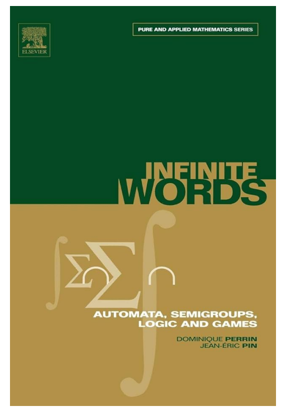
1980

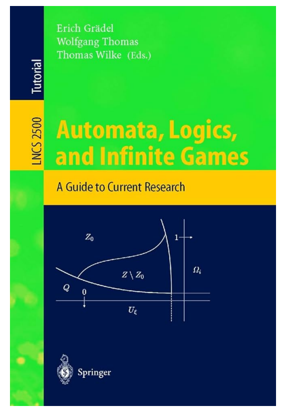
[Gurevich-Harrington] [Clarke-Emerson-Sistla]
[Mostowski] [Streett-Emerson]
[Vardi-Wolper]
[Safra]
[Pnueli-Rosner]
[Emerson-Jutla] [Wilke]
[Puri] [Walukiewicz]
[Zwick-Paterson] [Jurdziński]
[Niwinski-Walukiewicz] [Zielonka]

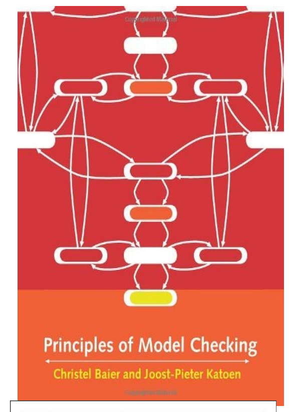


2000

[Löding]
[Kupferman-Vardi]
[Henzinger-Piterman]

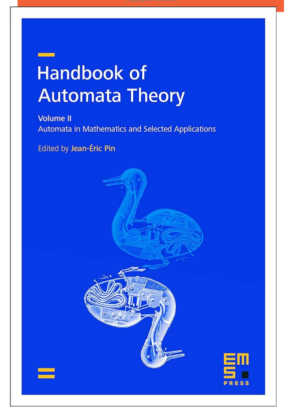


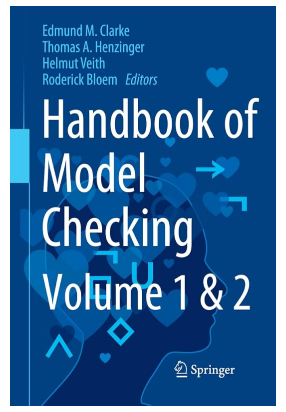




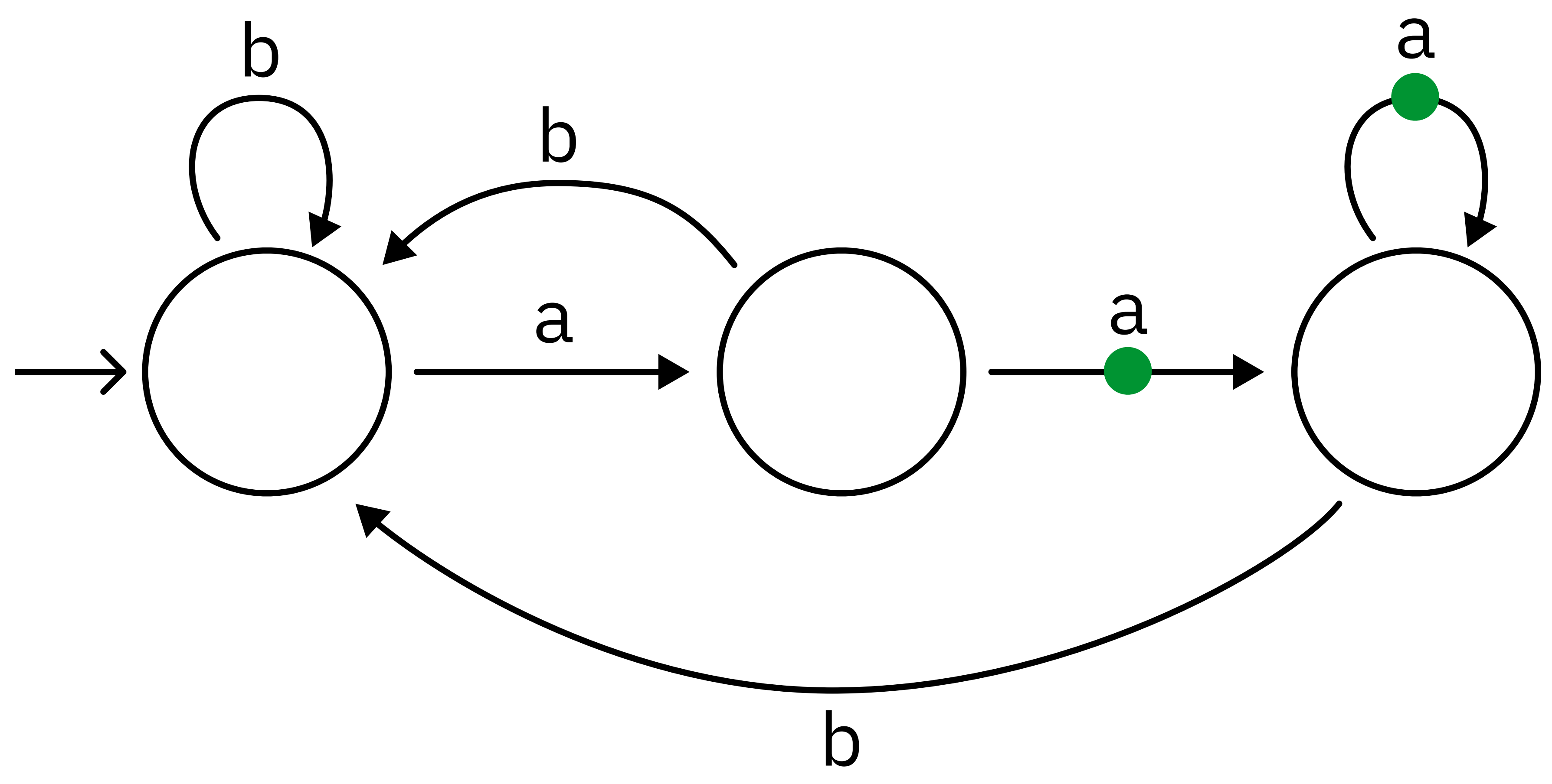
[Many, many, works]

