# Unambiguity in automata theory

Thomas Colcombet
DCFS 2015

Waterloo, June 26, 2015

# Unambiguity

A non-deterministic device is **unambiguous** if for all inputs there is at most one accepting execution.

# Unambiguity

> A non-deterministic device is **unambiguous** if for all inputs there is at most one accepting execution.

It is a semantic notion. It is not clear what is a witness of unambiguity…

# Unambiguity

A non-deterministic device is **unambiguous** if for all inputs there is at most one accepting execution.

It is a semantic notion. It is not clear what is a witness of unambiguity…

**Important remark:** As opposed to deterministic devices, unambiguous ones are not naturally closed under complement.

# Unambiguity

A non-deterministic device is **unambiguous** if for all inputs there is at most one accepting execution.

It is a semantic notion. It is not clear what is a witness of unambiguity…

**Important remark:** As opposed to deterministic devices, unambiguous ones are not naturally closed under complement.

[Jurdzinski 98]
ParityGames in UP∩CoUP

[Bourke&Tewari&Vinodchandran 07]
Planar reachability in UL (unambiguous logspace)

[Allender&Reinhardt 97]
UL and NL coincide in the non-uniform setting (open in the uniform one)

# The many forms of automata

Word automata

Transducers

Infinite word automata

Infinite tree automata

Tropical automata

Register automata

# The many forms of automata

Word automata

Transducers *

Infinite word automata

Infinite tree automata

Tropical automata

Register automata

} Non-deterministic and unambiguous automata have the same expressive power.

# The many forms of automata

- Word automata
- Transducers *
- Infinite word automata

Non-deterministic and unambiguous automata have the same expressive power.

- Infinite tree automata
- Tropical automata
- Register automata

Non-deterministic and unambiguous automata have different expressive power.

# The many forms of automata

- Word automata
- Transducers *
- Infinite word automata

Non-deterministic and unambiguous automata have the same expressive power.

- Infinite tree automata
- Tropical automata
- Register automata

Non-deterministic and unambiguous automata have different expressive power.

# The many forms of automata

- → Word automata
- Transducers *
- Infinite word automata

} Non-deterministic and unambiguous automata have the same expressive power.

- Infinite tree automata
- → Tropical automata
- Register automata

} Non-deterministic and unambiguous automata have different expressive power.

**Remark:** in general it is easy to decide if an automaton is unambiguous.
Proof: *Take the product of the automaton with itself + 1 bit, such that it accepts an input iff there exist two distinct runs of the original automaton. Test for emptiness.*

# Unambiguous finite word automata

# Unambiguous word automata

Remark 1: every word automaton can be made deterministic, and hence unambiguous.

# Unambiguous word automata

**Remark 1:** every word automaton can be made deterministic, and hence unambiguous.

**Theorem** [Leiss 81, Leung 98&05]**:** Unambiguous automata can be exponentially more succinct than deterministic automata.
Non-deterministic automata can be exponentially more succinct than unambiguous automata (even polynomially ambiguous).

The n'th letter from the end is an 'a'.

$$(0 + (01^*)^{n-1}0)^*$$

# Unambiguous word automata

**Remark 1:** every word automaton can be made deterministic, and hence unambiguous.

**Theorem** [Leiss 81, Leung 98&05]**:** Unambiguous automata can be exponentially more succinct than deterministic automata.
Non-deterministic automata can be exponentially more succinct than unambiguous automata (even polynomially ambiguous).

The n'th letter from the end is an 'a'.

$$(0 + (01^*)^{n-1}0)^*$$

**Fact:** Universality of non-deterministic word automata is PSPACE.

# Unambiguous word automata

**Remark 1:** every word automaton can be made deterministic, and hence unambiguous.

**Theorem** [Leiss 81, Leung 98&05]**:** Unambiguous automata can be exponentially more succinct than deterministic automata.
Non-deterministic automata can be exponentially more succinct than unambiguous automata (even polynomially ambiguous).

The n'th letter from the end is an 'a'.

$$(0 + (01^*)^{n-1}0)^*$$

**Fact:** Universality of non-deterministic word automata is PSPACE.

**Theorem** [Hunt&Stearns 81]**:** Universality of unambiguous automata is in P.

# Communication complexity

[Yao79]: given a problem where the inputs are split into two parts A and B, what quantity of information should be communicated between an A-process and a B-process for agreeing on the output ?

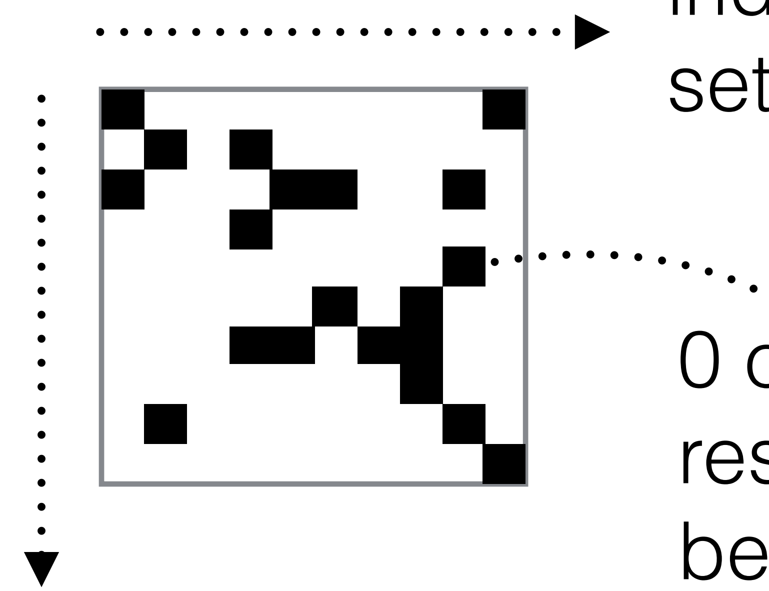
# Communication complexity

[Yao79]: given a problem where the inputs are split into two parts A and B, what quantity of information should be communicated between an A-process and a B-process for agreeing on the output ?

One sees the problem as the (input) matrix:

The communication complexity depends on the shape of the matrix.

indexed by the set of A-inputs



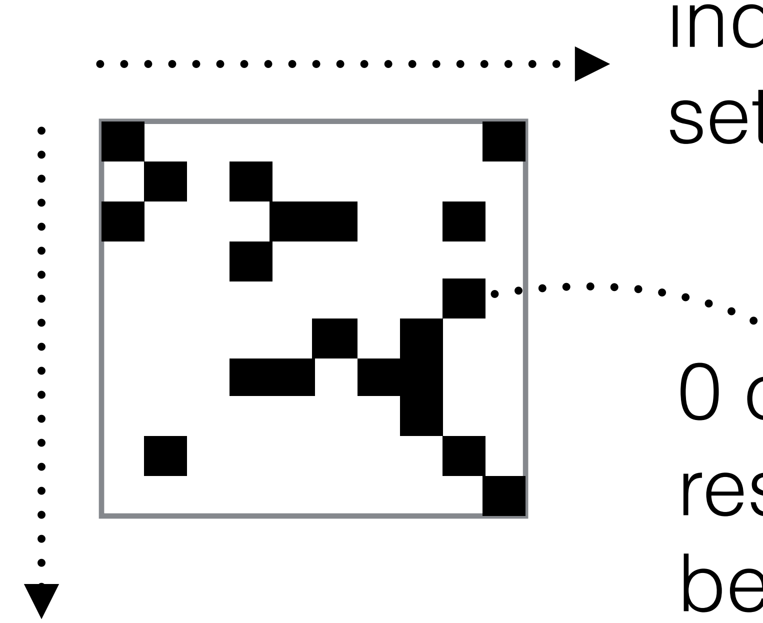
0 or 1 if the resulting input belongs to the problem.

indexed by the set of B-inputs

# Communication complexity

[Yao79]: given a problem where the inputs are split into two parts A and B, what quantity of information should be communicated between an A-process and a B-process for agreeing on the output ?

One sees the problem as the (input) matrix:



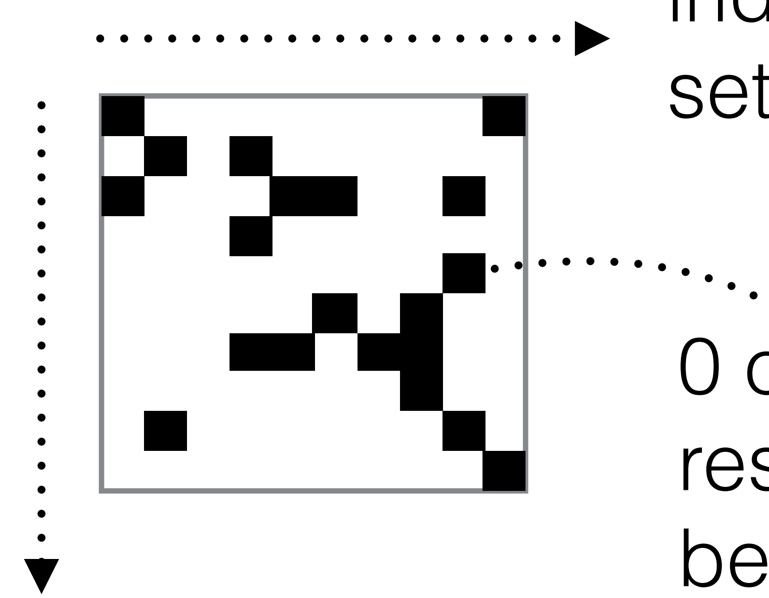

The communication complexity depends on the shape of the matrix.

**Example:** If both A and B get n bits, determining whether the two sequences are equal requires n bits of communication.

# Communication complexity

[Yao79]: given a problem where the inputs are split into two parts A and B, what quantity of information should be communicated between an A-process and a B-process for agreeing on the output ?

One sees the problem as the (input) matrix:




The communication complexity depends on the shape of the matrix.

**Example:** If both A and B get n bits, determining whether the two sequences are equal requires n bits of communication.

Usually the story continues with randomized protocols (processes can flip coins)… (yields O(log(n)) bits in the above example).

# CC for automata

# CC for automata

Given a language L, one can construct the language matrix indexed by words:

$$M_L(u, v) = \begin{cases} 1 & \text{if } uv \in L \\ 0 & \text{otherwise} \end{cases}$$

# CC for automata

Given a language L, one can construct the language matrix indexed by words:

$$M_L(u,v) = \begin{cases} 1 & \text{if } uv \in L \\ 0 & \text{otherwise} \end{cases}$$

Given a non-deterministic automaton with states Q, this matrix has value 1 at entry (u,v) if and only if there is a state q∈Q such that

$I \xrightarrow{u} q$ and $q \xrightarrow{v} F$

# CC for automata

Given a language L, one can construct the language matrix indexed by words:

$$M_L(u,v) = \begin{cases} 1 & \text{if } uv \in L \\ 0 & \text{otherwise} \end{cases}$$

Given a non-deterministic automaton with states Q, this matrix has value 1 at entry (u,v) if and only if there is a state q∈Q such that

$$I \xrightarrow{u} q \qquad \text{and} \qquad q \xrightarrow{v} F$$

Hence the matrix is obtained as the superposition of |Q|-many (combinatorial) rectangles of 1's.

# CC for automata

Given a language L, one can construct the language matrix indexed by words:

$$M_L(u,v) = \begin{cases} 1 & \text{if } uv \in L \\ 0 & \text{otherwise} \end{cases}$$

Given a non-deterministic automaton with states Q, this matrix has value 1 at entry (u,v) if and only if there is a state q∈Q such that

$$I \xrightarrow{u} q \quad \text{and} \quad q \xrightarrow{v} F$$

Hence the matrix is obtained as the superposition of |Q|-many (combinatorial) rectangles of 1's.

The non-deterministic complexity of a matrix is the least number of 1-rectangles that defines it.

# CC for automata

Given a language L, one can construct the language matrix indexed by words:

$$M_L(u,v) = \begin{cases} 1 & \text{if } uv \in L \\ 0 & \text{otherwise} \end{cases}$$

Given a non-deterministic automaton with states Q, this matrix has value 1 at entry (u,v) if and only if there is a state q∈Q such that

$$I \xrightarrow{u} q \qquad \text{and} \qquad q \xrightarrow{v} F$$

Hence the matrix is obtained as the superposition of |Q|-many (combinatorial) rectangles of 1's.

The non-deterministic complexity of a matrix is the least number of 1-rectangles that defines it.

This is a lower bound on the number of states of a non-deterministic automaton accepting L.

# CC for automata

Given a language L, one can construct the language matrix indexed by words:

$$M_L(u,v) = \begin{cases} 1 & \text{if } uv \in L \\ 0 & \text{otherwise} \end{cases}$$

Given a non-deterministic automaton with states Q, this matrix has value 1 at entry (u,v) if and only if there is a state q∈Q such that

$$I \xrightarrow{u} q \qquad \text{and} \qquad q \xrightarrow{v} F$$

Hence the matrix is obtained as the superposition of |Q|-many (combinatorial) rectangles of 1's.

The non-deterministic complexity of a matrix is the least number of 1-rectangles that defines it.

This is a lower bound on the number of states of a non-deterministic automaton accepting L.

