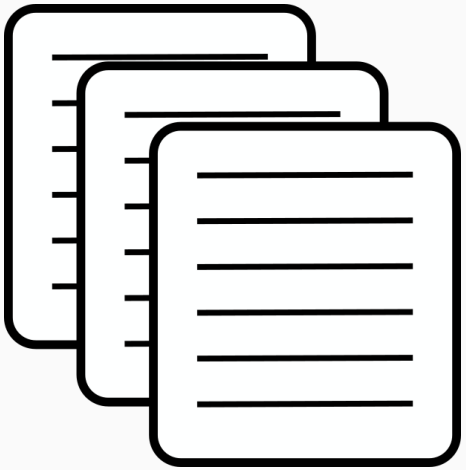
DIG Seminar April 2021

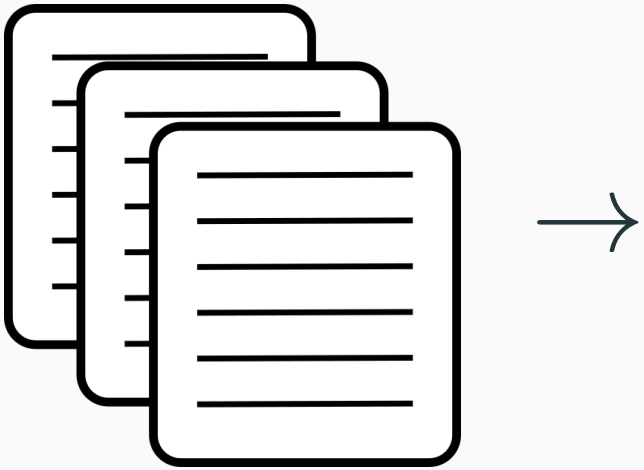
Ranked Enumeration of MSO Logic on Words

Louis Jachiet joint with Pierre Bourhis, Alejandro Grez and Cristian Riveros

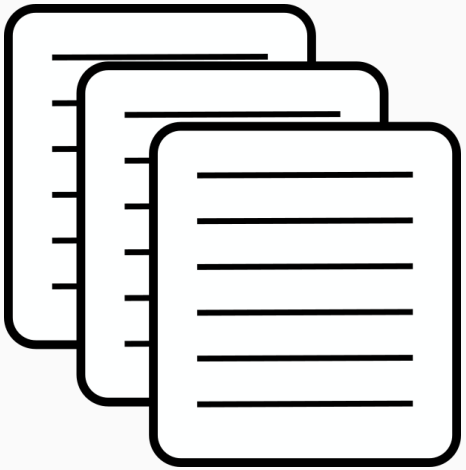
The problem



The problem



The problem

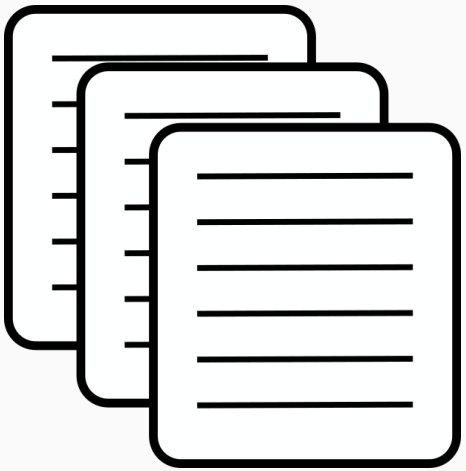


$\rightarrow$

Attribute_1	Attribute_2	Attribute_3
A_1	B_1	C_1
A_2	B_2	C_2
A_3	B_3	C_3
A_4	B_4	C_4

$\ldots$

The problem



$\rightarrow$

Attribute_1	Attribute_2	Attribute_3
A_1	B_1	C_1
A_2	B_2	C_2
A_3	B_3	C_3
A_4	B_4	C_4

...