2020





A transition-based ω-automaton

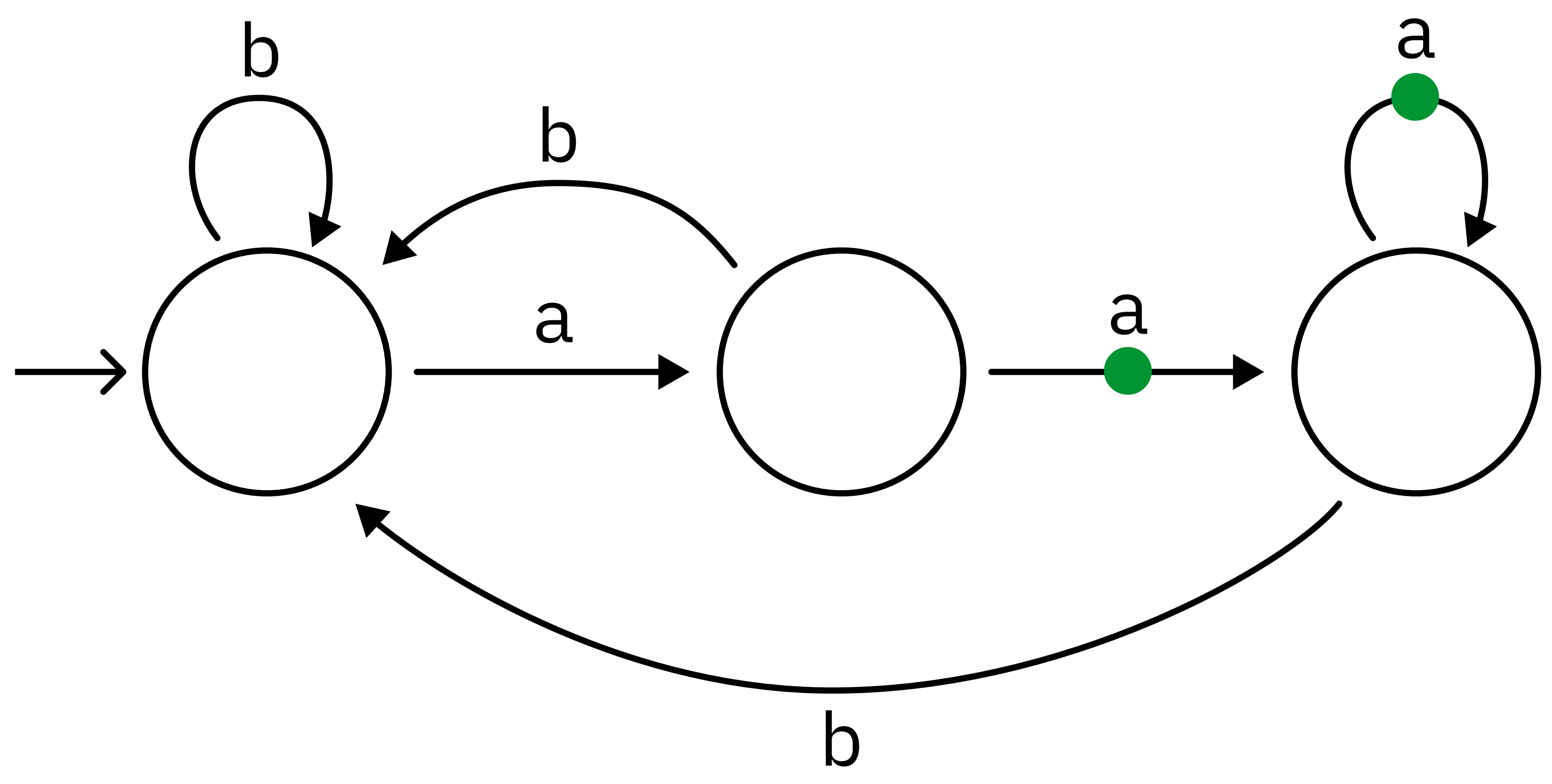


$$\mathcal{L}(\mathcal{A}) = \text{Words containing ‘}aa\text{’ infinitely often} \subseteq \Sigma^\omega$$

Büchi condition: We accept if ● visited infinitely often

Similar for Rabin, parity....

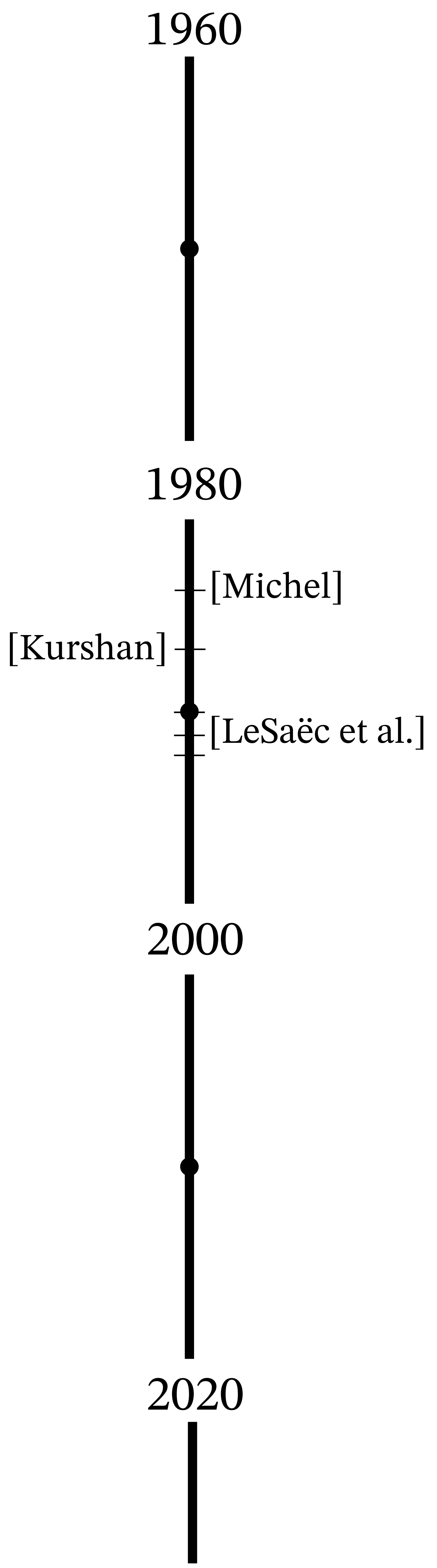
A transition-based ω-automaton



$$\mathcal{L}(\mathcal{A}) = \text{Words containing '}aa\text{' infinitely often} \subseteq \Sigma^\omega$$

Büchi condition: We accept if ● visited infinitely often

Similar for Rabin, parity....