(Extended) fooling set technique [Birget93, Glaister&Shallit97]:
If $u_1 v_1 \in L, \ldots, u_n v_n \in L$
and $u_i v_j \notin L$ or $u_j v_i \notin L$ for all $i \neq j$
then a non-deterministic automaton for L has at least n states.

# CC for automata

Given a language L, one can construct the language matrix indexed by words:

$$M_L(u,v) = \begin{cases} 1 & \text{if } uv \in L \\ 0 & \text{otherwise} \end{cases}$$

Given a non-deterministic automaton with states Q, this matrix has value 1 at entry (u,v) if and only if there is a state q∊Q such that

$I \xrightarrow{u} q$ and $q \xrightarrow{v} F$

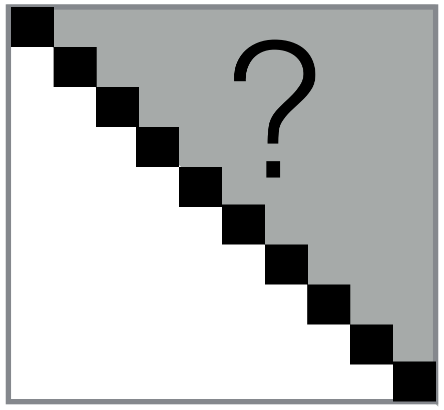
Hence the matrix is obtained as the superposition of |Q|-many (combinatorial) rectangles of 1's.

The non-deterministic complexity of a matrix is the least number of 1-rectangles that defines it.

This is a lower bound on the number of states of a non-deterministic automaton accepting L.

(Extended) fooling set technique [Birget93, Glaister&Shallit97]:
If $u_1v_1 \in L, \ldots, u_nv_n \in L$
and $u_iv_j \notin L$ or $u_jv_i \notin L$ for all $i \neq j$
then a non-deterministic automaton for L has at least n states.

Proof:



# CC for unambiguous automata

Given an non-deterministic automaton with states Q, the language matrix has value 1 at entry (u,v) if and only if there is a state q∈Q such that

$$I \xrightarrow{u} q \qquad \text{and} \qquad q \xrightarrow{v} F$$

# CC for unambiguous automata

Given an non-deterministic automaton with states Q, the language matrix has value 1 at entry (u,v) if and only if there is a state q∈Q such that

$$I \stackrel{u}{\to} q \qquad \text{and} \qquad q \stackrel{v}{\to} F$$

If the automaton is unambiguous, there can be at most one such state.

# CC for unambiguous automata

Given an non-deterministic automaton with states Q, the language matrix has value 1 at entry (u,v) if and only if there is a state q∈Q such that

$$I \xrightarrow{u} q \qquad \text{and} \qquad q \xrightarrow{v} F$$

If the automaton is unambiguous, there can be at most one such state.

Hence the language matrix is obtained as the disjoint superposition of |Q|-many (combinatorial) rectangles of 1's.

# CC for unambiguous automata

Given an non-deterministic automaton with states Q, the language matrix has value 1 at entry (u,v) if and only if there is a state q∈Q such that

$$I \xrightarrow{u} q \qquad \text{and} \qquad q \xrightarrow{v} F$$

If the automaton is unambiguous, there can be at most one such state.

Hence the language matrix is obtained as the disjoint superposition of |Q|-many (combinatorial) rectangles of 1's.

**Def:** The unambiguous complexity of a matrix is the minimal number of 1-rectangles, the disjoint union of which yields the matrix.

# CC for unambiguous automata

Given an non-deterministic automaton with states Q, the language matrix has value 1 at entry (u,v) if and only if there is a state $q \in Q$ such that

$$I \xrightarrow{u} q \qquad \text{and} \qquad q \xrightarrow{v} F$$

If the automaton is unambiguous, there can be at most one such state.

Hence the language matrix is obtained as the disjoint superposition of |Q|-many (combinatorial) rectangles of 1's.

**Def:** The unambiguous complexity of a matrix is the minimal number of 1-rectangles, the disjoint union of which yields the matrix.

The unambiguous complexity for a language matrix is a *lower bound* on the number of states for an unambiguous automaton.

# CC for unambiguous automata

Given an non-deterministic automaton with states Q, the language matrix has value 1 at entry (u,v) if and only if there is a state q∈Q such that

$$I \xrightarrow{u} q \qquad \text{and} \qquad q \xrightarrow{v} F$$

If the automaton is unambiguous, there can be at most one such state.

Hence the language matrix is obtained as the disjoint superposition of |Q|-many (combinatorial) rectangles of 1's.

**Remark:** disjoint superposition is the same as the sum.

**Def:** The unambiguous complexity of a matrix is the minimal number of 1-rectangles, the disjoint union of which yields the matrix.

The unambiguous complexity for a language matrix is a *lower bound* on the number of states for an unambiguous automaton.

# CC for unambiguous automata

Given an non-deterministic automaton with states Q, the language matrix has value 1 at entry (u,v) if and only if there is a state q∈Q such that

$$I \xrightarrow{u} q \quad \text{and} \quad q \xrightarrow{v} F$$

If the automaton is unambiguous, there can be at most one such state.

Hence the language matrix is obtained as the disjoint superposition of |Q|-many (combinatorial) rectangles of 1's.

**Def:** The unambiguous complexity of a matrix is the minimal number of 1-rectangles, the disjoint union of which yields the matrix.

The unambiguous complexity for a language matrix is a *lower bound* on the number of states for an unambiguous automaton.

**Remark:** disjoint superposition is the same as the sum.

Since rank(rectangle)=1 and rank is subadditive,

rank(M)≤unamb-comp(M)

# Application: the linear length of witnesses for non-universality

# Application: the linear length of witnesses for non-universality

**Fact:** for a non-universal non-deterministic automaton, the minimal size of a witness of non-universality can be of exponential length.

# Application: the linear length of witnesses for non-universality

**Fact:** for a non-universal non-deterministic automaton, the minimal size of a witness of non-universality can be of exponential length.

**Theorem** [Schmidt77]**:** the least size of a witness of non-universality for a language accepted by an unambiguous automaton of size n is n.

# Application: the linear length of witnesses for non-universality

**Fact:** for a non-universal non-deterministic automaton, the minimal size of a witness of non-universality can be of exponential length.

**Theorem** [Schmidt77]**:** the least size of a witness of non-universality for a language accepted by an unambiguous automaton of size n is n.

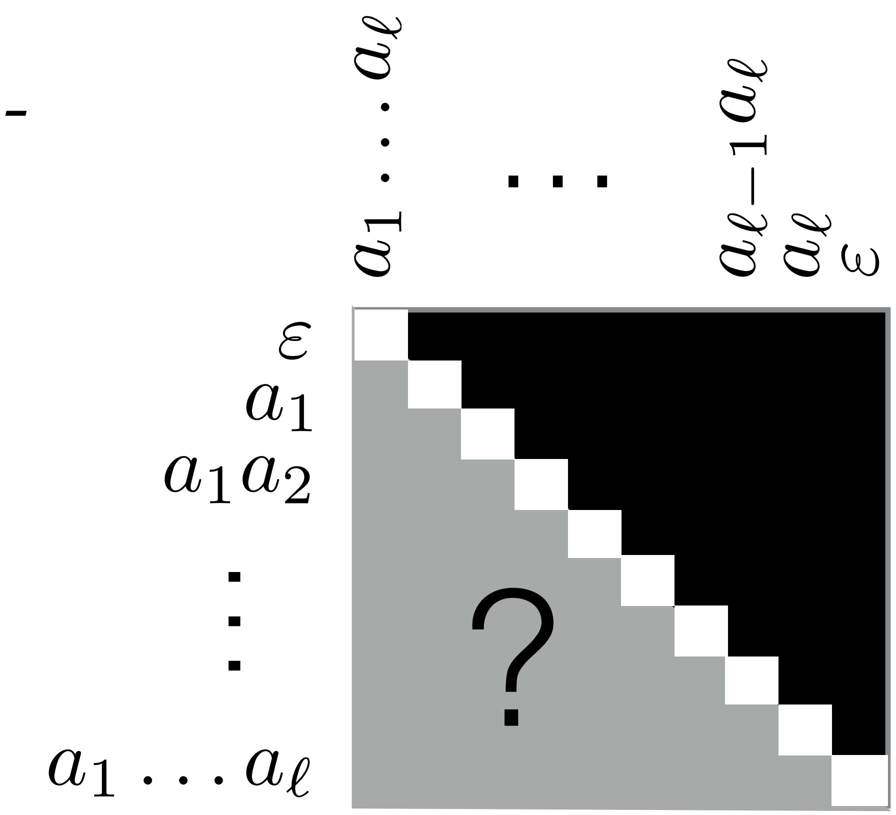
Proof: *consider a least size witness of non-universality* $a_1 \ldots a_\ell$ *and the following submatrix of the language matrix:*

# Application: the linear length of witnesses for non-universality

**Fact:** for a non-universal non-deterministic automaton, the minimal size of a witness of non-universality can be of exponential length.

**Theorem** [Schmidt77]**:** the least size of a witness of non-universality for a language accepted by an unambiguous automaton of size n is n.

Proof: *consider a least size witness of non-universality* $a_1 \dots a_\ell$ *and the following submatrix of the language matrix:*



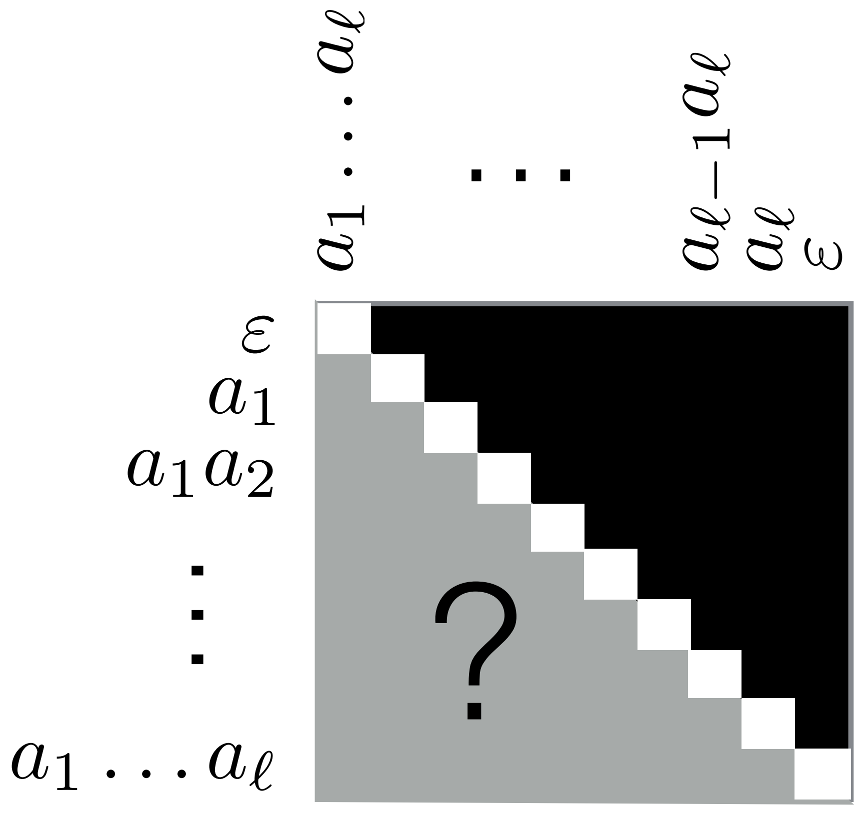
# Application: the linear length of witnesses for non-universality

**Fact:** for a non-universal non-deterministic automaton, the minimal size of a witness of non-universality can be of exponential length.

**Theorem** [Schmidt77]**:** the least size of a witness of non-universality for a language accepted by an unambiguous automaton of size n is n.