Information Extraction

Enumeration algorithms

We use the enumeration framework where the computation is split into two phases:

- **The preprocessing phase**

  where we read the input and build some sort of index

- **The enumeration phase**

  where we actually output the solutions one-by-one

Enumeration algorithms

We use the enumeration framework where the computation is split into two phases:

- **The preprocessing phase**

  where we read the input and build some sort of index

  PREPROCESSING

- **The enumeration phase**

  where we actually output the solutions one-by-one

  ENUMERATION

Enumeration algorithms

We use the enumeration framework where the computation is split into two phases:

- **The preprocessing phase**

  where we read the input and build some sort of index

  PREPROCESSING TIME

- **The enumeration phase**

  where we actually output the solutions one-by-one

  DELAY

Enumeration algorithms

We use the enumeration framework where the computation is split into two phases:

- **The preprocessing phase**

  where we read the input and build some sort of index

  PREPROCESSING TIME

  Goal: Input Linear

- **The enumeration phase**

  where we actually output the solutions one-by-one

  DELAY

  Goal: Constant

Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit 2020-03-14 aliquam. Mauris suscipit, suscipit est, sit amet laoreet lorem nisl non lorem. Nam quis eleifend mauris. Donec volutpat turpis rutrum magna pellentesque, sed sodales leo blandit. Proin dolor purus, fringilla non nisl sed, fermentum tincidunt mi. In placerat 2012-01-03 nulla eget leo laoreet, vel gravida lorem finibus. Integer nec imperdiet mauris. Mauris a porttitor ipsum. Nulla facilisi. Phasellus scelerisque hendrerit velit, at euismod nulla 1952-05-12 consequat et. 2015-11-05 massa, id cursus.

Extract dates

date

Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit 2020-03-14 aliquam. Mauris suscipit, suscipit est, sit amet laoreet lorem nisl non lorem. Nam quis eleifend mauris. Donec volutpat turpis rutrum magna pellentesque, sed sodales leo blandit. Proin dolor purus, fringilla non nisl sed, fermentum tincidunt mi. In placerat 2012-01-03 nulla eget leo laoreet, vel gravida lorem finibus. Integer nec imperdiet mauris. Mauris a porttitor ipsum. Nulla facilisi. Phasellus scelerisque hendrerit velit, at euismod nulla 1952-05-12 consequat et. 2015-11-05 massa, id cursus.

Extract dates

date
2020-03-14

Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit 2020-03-14 aliquam. Mauris suscipit, suscipit est, sit amet laoreet lorem nisl non lorem. Nam quis eleifend mauris. Donec volutpat turpis rutrum magna pellentesque, sed sodales leo blandit. Proin dolor purus, fringilla non nisl sed, fermentum tincidunt mi. In placerat 2012-01-03 nulla eget leo laoreet, vel gravida lorem finibus. Integer nec imperdiet mauris. Mauris a porttitor ipsum. Nulla facilisi. Phasellus scelerisque hendrerit velit, at euismod nulla 1952-05-12 consequat et. 2015-11-05 massa, id cursus.

Extract dates

date
2020-03-14
2012-01-03

Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit 2020-03-14 aliquam. Mauris suscipit, suscipit est, sit amet laoreet lorem nisl non lorem. Nam quis eleifend mauris. Donec volutpat turpis rutrum magna pellentesque, sed sodales leo blandit. Proin dolor purus, fringilla non nisl sed, fermentum tincidunt mi. In placerat 2012-01-03 nulla eget leo laoreet, vel gravida lorem finibus. Integer nec imperdiet mauris. Mauris a porttitor ipsum. Nulla facilisi. Phasellus scelerisque hendrerit velit, at euismod nulla 1952-05-12 consequat et. 2015-11-05 massa, id cursus.

Extract dates

date
2020-03-14
2012-01-03
1952-05-12

Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit 2020-03-14 aliquam. Mauris suscipit, suscipit est, sit amet laoreet lorem nisl non lorem. Nam quis eleifend mauris. Donec volutpat turpis rutrum magna pellentesque, sed sodales leo blandit. Proin dolor purus, fringilla non nisl sed, fermentum tincidunt mi. In placerat 2012-01-03 nulla eget leo laoreet, vel gravida lorem finibus. Integer nec imperdiet mauris. Mauris a porttitor ipsum. Nulla facilisi. Phasellus scelerisque hendrerit velit, at euismod nulla 1952-05-12 consequat et. 2015-11-05 massa, id cursus.