Fact

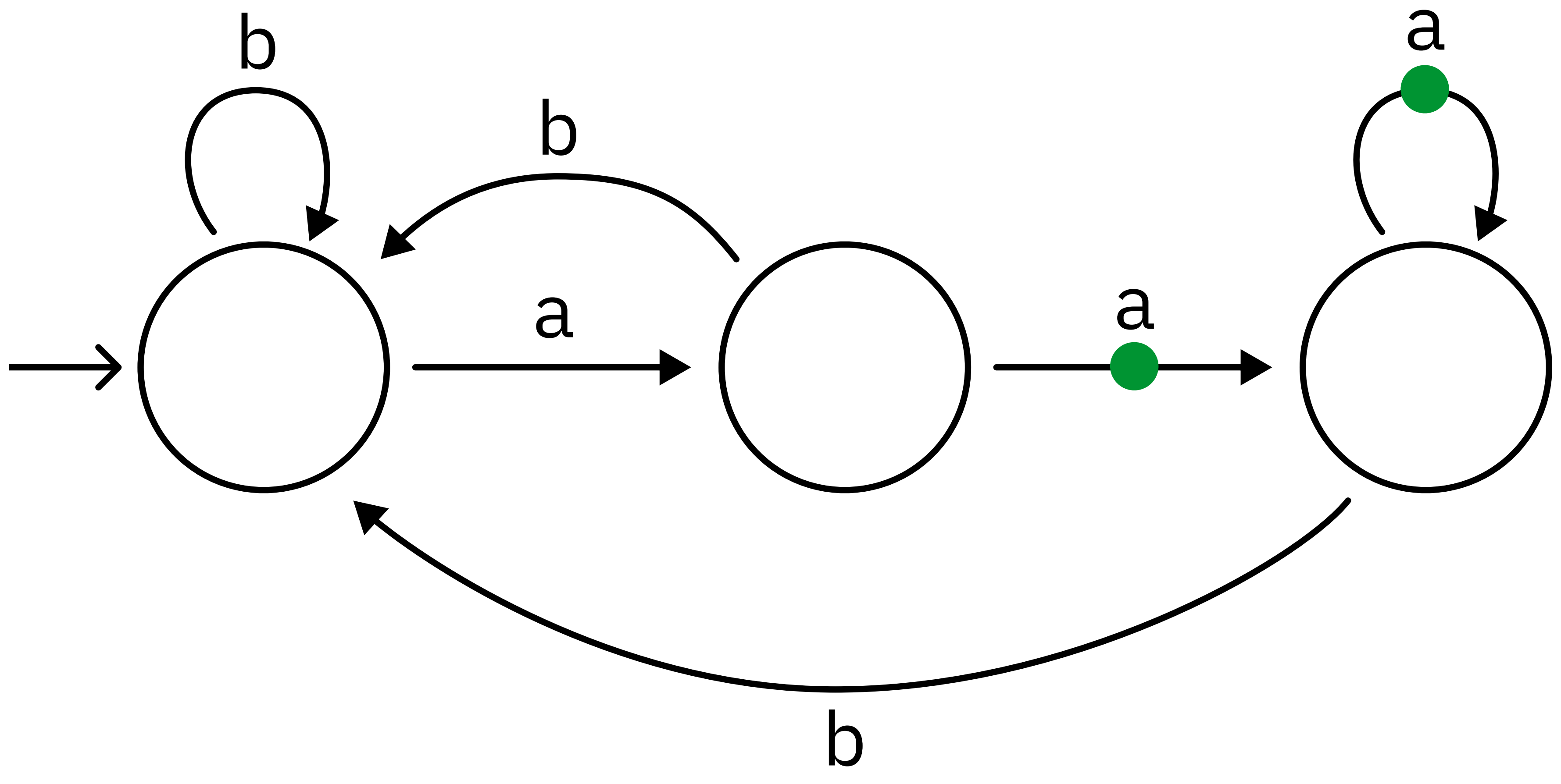
In general, there is no unique minimal deterministic ω-automata for a given ω-regular language.

(Van, Le Saëc, Litovsky '95)

Characterization of languages L that admit a unique minimal deterministic Muller automaton.

Transition-based!