Proof: *consider a least size witness of non-universality* $a_1 \dots a_\ell$ *and the following submatrix of the language matrix:*

*Its rank is* $\geq \ell$*.*

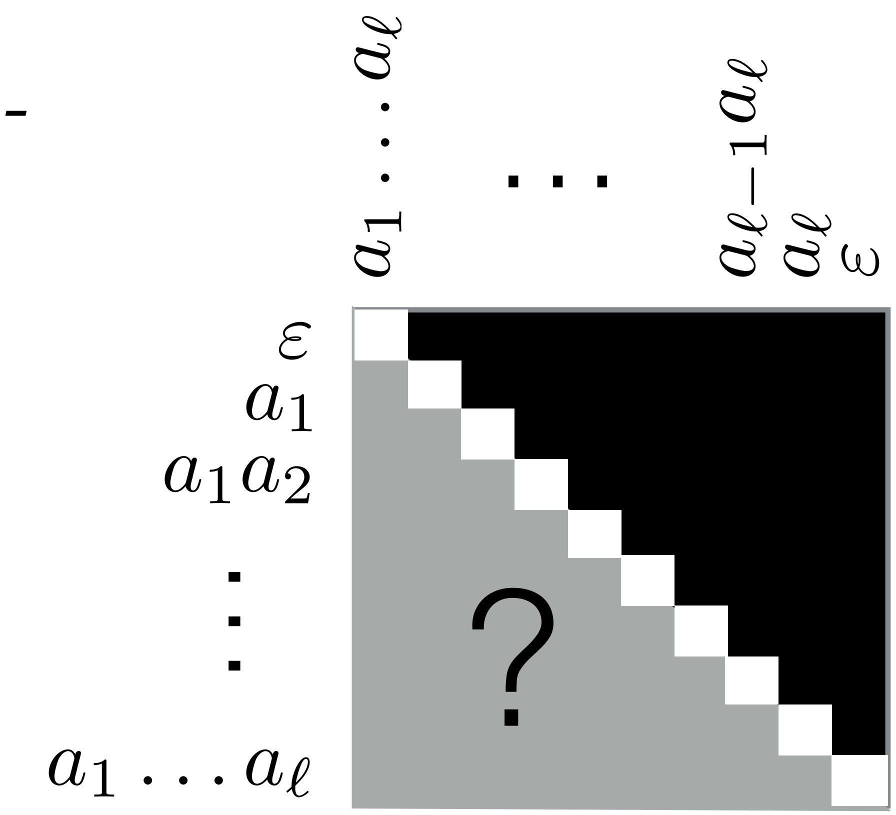


# Application: the linear length of witnesses for non-universality

**Fact:** for a non-universal non-deterministic automaton, the minimal size of a witness of non-universality can be of exponential length.

**Theorem** [Schmidt77]**:** the least size of a witness of non-universality for a language accepted by an unambiguous automaton of size n is n.

Proof: *consider a least size witness of non-universality* $a_1 \dots a_\ell$ *and the following submatrix of the language matrix:*




*Its rank is* $\geq \ell$.

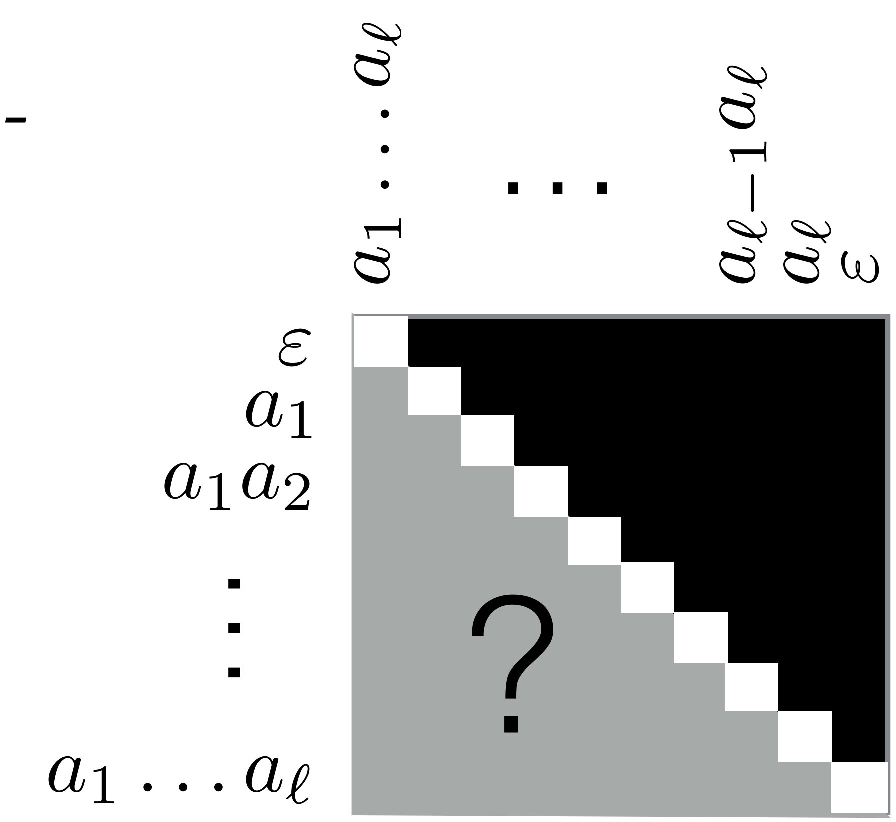
*Thus* $\ell \leq \mathrm{rank}(M_L) \leq n$.

# Application: the linear length of witnesses for non-universality

**Fact:** for a non-universal non-deterministic automaton, the minimal size of a witness of non-universality can be of exponential length.

**Theorem** [Schmidt77]**:** the least size of a witness of non-universality for a language accepted by an unambiguous automaton of size n is n.

Proof: *consider a least size witness of non-universality* $a_1 \ldots a_\ell$ *and the following submatrix of the language matrix:*




*Its rank is* $\geq \ell$.

*Thus* $\ell \leq \mathrm{rank}(M_L) \leq n$ .

**Corrolary:** The universality of unambiguous automata is in CoNP.

# Universality is in P

Theorem [Hunt&Stearns81]:

The universality of unambiguous automata is in P.

# Universality is in P

Theorem [Hunt&Stearns81]:
The universality of unambiguous automata is in P.

*Note that an unambiguous automaton of size n is universal if and only if it accepts* $1 + |A| + |A|^2 + \cdots + |A|^n$ *words of size at most n.*

# Universality is in P

Theorem [Hunt&Stearns81]:
The universality of unambiguous automata is in P.

*Note that an unambiguous automaton of size n is universal if and only if it accepts* $1 + |A| + |A|^2 + \cdots + |A|^n$ *words of size at most n.*

*Consider the Q×Q matrix*

$A_m(p, q)$ *= the number of words of length m accepted by a run from p to q.*

# Universality is in P

Theorem [Hunt&Stearns81]:
The universality of unambiguous automata is in P.

*Note that an unambiguous automaton of size n is universal if and only if it accepts* $1 + |A| + |A|^2 + \cdots + |A|^n$ *words of size at most n.*

*Consider the Q×Q matrix*

$A_m(p, q)$ *= the number of words of length m accepted by a run from p to q.*

*Using unambiguity (and trimmed assumption), we have:*

$A_m(p, q)$ *= the number of runs from p to q of length m.*

# Universality is in P

**Theorem** [Hunt&Stearns81]:
The universality of unambiguous automata is in P.

*Note that an unambiguous automaton of size n is universal if and only if it accepts* $1 + |A| + |A|^2 + \cdots + |A|^n$ *words of size at most n.*

*Consider the Q×Q matrix*

$A_m(p,q)$ *= the number of words of length m accepted by a run from p to q.*

*Using unambiguity (and trimmed assumption), we have:*

$A_m(p,q)$ *= the number of runs from p to q of length m.*

*Let* $M(p,q)$ *= the number of letters triggering a transition from p to q.*

*We have* $A_{m+1} = A_m M$ *.*

# Universality is in P

Theorem [Hunt&Stearns81]:
The universality of unambiguous automata is in P.

*Note that an unambiguous automaton of size n is universal if and only if it accepts* $1 + |A| + |A|^2 + \cdots + |A|^n$ *words of size at most n.*

*Consider the Q×Q matrix*

$A_m(p,q)$ *= the number of words of length m accepted by a run from p to q.*

*Using unambiguity (and trimmed assumption), we have:*

$A_m(p,q)$ *= the number of runs from p to q of length m.*

*Let* $M(p,q)$ *= the number of letters triggering a transition from p to q.*

*We have* $A_{m+1} = A_m M$*.*

*Hence, the number of accepted words of length at most n is computable in time polynomial in n and Universality is in P.*

# Conclusion on finite words

# Conclusion on finite words

Unambiguous automata can be exponentially more succinct than deterministic ones.

# Conclusion on finite words

Unambiguous automata can be exponentially more succinct than deterministic ones.

Non-deterministic automata can be exponentially more succinct than unambiguous automata.

# Conclusion on finite words

Unambiguous automata can be exponentially more succinct than deterministic ones.

Non-deterministic automata can be exponentially more succinct than unambiguous automata.

Universality of unambiguous automata is in P, and the minimal witnesses are of linear length.

# Conclusion on finite words

Unambiguous automata can be exponentially more succinct than deterministic ones.

Non-deterministic automata can be exponentially more succinct than unambiguous automata.

Universality of unambiguous automata is in P, and the minimal witnesses are of linear length.

**Open question:** Is it possible to complement unambiguous automata in polynomial space ? (into an unambiguous or a non-deterministic)

# Conclusion on finite words

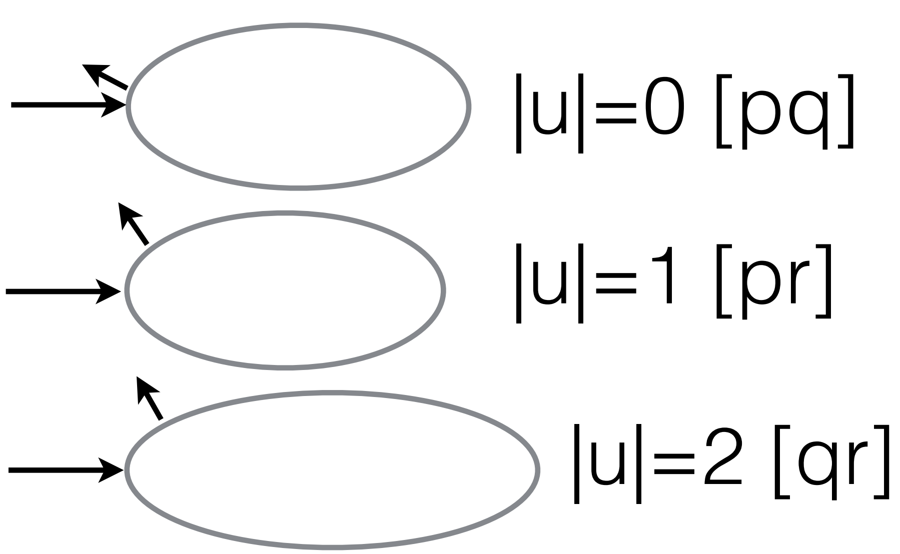
Unambiguous automata can be exponentially more succinct than deterministic ones.

Non-deterministic automata can be exponentially more succinct than unambiguous automata.

Universality of unambiguous automata is in P, and the minimal witnesses are of linear length.

**Open question:** Is it possible to complement unambiguous automata in polynomial space ? (into an unambiguous or a non-deterministic)

**Remark:** This cannot be done in linear space:
take three distinct primes p,q,r≥3.



# Conclusion on finite words

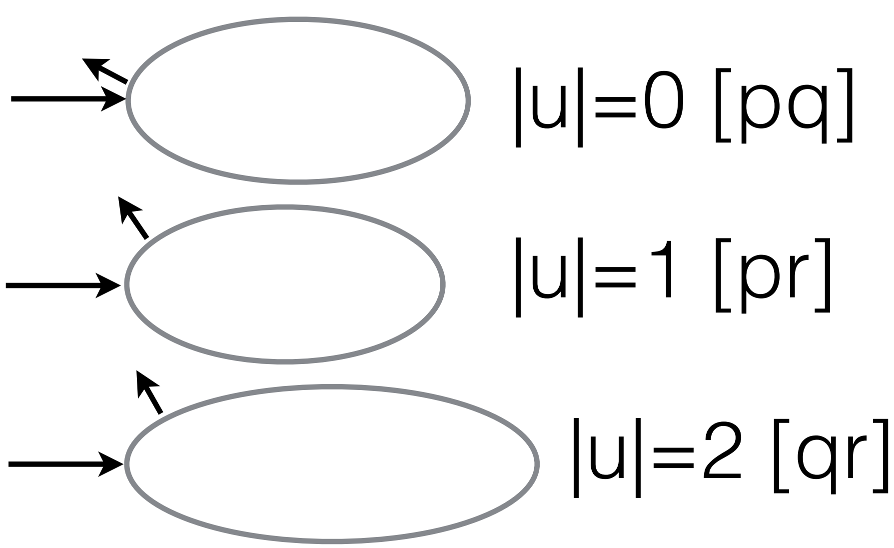
Unambiguous automata can be exponentially more succinct than deterministic ones.

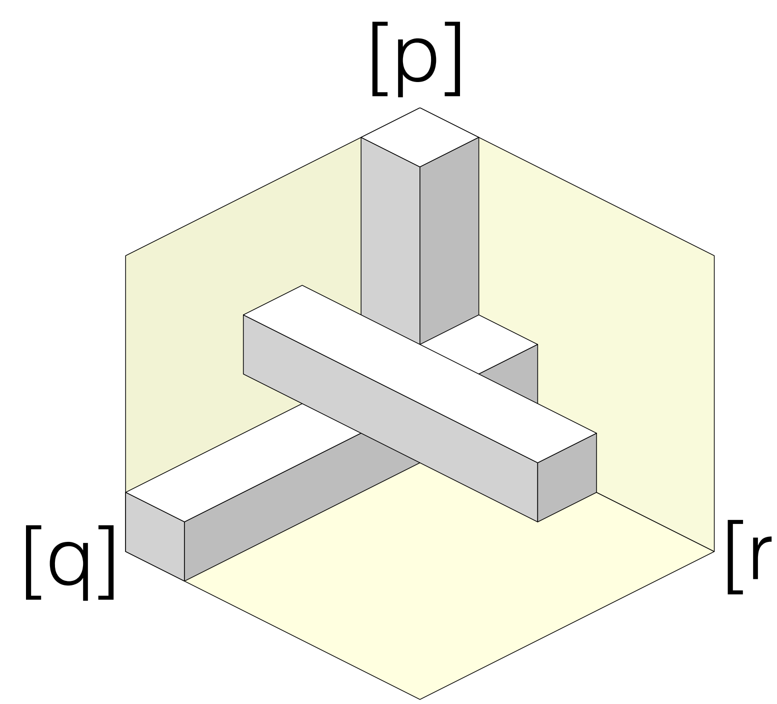
Non-deterministic automata can be exponentially more succinct than unambiguous automata.