Extract dates

date
2020-03-14
2012-01-03
1952-05-12
2015-11-05

Example

Mr. Emmanuel Macron was born on the 1977-12-21 and is the President of France since the 2017-05-14. Ms. Angela Merkel was born on the 1954-07-17 and she is the head of Germany since the 2005-11-22.

Extract pairs people-date appearing in the same sentence.

person	date

Example

Mr. Emmanuel Macron was born on the 1977-12-21 and is the President of France since the 2017-05-14. Ms. Angela Merkel was born on the 1954-07-17 and she is the head of Germany since the 2005-11-22.

Extract pairs people-date appearing in the same sentence.

person	date
Emmanuel Macron	1977-12-21

Example

Mr. Emmanuel Macron was born on the 1977-12-21 and is the President of France since the 2017-05-14. Ms. Angela Merkel was born on the 1954-07-17 and she is the head of Germany since the 2005-11-22.

Extract pairs people-date appearing in the same sentence.

person	date
Emmanuel Macron	1977-12-21
Emmanuel Macron	2017-05-14

Example

Mr. Emmanuel Macron was born on the 1977-12-21 and is the President of France since the 2017-05-14. Ms. Angela Merkel was born on the 1954-07-17 and she is the head of Germany since the 2005-11-22.

Extract pairs people-date appearing in the same sentence.

person	date
Emmanuel Macron	1977-12-21
Emmanuel Macron	2017-05-14
Angela Merkel	1954-07-17

Example

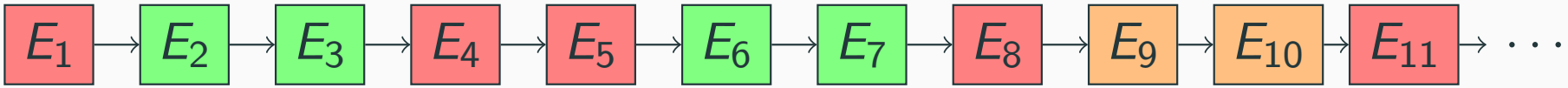
Mr. Emmanuel Macron was born on the 1977-12-21 and is the President of France since the 2017-05-14. Ms. Angela Merkel was born on the 1954-07-17 and she is the head of Germany since the 2005-11-22.

Extract pairs people-date appearing in the same sentence.

person	date
Emmanuel Macron	1977-12-21
Emmanuel Macron	2017-05-14
Angela Merkel	1954-07-17
Angela Merkel	2005-11-22

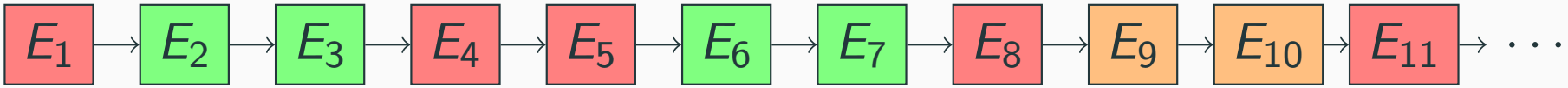
Another problem

A stream of events



Another problem

A stream of events

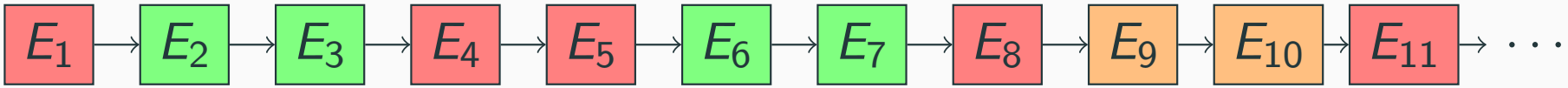


Pairs of red events with no green in-between:

Start	End

Another problem

A stream of events

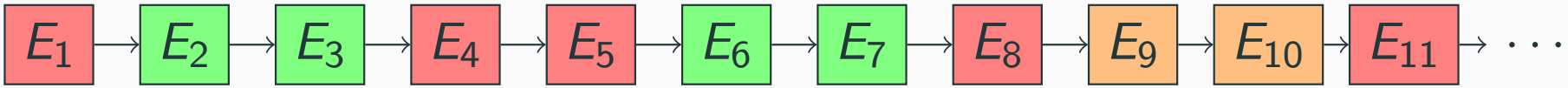


Pairs of red events with no green in-between:

Start	End
E_4	E_5

Another problem

A stream of events



Pairs of red events with no green in-between:

Start	End
E_4	E_5
E_8	E_{11}

Queries

Linearly ordered data

- Text
- Streams
- ...

Queries

Linearly ordered data

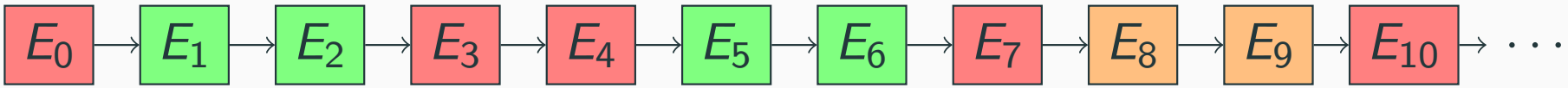
- Text
- Streams
- ...

Monadic Second Order logic

Basically "regular automaton logic"

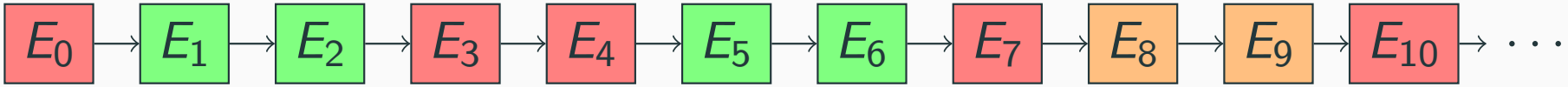
Another problem

A stream of events

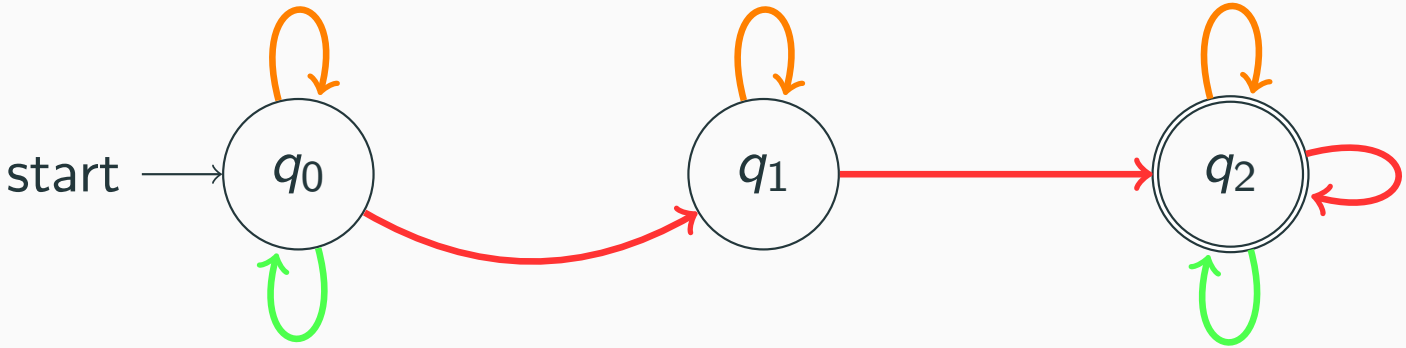


Another problem

A stream of events

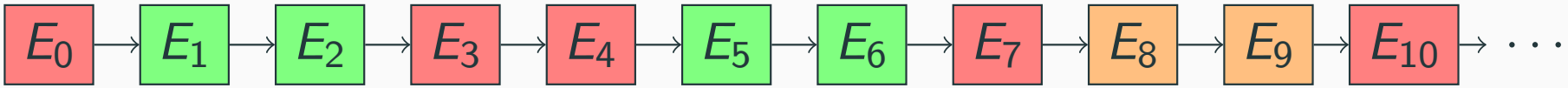


Pairs (x, y) of red events with no green in-between:

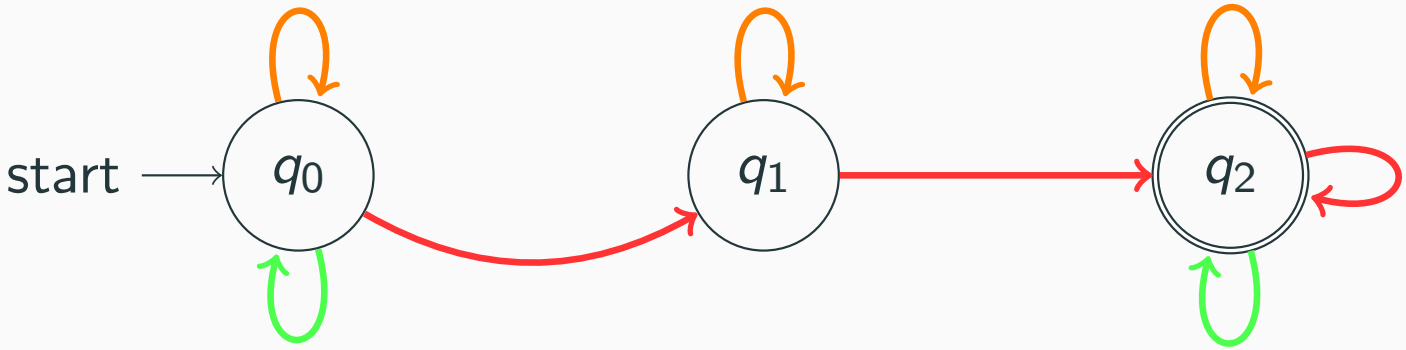


Another problem

A stream of events



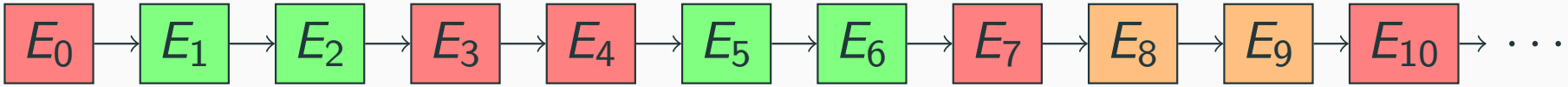
Pairs (x, y) of red events with no green in-between:



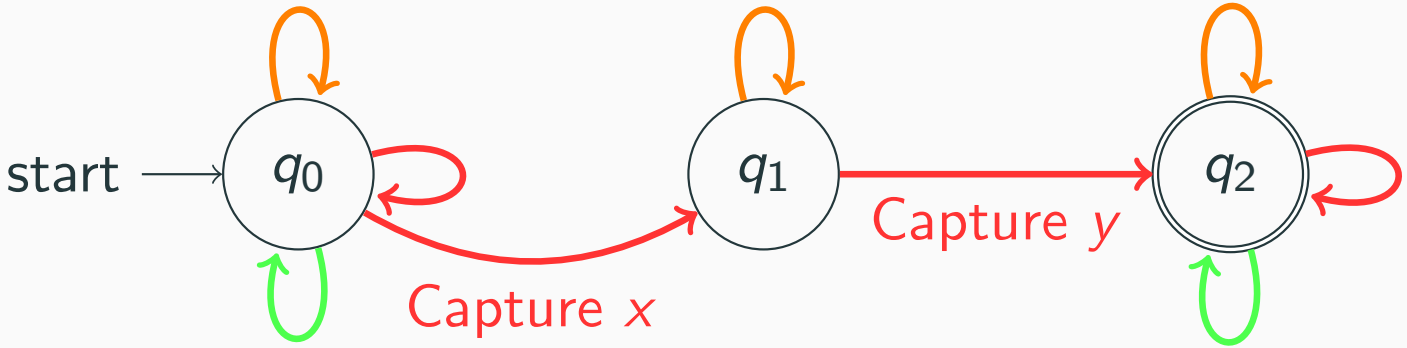
This accepts streams but what does it captures?