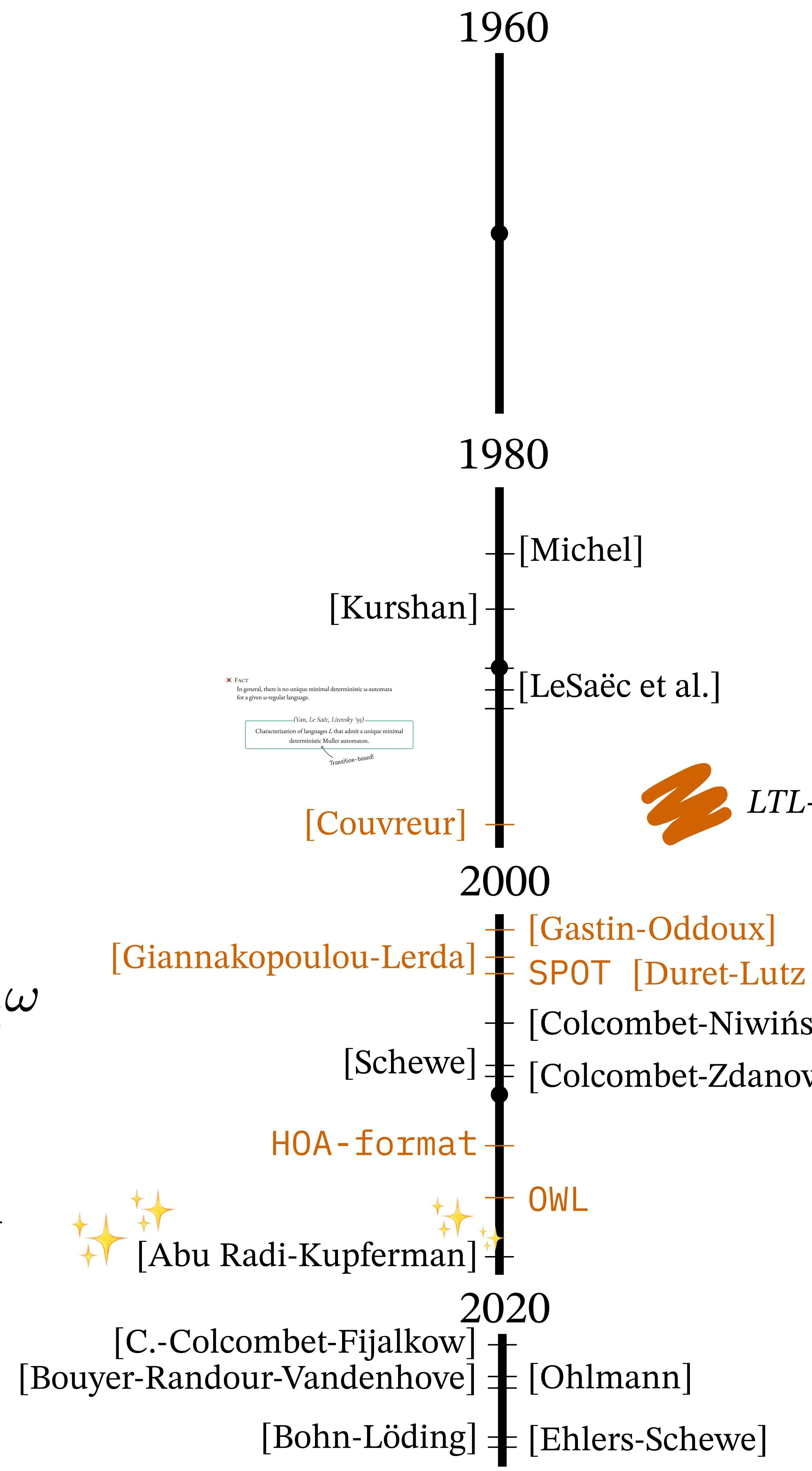
A transition-based ω-automaton

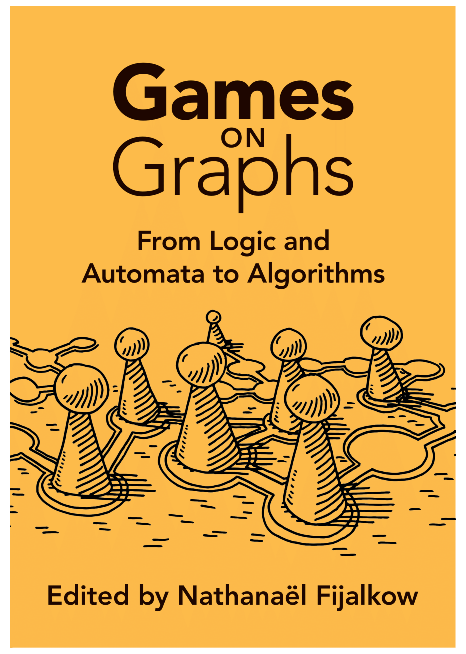


$\mathcal{L}(\mathcal{A}) =$ Words containing ‘*aa*’ infinitely often $\subseteq \Sigma^\omega$

Büchi condition: We accept if ● visited infinitely often

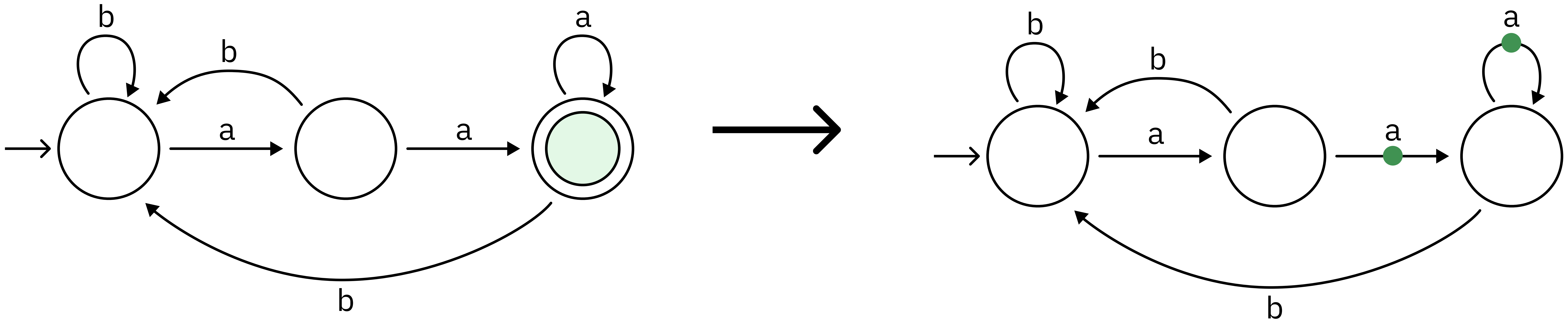
Similar for Rabin, parity....