Universality of unambiguous automata is in P, and the minimal witnesses are of linear length.

**Open question:** Is it possible to complement unambiguous automata in polynomial space ? (into an unambiguous or a non-deterministic)

**Remark:** This cannot be done in linear space:
take three distinct primes p,q,r≥3.



# Conclusion on finite words

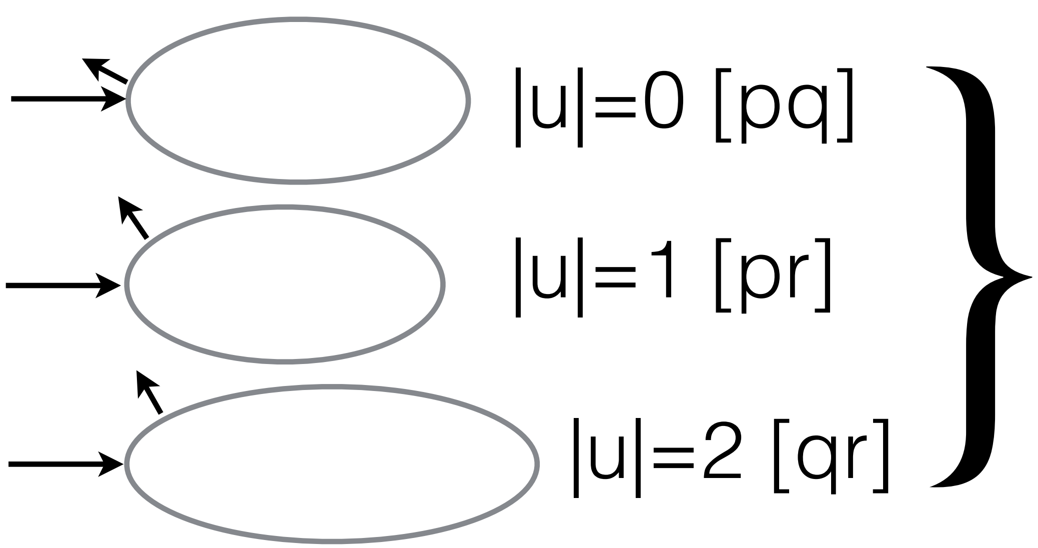
Unambiguous automata can be exponentially more succinct than deterministic ones.

Non-deterministic automata can be exponentially more succinct than unambiguous automata.

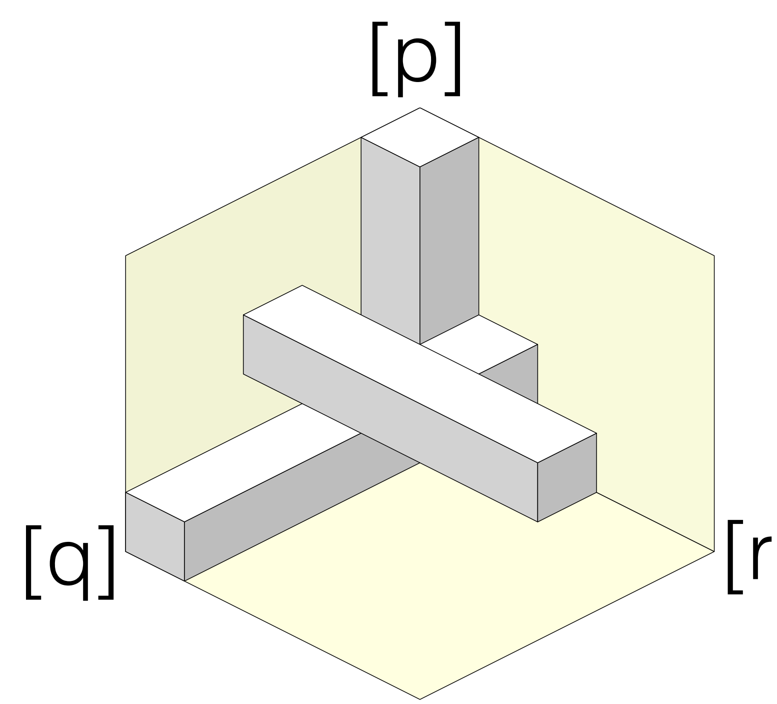
Universality of unambiguous automata is in P, and the minimal witnesses are of linear length.

**Open question:** Is it possible to complement unambiguous automata in polynomial space ? (into an unambiguous or a non-deterministic)

**Remark:** This cannot be done in linear space:
take three distinct primes p,q,r≥3.




An automaton
for the complement
has to contain a
cycle of length pqr.



# Unambiguous tropical automata

# Tropical automata

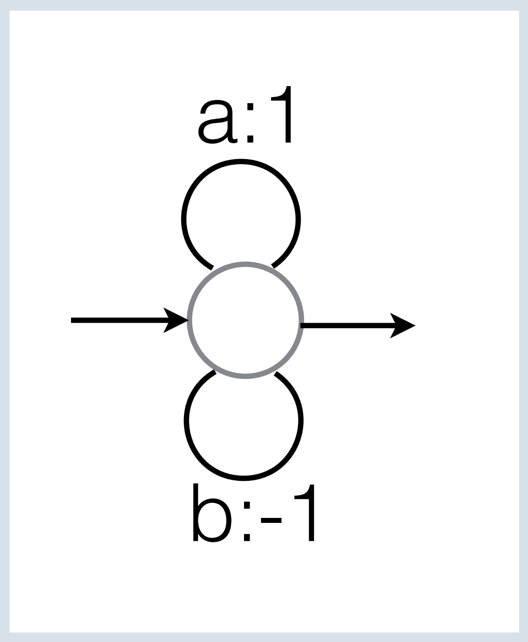
# Tropical automata

Hashiguchi
Simon

A tropical automaton is a non-deterministic automaton weighted by integers. It computes a function from words to integers (and +/-∞).

# Tropical automata

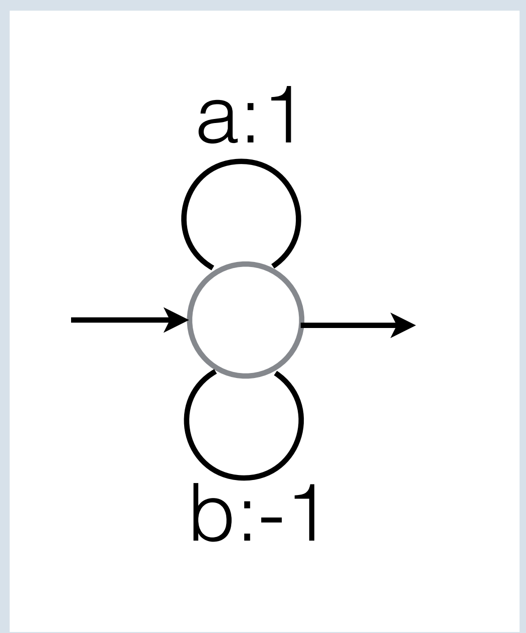
Hashiguchi
Simon

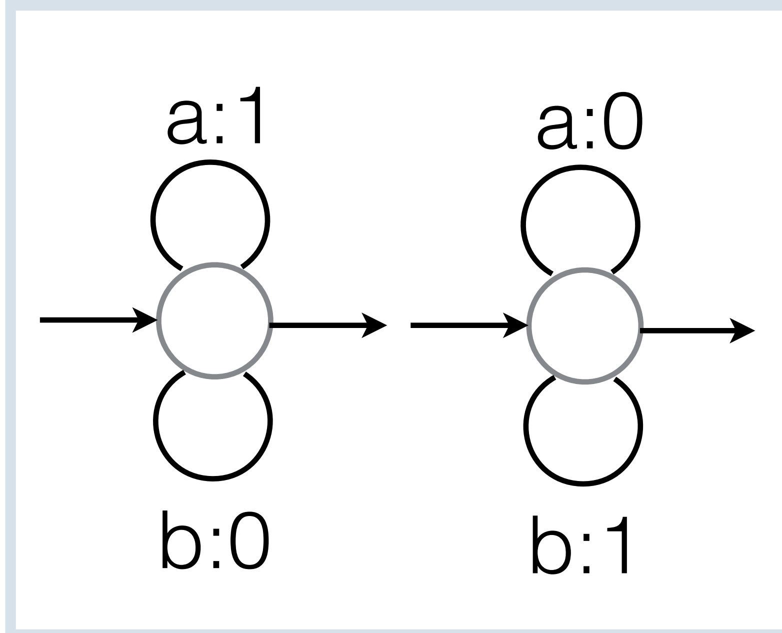



A tropical automaton is a non-deterministic automaton weighted by integers. It computes a function from words to integers (and +/-∞).

# Tropical automata

Hashiguchi
Simon

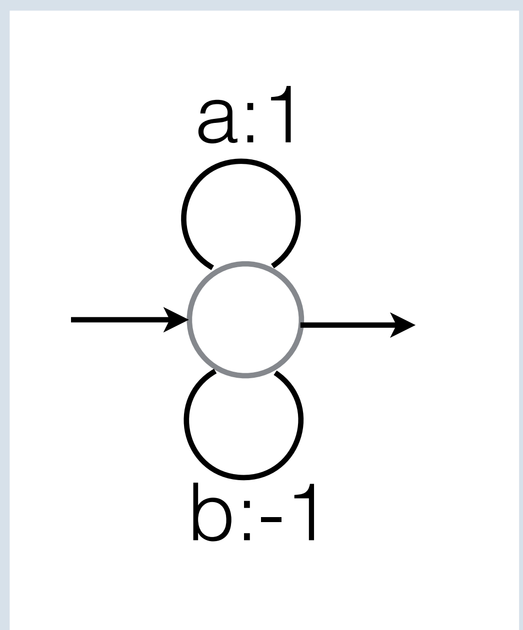


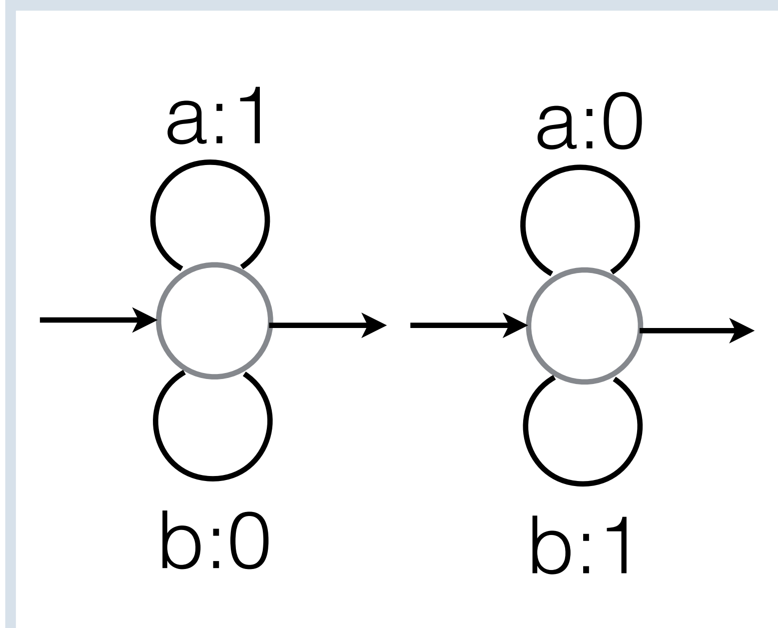



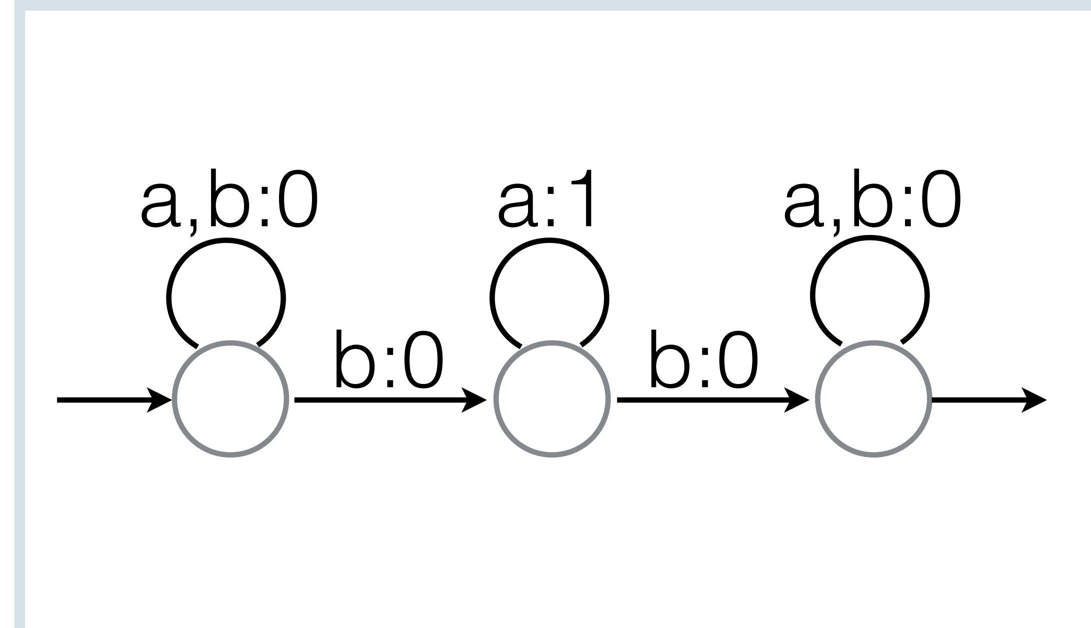

A tropical automaton is a non-deterministic automaton weighted by integers. It computes a function from words to integers (and +/-∞).

# Tropical automata

Hashiguchi
Simon



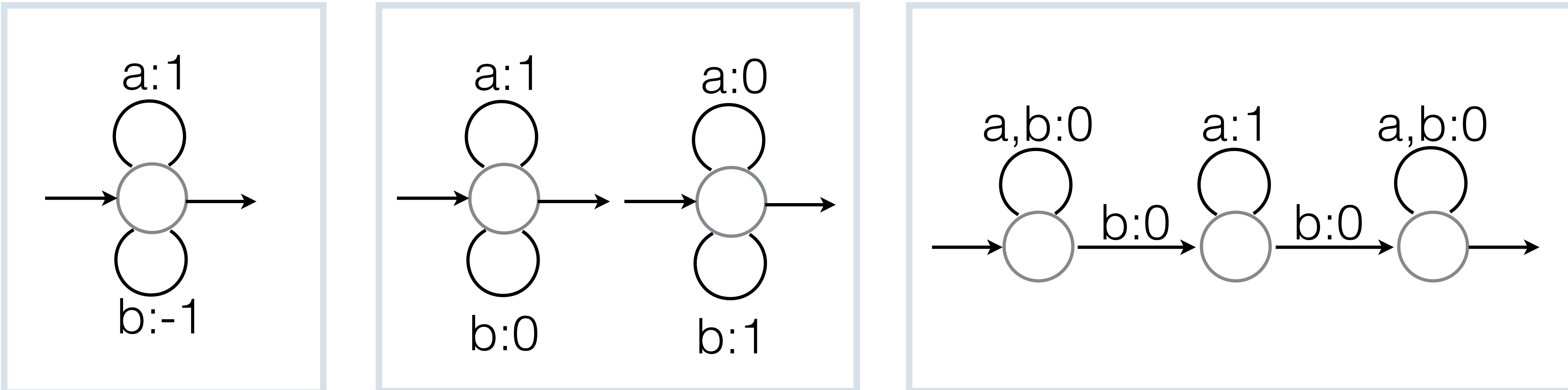




A tropical automaton is a non-deterministic automaton weighted by integers. It computes a function from words to integers (and +/-∞).

# Tropical automata

Hashiguchi
Simon

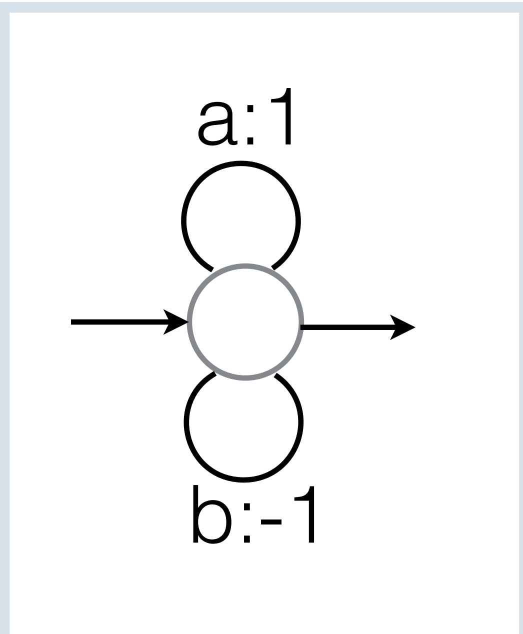


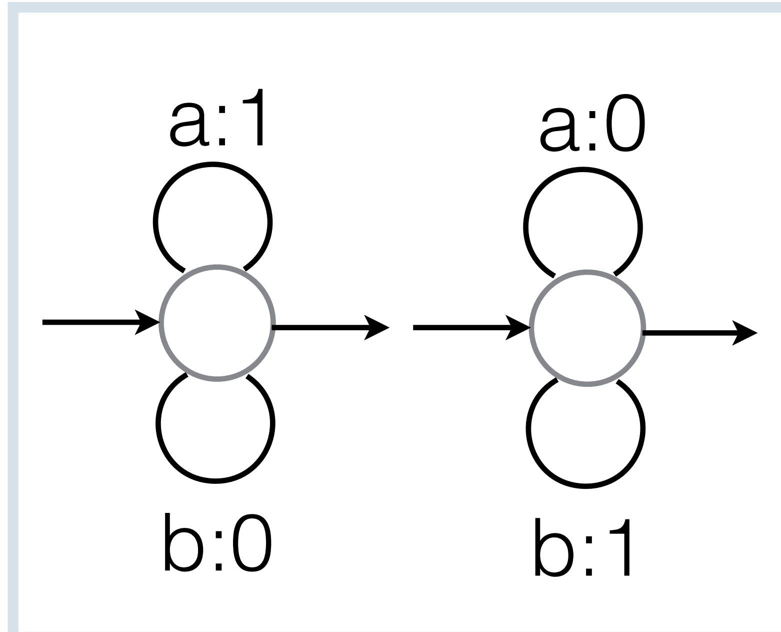

A tropical automaton is a non-deterministic automaton weighted by integers. It computes a function from words to integers (and +/-∞).

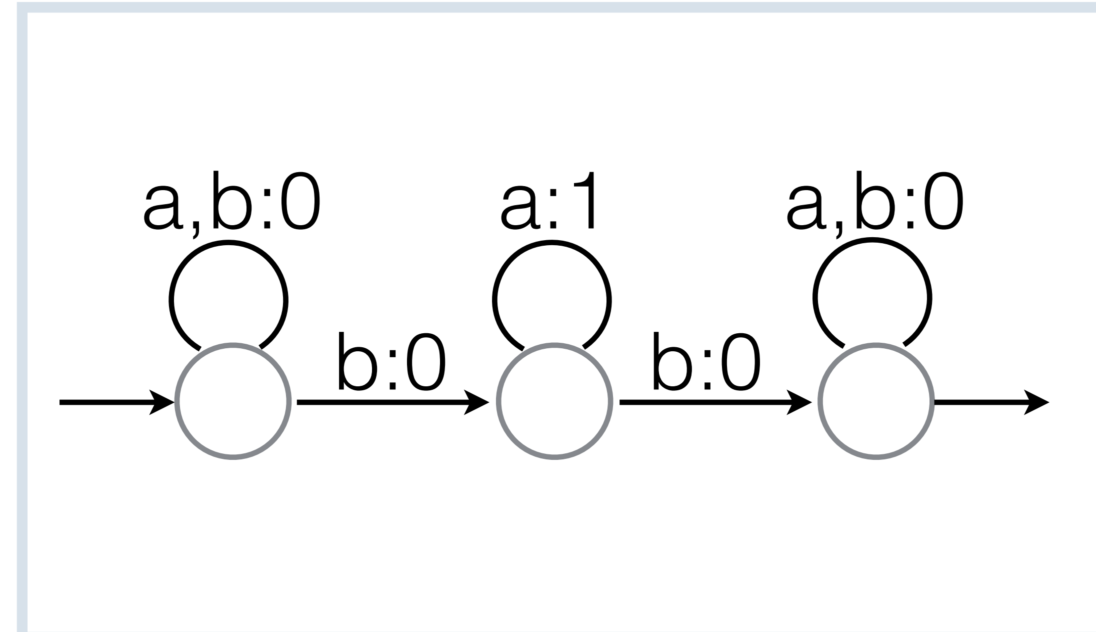
**min-+**: outputs the minimum over all accepting runs of the total weight

# Tropical automata

Hashiguchi
Simon



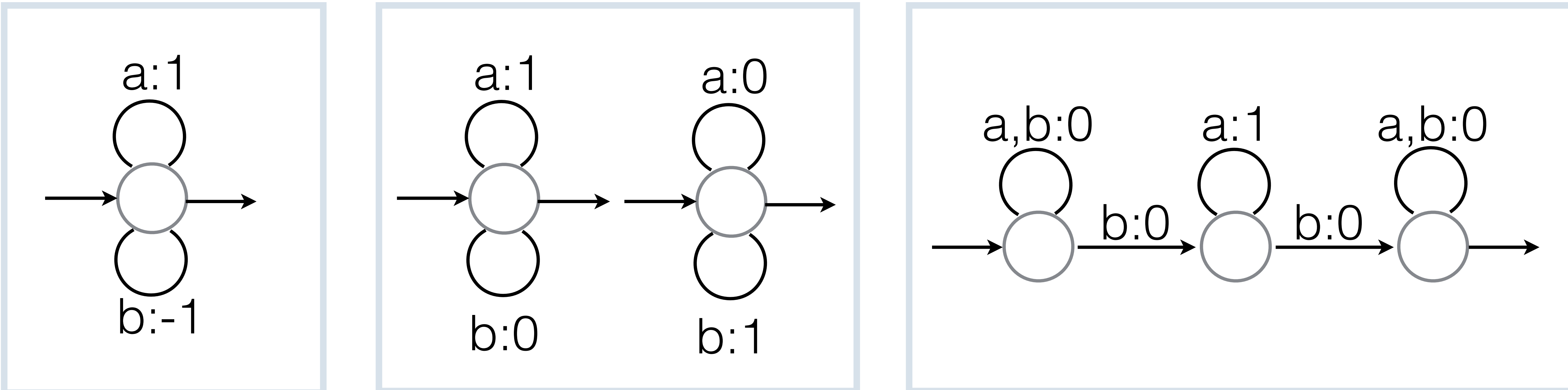

A tropical automaton is a non-deterministic automaton weighted by integers. It computes a function from words to integers (and +/-∞).

**min-+**: outputs the minimum over all accepting runs of the total weight

**max-+**: outputs the maximum over all accepting runs of the total weight

# Tropical automata

Hashiguchi
Simon



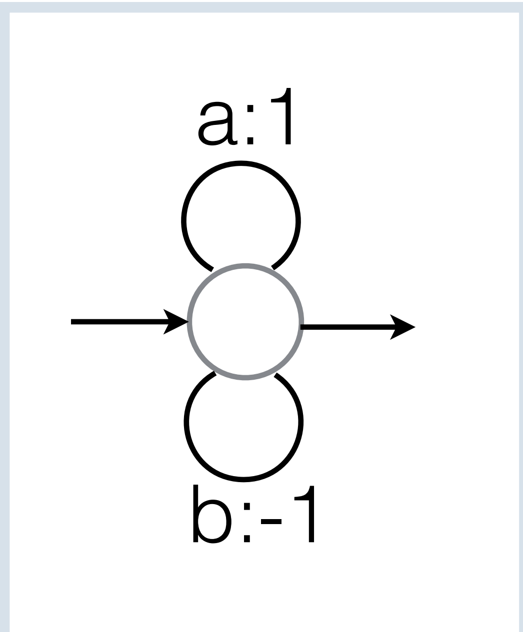

A tropical automaton is a non-deterministic automaton weighted by integers. It computes a function from words to integers (and +/-∞).