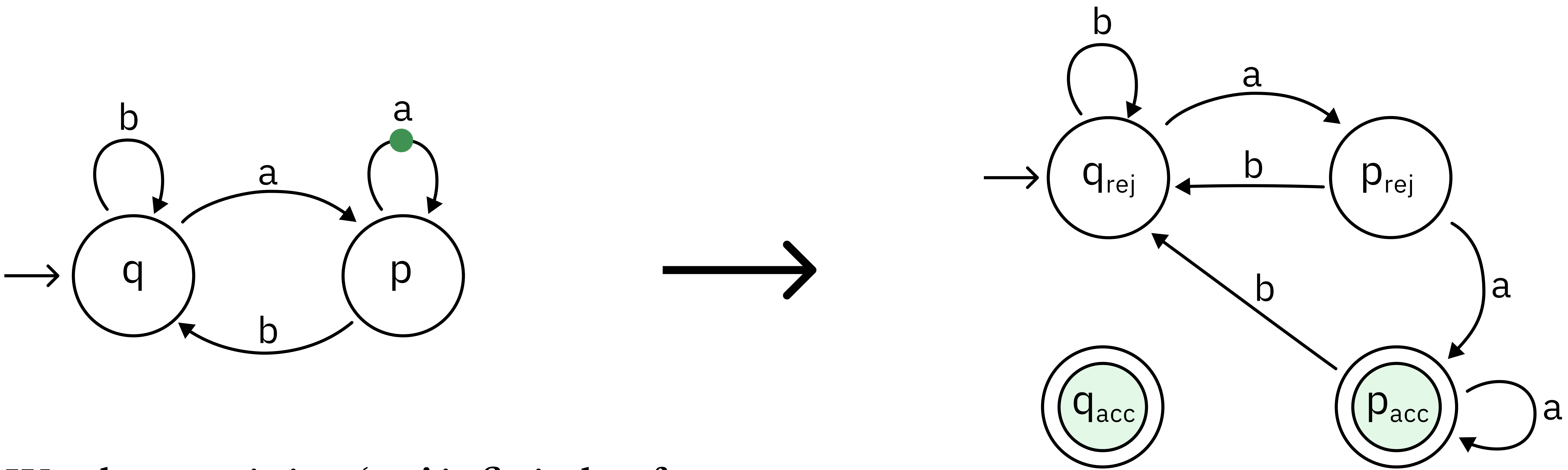
Both models are equivalent

From states to transitions



No extra states needed

From transitions to states

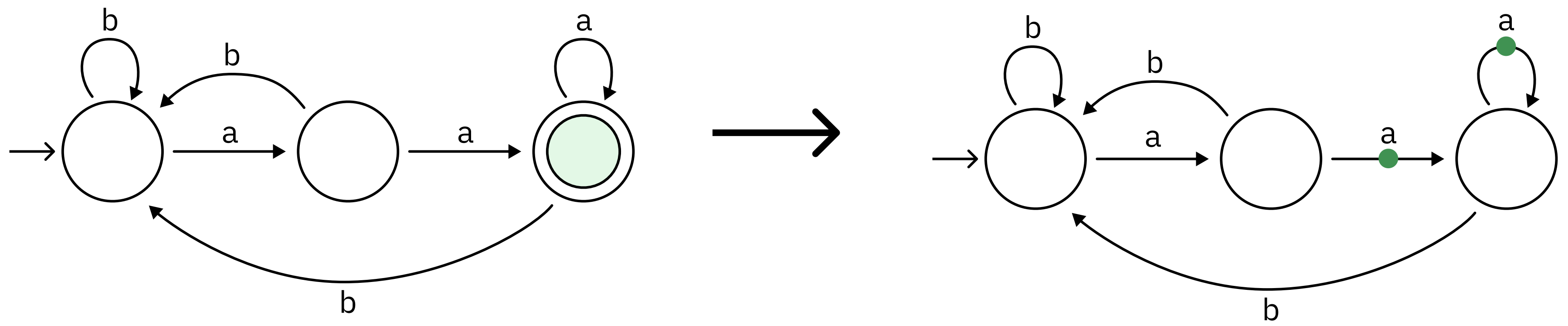


$\mathcal{L}(\mathcal{A}) =$ Words containing ‘*aa*’ infinitely often

We may need to double the number of states

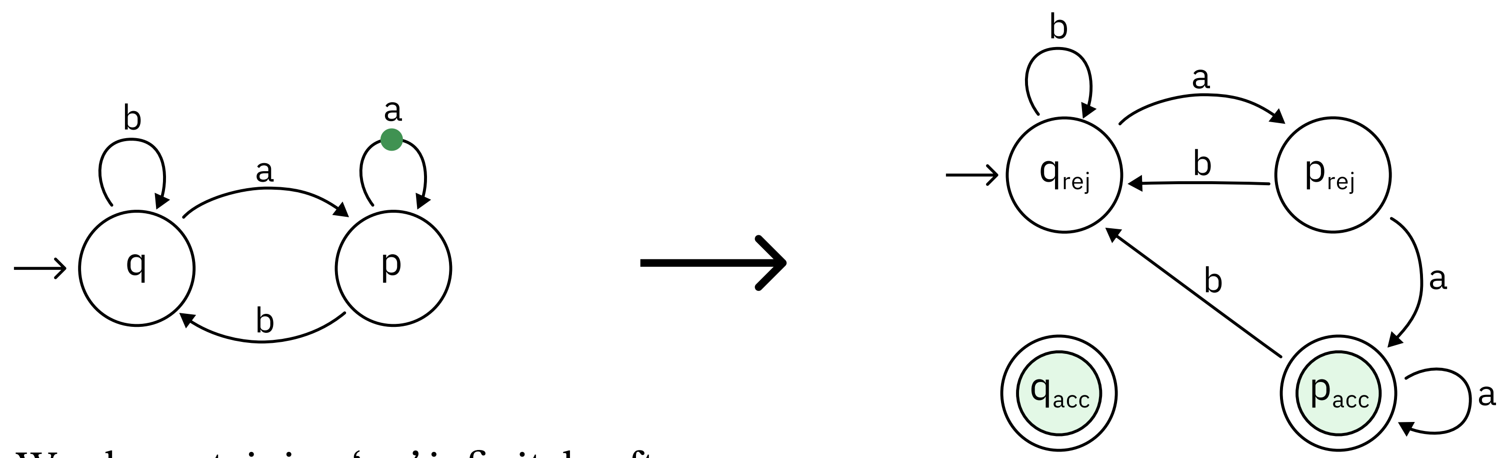
Both models are not so equivalent

From states to transitions



No extra states needed

From transitions to states



$\mathcal{L}(\mathcal{A}) =$ Words containing '*aa*' infinitely often

We may need to double the number of states

✦ Transition-based automata are smaller

From transitions to states optimally:

`Input:` Transition-based Büchi automaton

`Question:` What is the minimal number of states that we need to duplicate to obtain an equivalent state-based automaton?