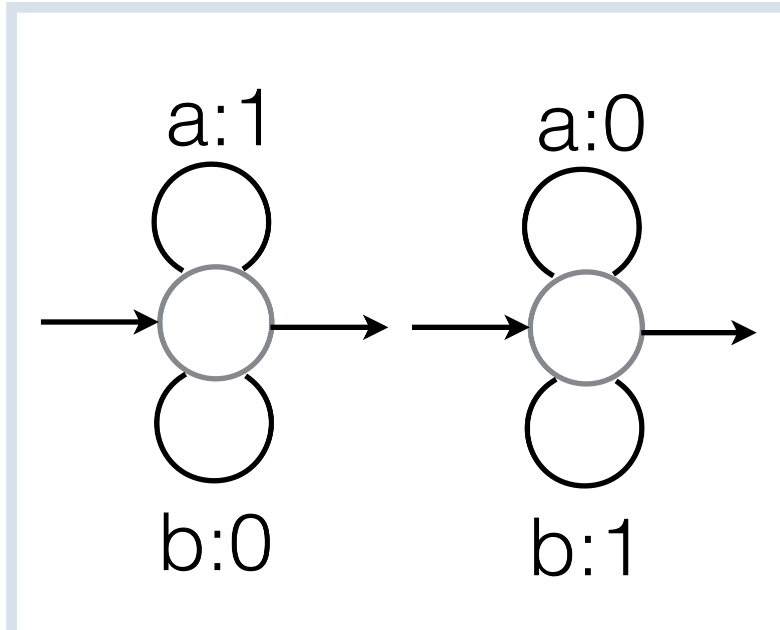
- **min-+**: outputs the minimum over all accepting runs of the total weight
- **max-+**: outputs the maximum over all accepting runs of the total weight

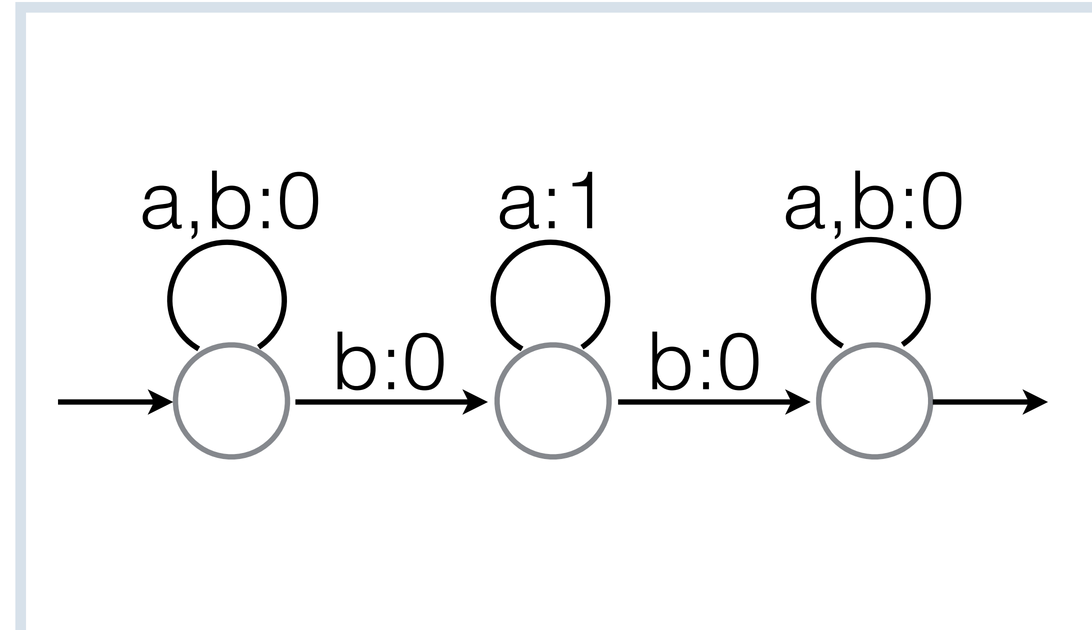
**Theorem:** It is decidable if a min-+ rational function f satisfies f≥0.
(resp. g≤0 for g max-+)

# Tropical automata

Hashiguchi
Simon




A tropical automaton is a non-deterministic automaton weighted by integers. It computes a function from words to integers (and +/-∞).

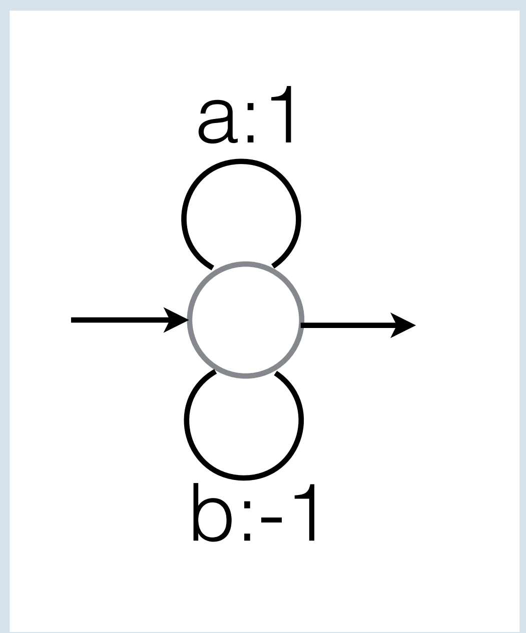
- **min-+**: outputs the minimum over all accepting runs of the total weight
- **max-+**: outputs the maximum over all accepting runs of the total weight

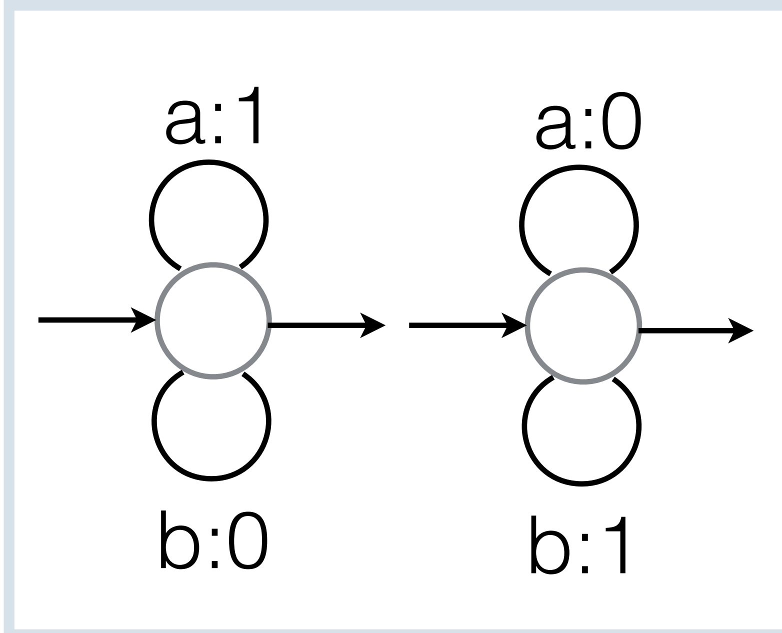
**Theorem:** It is decidable if a min-+ rational function f satisfies f≥0.
(resp. g≤0 for g max-+)

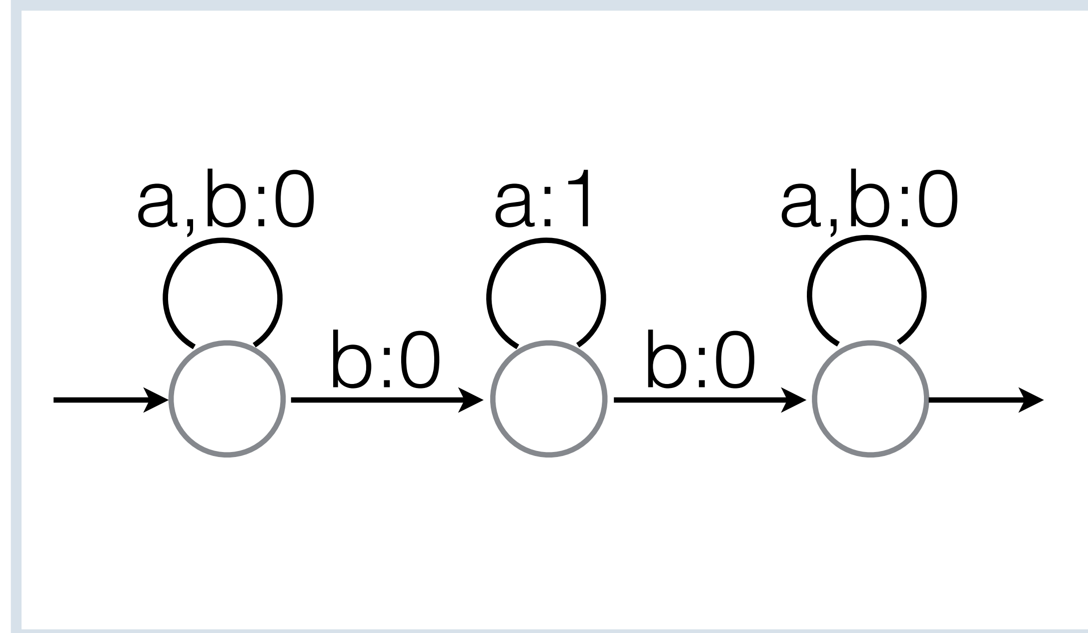
**Theorem [Krob94]:** It is undecidable if a max-+ rational function f satisfies f≥0. (resp. g≤0 for g min-+)

# Tropical automata

Hashiguchi
Simon




A tropical automaton is a non-deterministic automaton weighted by integers. It computes a function from words to integers (and +/-∞).

- **min-+**: outputs the minimum over all accepting runs of the total weight
- **max-+**: outputs the maximum over all accepting runs of the total weight

**Theorem:** It is decidable if a min-+ rational function f satisfies f≥0. (resp. g≤0 for g max-+)

**Theorem [Krob94]:** It is undecidable if a max-+ rational function f satisfies f≥0. (resp. g≤0 for g min-+)

Note that min-+ and max-+ semantics coincide over unambiguous automata. This yields unambiguous tropical automata.

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

No accepting run has a negative weight.

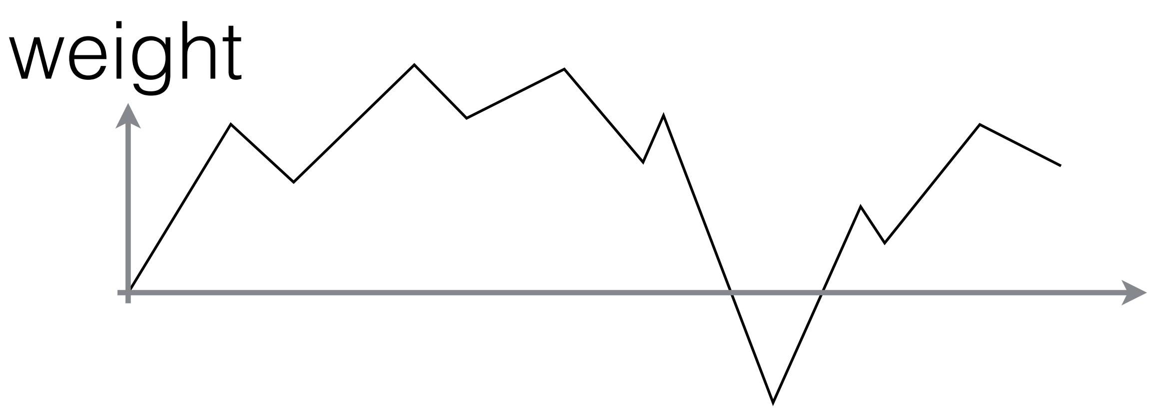
# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

No accepting run has a negative weight.

Proof: *Consider an initial run, and draw:*



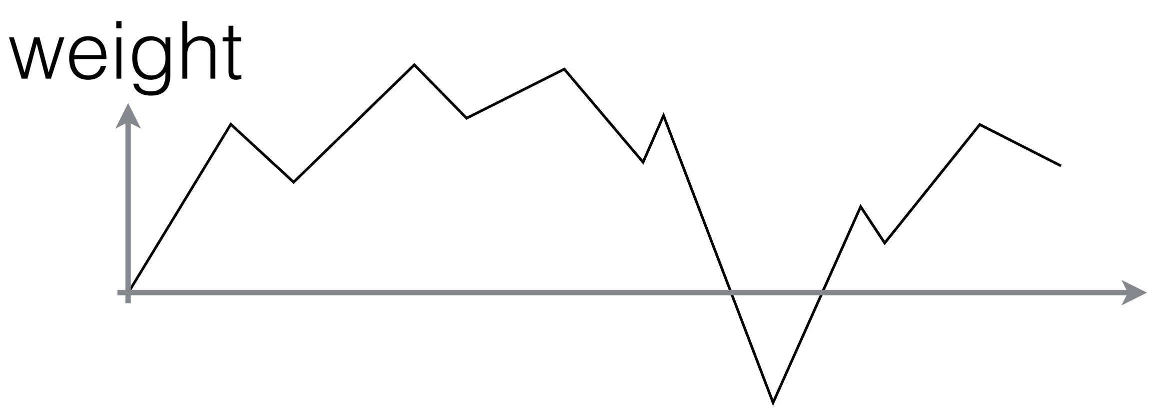
# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

No accepting run has a negative weight.

Proof: *Consider an initial run, and draw:*




Claim 1: *There exists k such that the run never goes below -k.*

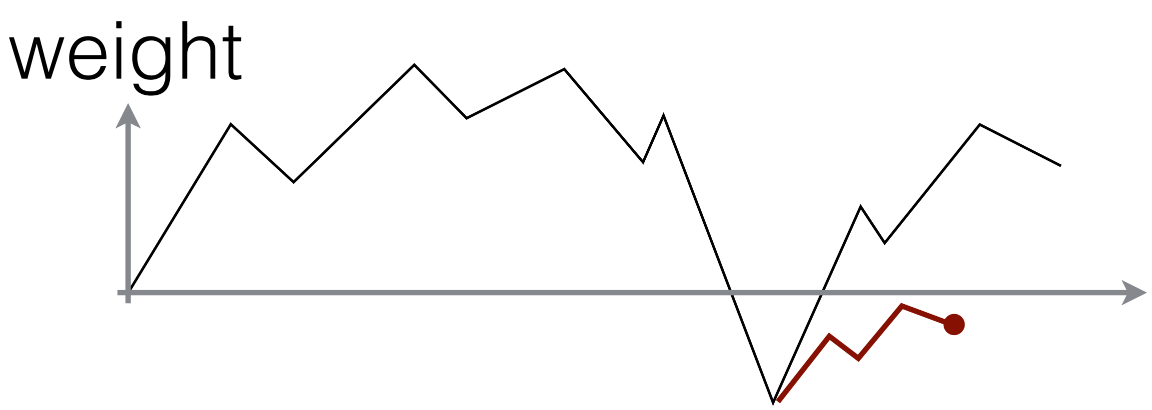
# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

No accepting run has a negative weight.