✗ This problem is NP-complete! *(Schewe '09 - C. '23)*

SOURCE OF NON-CANONICITY

A landscape of problems

States

Transitions

THEOREM *(Schewe '09)*

Minimisation of deterministic state-based Büchi automata is NP-complete.

The reduction does not generalise

THEOREM *(Schewe '20)*

Minimisation of state-based history-deterministic coBüchi automata is NP-complete.

THEOREM *(Abu Radi-Kupferman '19)*

Minimisation of transitions-based history-deterministic coBüchi automata in PTIME.

States

Transitions

Highly technical

THEOREM *(Schewe '09)*

Minimisation of deterministic state-based Büchi automata is NP-complete.

THEOREM *(Abu Radi-Ehlers '25)*

Minimisation of deterministic transition-based Büchi automata is NP-complete.

THEOREM *(Schewe '20)*

Minimisation of state-based history-deterministic coBüchi automata is NP-complete.

THEOREM *(Abu Radi-Kupferman '19)*

Minimisation of transitions-based history-deterministic coBüchi automata in PTIME.

Minimisation

Determinisation

States

✗ THEOREM *(Schewe '09)*
Minimisation of deterministic state-based Büchi automata is NP-complete.

✗ THEOREM *(Schewe '20)*
Minimisation of state-based history-deterministic coBüchi automata is NP-complete.

Transitions

Highly technical

✗ THEOREM *(Abu Radi-Ehlers '25)*
Minimisation of deterministic transition-based Büchi automata is NP-complete.

✦ THEOREM *(Abu Radi-Kupferman '19)*
Minimisation of transitions-based history-deterministic coBüchi automata in PTIME.

A landscape of problems

ND Büchi Automaton → *[Safra '88]* → Deterministic Rabin Automaton

[Piterman '06], [Schewe '09] TRANSITION-BASED

Automaton of size n → Automaton of size $\det(n) \simeq (1.66n)^n$

✦ THEOREM *(tight bounds)* *(Colcombet-Zdanowski '09)*
There are ND Büchi automata of size n, such that a minimal equivalent transition-based deterministic Rabin automaton has size exactly $\det(n)$.

TIGHT UP TO 0 STATES!

✗ No such tight bounds for state-based automata

Minimisation

States

✗ THEOREM *(Schewe '09)*
Minimisation of deterministic state-based Büchi automata is NP-complete.

✗ THEOREM *(Schewe '20)*
Minimisation of state-based history-deterministic coBüchi automata is NP-complete.

Transitions

Highly technical

✗ THEOREM *(Abu Radi-Ehlers '25)*
Minimisation of deterministic transition-based Büchi automata is NP-complete.

✦ THEOREM *(Abu Radi-Kupferman '19)*
Minimisation of transitions-based history-deterministic coBüchi automata in PTIME.

Determinisation

ND Büchi Automaton → Deterministic Rabin Automaton
[Safra '88]
[Piterman '06], [Schewe '09] TRANSITION-BASED

Automaton of size n → Automaton of size $\det(n) \simeq (1.66n)^n$

✦ THEOREM *(tight bounds) (Colcombet-Zdanowski '09)*
There are ND Büchi automata of size n, such that a minimal equivalent transition-based deterministic Rabin automaton has size exactly $\det(n)$.

TIGHT UP TO 0 STATES!

✗ No such tight bounds for state-based automata

A landscape of problems

Automata Transformations

Generalised-Büchi Automaton

$\{ \bullet_1, \bullet_2, \dots, \bullet_k \}$

See all colours inf. often

⟶

Büchi Automaton (Extra states)

See ● inf. often

Generalised-Büchi Automaton

$\{ \bullet_1, \bullet_2, \dots, \bullet_k \}$

See all colours inf. often

(always possible) ⟶

Büchi Automaton (Extra states)

See $\bullet$ inf. often

Input: Generalised-Büchi automaton

Question: What is the minimal number of states that we need to duplicate to define an equivalent Büchi automaton?

✦ THEOREM *(C.-Colcombet-Fijalkow '21)*

PTIME for transition-based automata.

✗ THEOREM *(C. '23)*

NP-complete for state-based automata.

Minimisation

States

✗ THEOREM *(Schewe '09)*
Minimisation of deterministic state-based Büchi automata is NP-complete.

✗ THEOREM *(Schewe '20)*
Minimisation of state-based history-deterministic coBüchi automata is NP-complete.

Transitions

Highly technical

✗ THEOREM *(Abu Radi-Ehlers '25)*
Minimisation of deterministic transition-based Büchi automata is NP-complete.

✦ THEOREM *(Abu Radi-Kupferman '19)*
Minimisation of transitions-based history-deterministic coBüchi automata in PTIME.

Determinisation

ND Büchi Automaton → Deterministic Rabin Automaton *[Safra '88]*, *[Piterman '06], [Schewe '09]* TRANSITION-BASED

Automaton of size n → Automaton of size det(n) ≃ $(1.66n)^n$

✦ THEOREM *(tight bounds)* *(Colcombet-Zdanowski '09)*
There are ND Büchi automata of size n, such that a minimal equivalent transition-based deterministic Rabin automaton has size exactly det(n).

TIGHT UP TO 0 STATES!

✗ No such tight bounds for state-based automata

A landscape of problems

Automata Transformations

Generalised-Büchi Automaton $\{●_1, ●_2, \dots, ●_k\}$ See all colours inf. often → *(always possible)* Büchi Automaton See ● inf. often (Extra states)

Input: Generalised-Büchi automaton

Question: What is the minimal number of states that we need to duplicate to define an equivalent Büchi automaton?

✦ THEOREM *(C.-Colcombet-Fijalkow '21)*
PTIME for transition-based automata.

✗ THEOREM *(C. '23)*
NP-complete for state-based automata.