Proof: *Consider an initial run, and draw:*




Claim 1: *There exists k such that the run never goes below -k.*

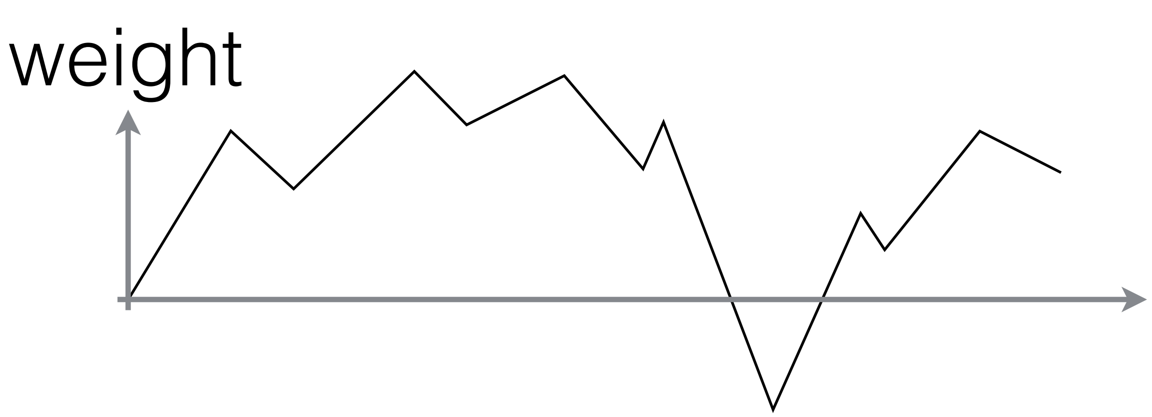
# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

No accepting run has a negative weight.

Proof: *Consider an initial run, and draw:*




Claim 1: *There exists k such that the run never goes below -k.*

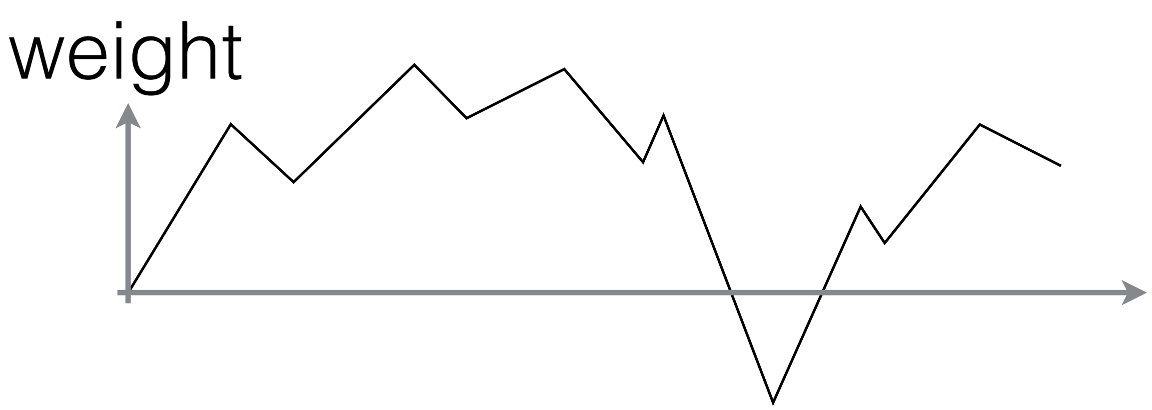
# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

No accepting run has a negative weight.

Proof: *Consider an initial run, and draw:*




Claim 2: *There is k' such that the partial weight cannot decrease of more than k'.*

Claim 1: *There exists k such that the run never goes below -k.*

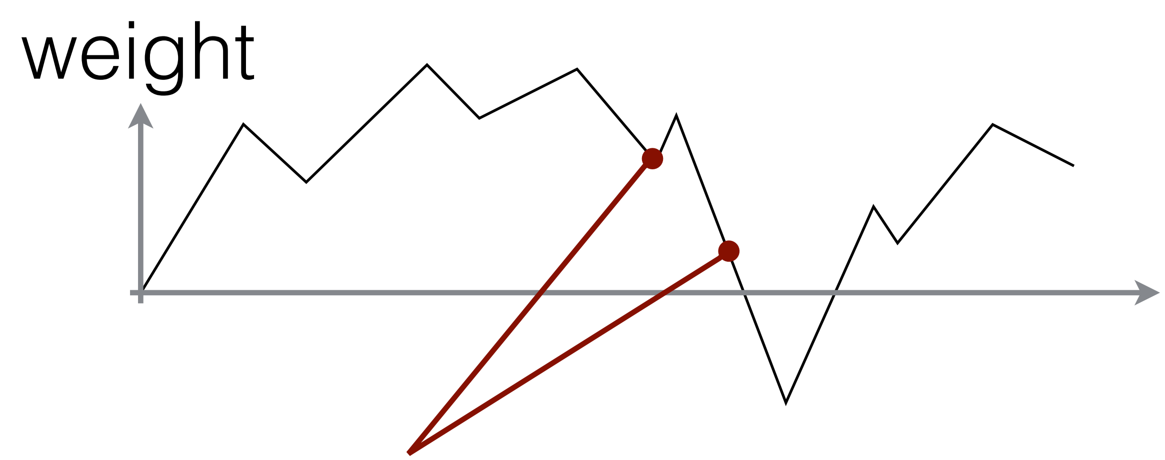
# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

No accepting run has a negative weight.

Proof: *Consider an initial run, and draw:*




Claim 2: *There is k' such that the partial weight cannot decrease of more than k'.*

Claim 1: *There exists k such that the run never goes below -k.*

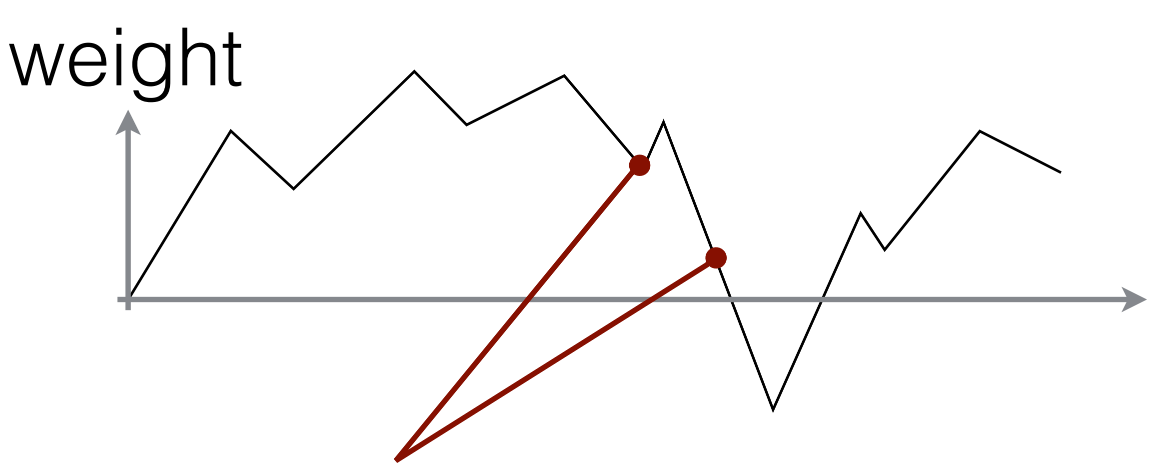
# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

No accepting run has a negative weight.

Proof: *Consider an initial run, and draw:*




*Thus, all runs of weight 0 have all their intermediate values the interval [-k,k'].*

Claim 2: *There is k' such that the partial weight cannot decrease of more than k'.*

Claim 1: *There exists k such that the run never goes below -k.*

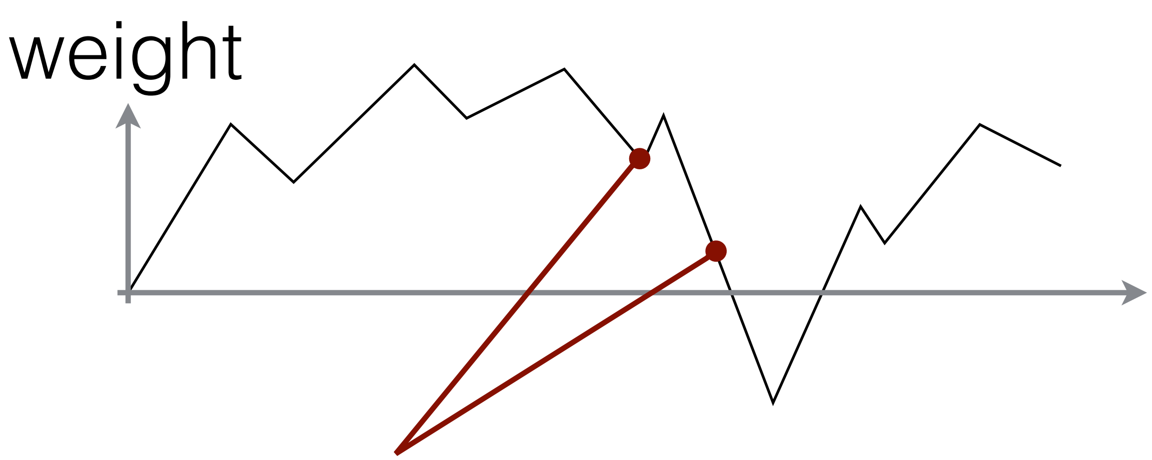
# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

No accepting run has a negative weight.

Proof: *Consider an initial run, and draw:*




Claim 2: *There is k' such that the partial weight cannot decrease of more than k'.*

Claim 1: *There exists k such that the run never goes below -k.*

*Thus, all runs of weight 0 have all their intermediate values the interval [-k,k'].*

*Hence, an automaton keeping weights in this interval can recognize runs of weight 0.*

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

( Along the same ideas:

**Proposition** [Krob94] **(Fatou property):**
If a min-+ rational function f is such that f≥0
Then it is recognized by a min-+ automaton with only non-negative weights. )

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that $f \geq 0$, the set of accepting runs of weight 0 is (effectively) regular.

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

Proof of the theorem: *Consider a min-+ automaton A for f, and a max-+ automaton B for g such that f=g.*

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

Proof of the theorem: *Consider a min-+ automaton A for f, and a max-+ automaton B for g such that f=g.*

*Construct the product automaton of A and B, with two weights*
*(a) the A weights, and*
*(b) the A-B weights.*

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

Proof of the theorem: *Consider a min-+ automaton A for f, and a max-+ automaton B for g such that f=g.*

*Construct the product automaton of A and B, with two weights*
*(a) the A weights, and*
*(b) the A-B weights.*

*The min-+ (b)-automaton computes f-g=0.*

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

Proof of the theorem: *Consider a min-+ automaton A for f, and a max-+ automaton B for g such that f=g.*

*Construct the product automaton of A and B, with two weights*
*(a) the A weights, and*
*(b) the A-B weights.*

*The min-+ (b)-automaton computes f-g=0.*

*By Lemma A, we can restrict it to the runs of (b)-weight 0.*

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

Proof of the theorem: *Consider a min-+ automaton A for f, and a max-+ automaton B for g such that f=g.*

*Construct the product automaton of A and B, with two weights*
*(a) the A weights, and*
*(b) the A-B weights.*

*The min-+ (b)-automaton computes f-g=0.*

*By Lemma A, we can restrict it to the runs of (b)-weight 0.*

*The resulting (a) automaton is such that:*

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

Proof of the theorem: *Consider a min-+ automaton A for f, and a max-+ automaton B for g such that f=g.*

*Construct the product automaton of A and B, with two weights*
*(a) the A weights, and*
*(b) the A-B weights.*

*The min-+ (b)-automaton computes f-g=0.*

*By Lemma A, we can restrict it to the runs of (b)-weight 0.*

*The resulting (a) automaton is such that:*

*a) all inputs u have an accepting run*

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

Proof of the theorem: *Consider a min-+ automaton A for f, and a max-+ automaton B for g such that f=g.*

*Construct the product automaton of A and B, with two weights*
*(a) the A weights, and*
*(b) the A-B weights.*

*The min-+ (b)-automaton computes f-g=0.*

*By Lemma A, we can restrict it to the runs of (b)-weight 0.*

*The resulting (a) automaton is such that:*

*a) all inputs u have an accepting run*
*b) all accepting runs have the weight f(u)*

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ rational is (effectively) recognized by an unambiguous tropical automaton.