Positionality in games

✦ THEOREM *(Mostowski '84, Emerson-Jutla '91)*

Games using a parity language as winning condition are positionally determined.

$$\text{parity}_{[0,d]} = \{w \in [0,d]^\omega \mid \limsup w \text{ is even}\}$$

✦ THEOREM *(Colcombet-Niwiński '06)*

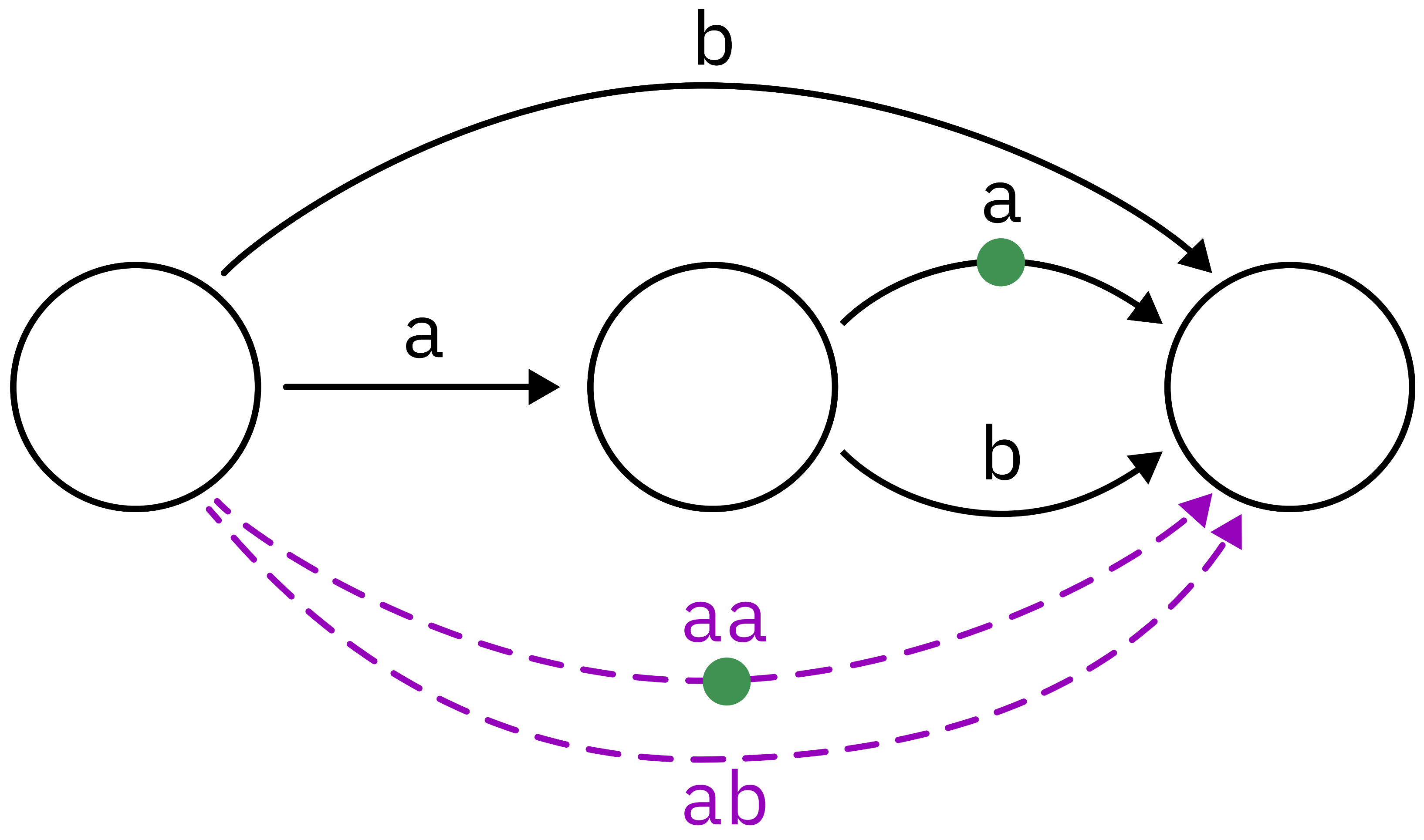
The only prefix-independent languages L such that all games with condition L are positionally determined are parity languages.

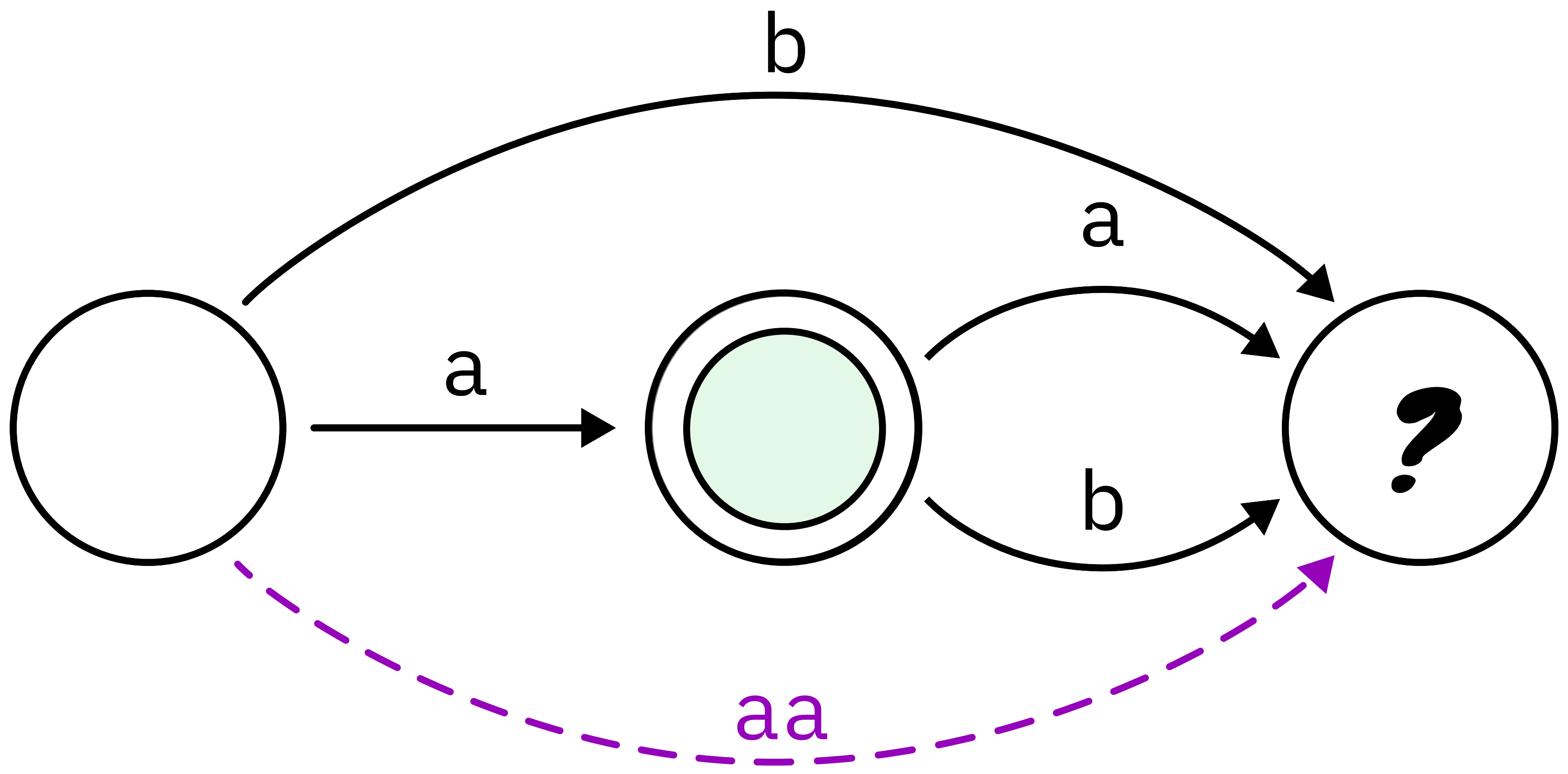
Over transition-based games!

✘ No characterisation for state-coloured games.

Why all these differences?

A natural algebraic operation





- Compositionality
- No natural way of composing transitions
- Connections with algebra *[Wilke '91, LeSaëc-Pin-Weil '91, Colcombet '11]*
- Saturation techniques *[Colcombet-Fijalkow '18, Ohlmann '23, C.-Ohlmann '25]*

Conclusion

Transition-based models are better fitted for both theoretical and practical purposes.

Recent (transition-based) canonical models

All ω-regular languages

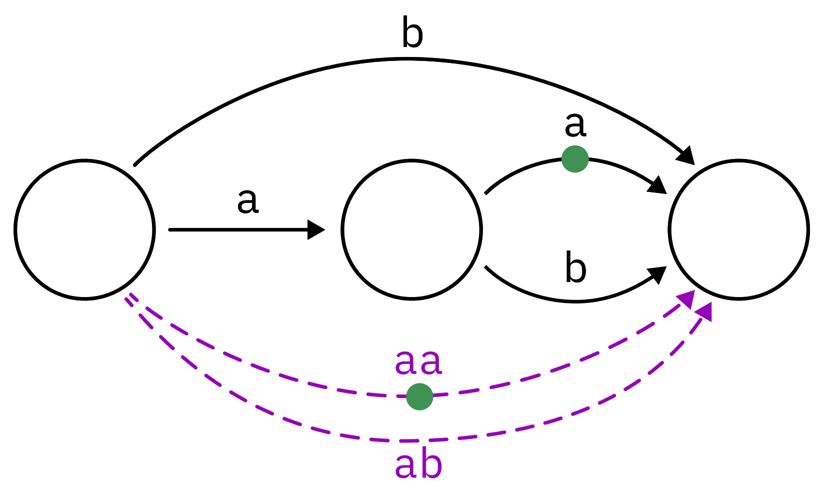
- HD-coBüchi automata *(Abu Radi-Kupferman '19)*

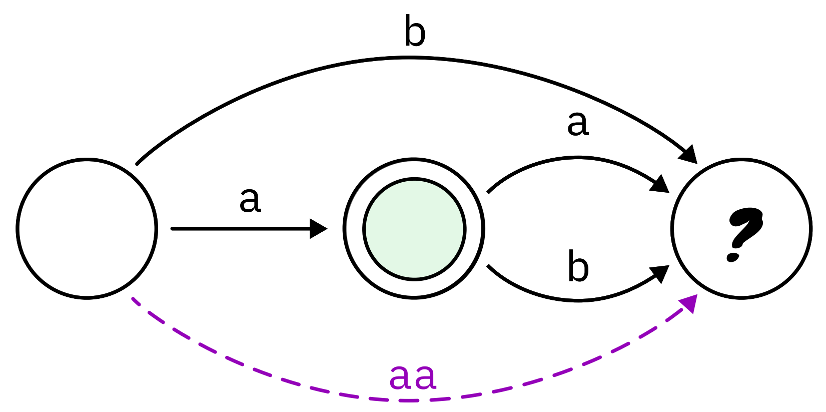
A subclass of ω-regular languages

- Chains of HD-coBüchi automata *(Ehlers-Schewe '22)*
- Rerailing automata *(Ehlers '25)*
- Layered automata *(C.-Löding-Walukiewicz. Soon on arxiv!)*
  - Canonicity expressed in terms of morphisms
  - Congruence-based characterization

Why all these differences?

A natural algebraic operation





✦ Compositionality

✗ No natural way of composing transitions

✦ Connections with algebra *[Wilke '91, LeSaëc-Pin-Weil'91, Colcombet '11]*

✦ Saturation techniques *[Colcombet-Fijalkow '18, Ohlmann'23, C.-Ohlmann'25]*

Conclusion

Transition-based models are better fitted for both theoretical and practical purposes.

Recent (transition-based) canonical models

All ω-regular languages

✦ HD-coBüchi automata *(Abu Radi-Kupferman '19)*

A subclass of ω-regular languages

✦ Chains of HD-coBüchi automata *(Ehlers-Schewe '22)*

✦ Rerailing automata *(Ehlers '25)*

✦ Layered automata *(C.-Löding-Walukiewicz. Soon on arxiv!)*

- Canonicity expressed in terms of morphisms
- Congruence-based characterization

Thanks for your attention!