**Lemma A:** Given a min-+ function f such that f≥0, the set of accepting runs of weight 0 is (effectively) regular.

Proof of the theorem: *Consider a min-+ automaton A for f, and a max-+ automaton B for g such that f=g.*

*Construct the product automaton of A and B, with two weights*
*(a) the A weights, and*
*(b) the A-B weights.*

*The min-+ (b)-automaton computes f-g=0.*

*By Lemma A, we can restrict it to the runs of (b)-weight 0.*

*The resulting (a) automaton is such that:*

*a) all inputs u have an accepting run*
*b) all accepting runs have the weight f(u)*

*It can be made unambiguous by keeping the lexicographic least run.*

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ recognized is (effectively) recognized by an unambiguous tropical automaton.

# Being both min and max

**Theorem** [Lombardy&Mairesse06]: A function from words to integers that is both min-+ and max-+ recognized is (effectively) recognized by an unambiguous tropical automaton.

**Conjecture (separation):** Given a max-+ regular function f and a min-+ regular function g such that $f \leq g$, then there exists an unambiguous regular function h such that

$$f \leq h \leq g \,.$$

# Unambiguity for other forms of automata

# Other forms of automata

Transducers

Infinite word automata

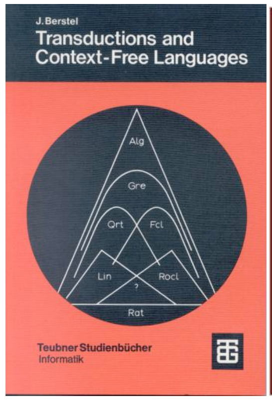
Register automata

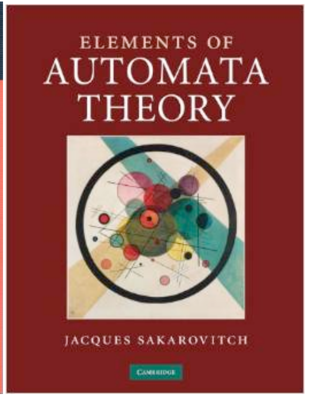
Infinite tree automata

# Other forms of automata

## Transducers

A transducer that recognizes a relation that happens to be a function is equivalent to an unambiguous transducer.



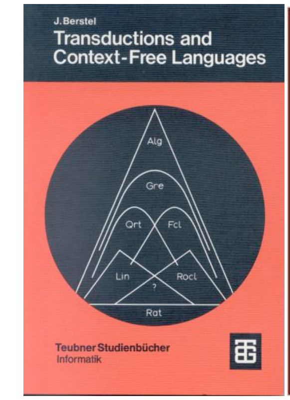



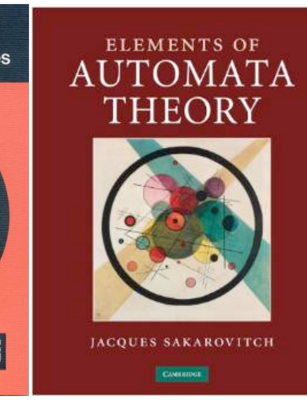

## Infinite word automata

## Register automata

## Infinite tree automata

# Other forms of automata

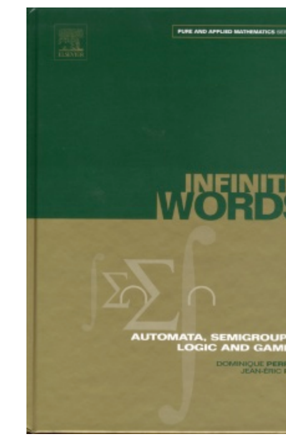



## Transducers

A transducer that recognizes a relation that happens to be a function is equivalent to an unambiguous transducer.

## Infinite word automata

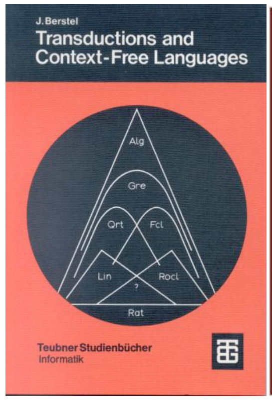


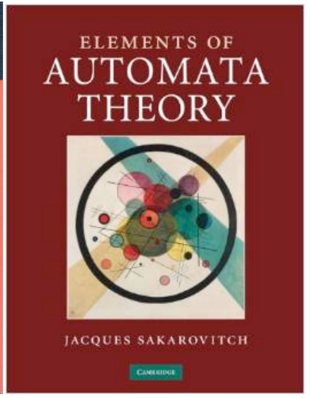

Non-deterministic Büchi automata are closed under Boolean operations and projection, but can't be determinized!
However, these can be made unambiguous [Carton&Michel00] (prophetic).

## Register automata

## Infinite tree automata

# Other forms of automata

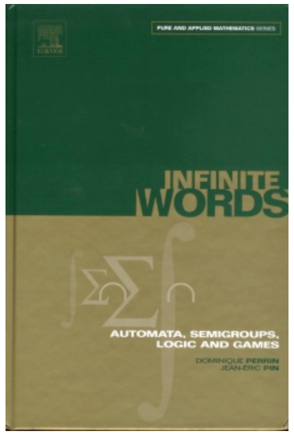



## Transducers

A transducer that recognizes a relation that happens to be a function is equivalent to an unambiguous transducer.

## Infinite word automata



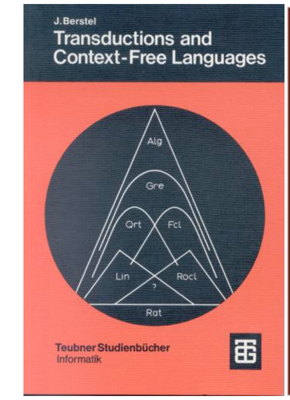

Non-deterministic Büchi automata are closed under Boolean operations and projection, but can't be determinized! However, these can be made unambiguous [Carton&Michel00] (prophetic).

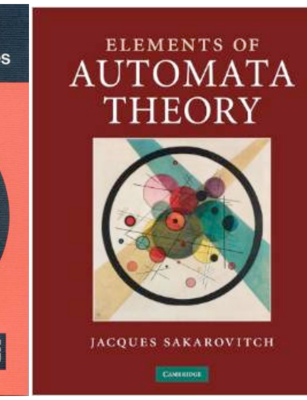
## Register automata

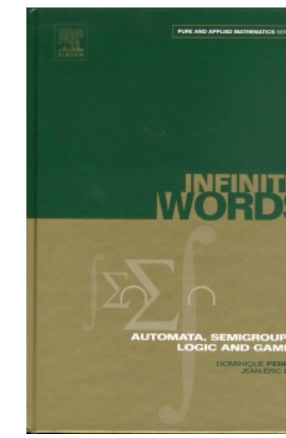
Register automata are not closed under complemented and universality is undecidable.

## Infinite tree automata

# Other forms of automata




## Transducers

A transducer that recognizes a relation that happens to be a function is equivalent to an unambiguous transducer.

## Infinite word automata

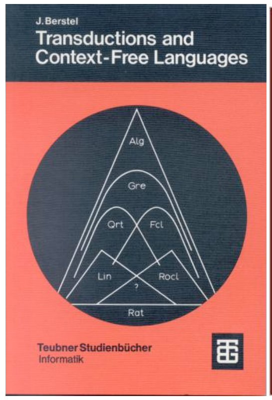
Non-deterministic Büchi automata are closed under Boolean operations and projection, but can't be determinized!
However, these can be made unambiguous [Carton&Michel00] (prophetic).

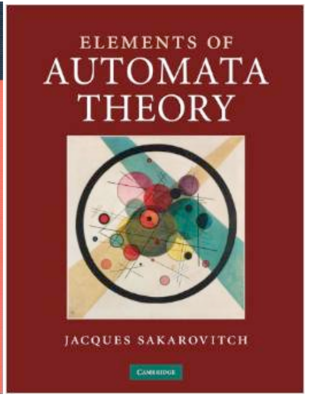
## Register automata

Register automata are not closed under complemented and universality is undecidable.
Conjecture: Unambiguous automata are closed under complement, and universality is decidable.

## Infinite tree automata

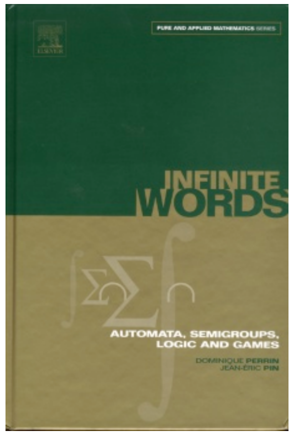
# Other forms of automata






## Transducers

A transducer that recognizes a relation that happens to be a function is equivalent to an unambiguous transducer.




## Infinite word automata

Non-deterministic Büchi automata are closed under Boolean operations and projection, but can't be determinized! However, these can be made unambiguous [Carton&Michel00] (prophetic).

## Register automata

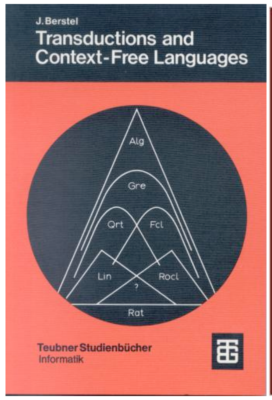
Register automata are not closed under complemented and universality is undecidable.

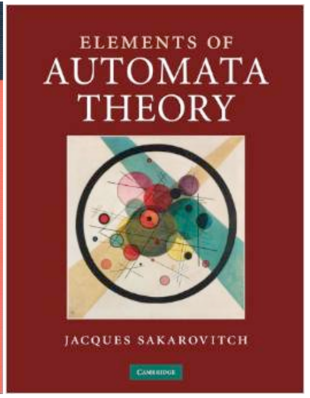
Conjecture: Unambiguous automata are closed under complement, and universality is decidable.

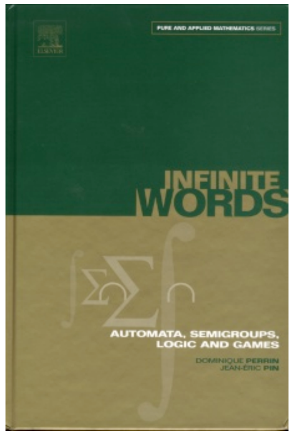
## Infinite tree automata

Unambiguous infinite tree automata are strictly weaker than non-deterministic ones. Related to the non-definability of choice.

# Other forms of automata




## Transducers

A transducer that recognizes a relation that happens to be a function is equivalent to an unambiguous transducer.

## Infinite word automata

Non-deterministic Büchi automata are closed under Boolean operations and projection, but can't be determinized!
However, these can be made unambiguous [Carton&Michel00] (prophetic).

## Register automata

Register automata are not closed under complemented and universality is undecidable.

Conjecture: Unambiguous automata are closed under complement, and universality is decidable.

## Infinite tree automata

Unambiguous infinite tree automata are strictly weaker than non-deterministic ones. Related to the non-definability of choice.

Problem: Decide if a language is accepted by an unambiguous automaton.

# Conclusion

# To conclude

# To conclude

Unambiguity arises naturally in automata theory in many situations:

- When non-deterministic automata are too wild (bad complexity or even undecidability of universality/equivalence, etc),
  but deterministic are too weak (because not closed under mirror…)
- When regular lookahead is added to deterministic automata.
- When this corresponds to a characterization result
  (e.g. min-plus and max-plus, transducers).

# To conclude

Unambiguity arises naturally in automata theory in many situations:

- When non-deterministic automata are too wild (bad complexity or even undecidability of universality/equivalence, etc),
  but deterministic are too weak (because not closed under mirror…)
- When regular lookahead is added to deterministic automata.
- When this corresponds to a characterization result
  (e.g. min-plus and max-plus, transducers).

The situations are very different depending of the model of automata under consideration.

# To conclude

Unambiguity arises naturally in automata theory in many situations:

- When non-deterministic automata are too wild (bad complexity or even undecidability of universality/equivalence, etc),
  but deterministic are too weak (because not closed under mirror…)
- When regular lookahead is added to deterministic automata.
- When this corresponds to a characterization result
  (e.g. min-plus and max-plus, transducers).

The situations are very different depending of the model of automata under consideration.

Some interesting tools are available
(e.g., communication complexity and rank).

# To conclude

Unambiguity arises naturally in automata theory in many situations:

- When non-deterministic automata are too wild (bad complexity or even undecidability of universality/equivalence, etc),
  but deterministic are too weak (because not closed under mirror…)
- When regular lookahead is added to deterministic automata.
- When this corresponds to a characterization result
  (e.g. min-plus and max-plus, transducers).

The situations are very different depending of the model of automata under consideration.

Some interesting tools are available
(e.g., communication complexity and rank).

Things are not yet well understood, and many questions remain open.

# Some open problems

# Some open problems

Is it polynomial to complement unambiguous word automata?

Is it possible to separate disjoint non-deterministic automata by unambiguous automata of polynomial size?

Is it possible to separate min-+ and max-+ automata by unambiguous tropical automata?

Can we decide if a min-+ automaton is equivalent to an unambiguous one?

Can we complement unambiguous register automata, and decide universality?

Is it possible to separate register automata by unambiguous ones?

Is it possible to decide if a language of infinite trees is recognized by some unambiguous automaton?

Are unambiguous automata over tame trees as expressive as general automata?