Another problem

A stream of events



Pairs (x, y) of red events with no green in-between:



Some formalism on automata

Run

A run of an automaton $\mathcal{A}$ on a word $w_0 \dots w_{n-1}$ is a sequence $q_0 \dots q_n$ such that:

- $q_i \xrightarrow{w_i} q_{i_1}$ is a valid transition of $\mathcal{A}$
- q_0 is the initial state of $\mathcal{A}$
- q_n is a final state of $\mathcal{A}$

Some formalism on automata

Run

A run of an automaton $\mathcal{A}$ on a word $w_0 \dots w_{n-1}$ is a sequence $q_0 \dots q_n$ such that:

- $q_i \xrightarrow{w_i} q_{i_1}$ is a valid transition of $\mathcal{A}$
- q_0 is the initial state of $\mathcal{A}$
- q_n is a final state of $\mathcal{A}$

Trace of a run

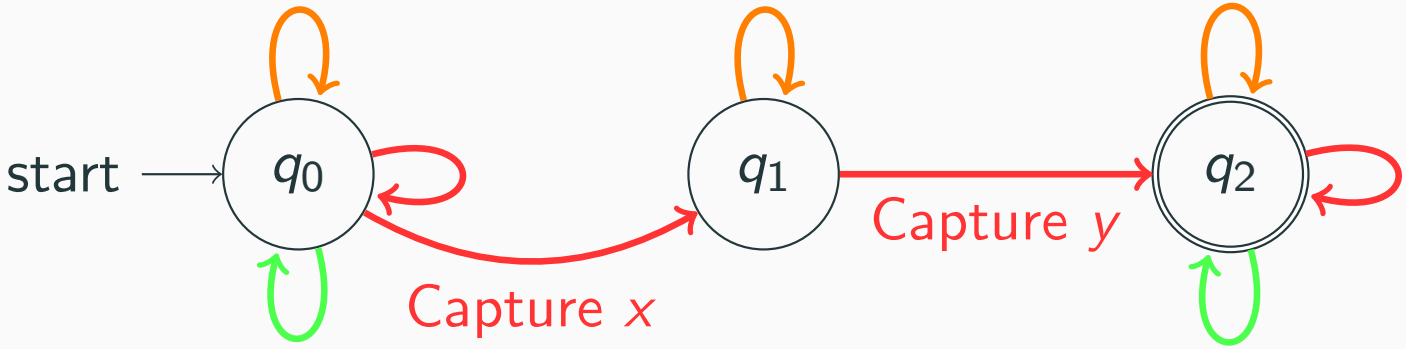
Each transition of $\mathcal{A}$ has an output. The trace of a run $q_0 \dots q_n$ is $\{(i, out(q_i, w_i, q_{i+1}) \mid out(q_i, w_i, q_{i+1}) \neq \emptyset\}$.

Unambiguous automaton (sort of determinism)

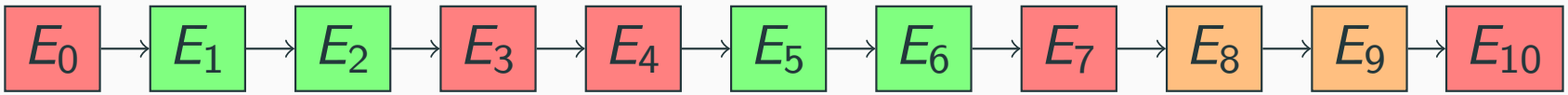
An automaton is unambiguous if we cannot have two different runs with the same trace.

Running an automaton

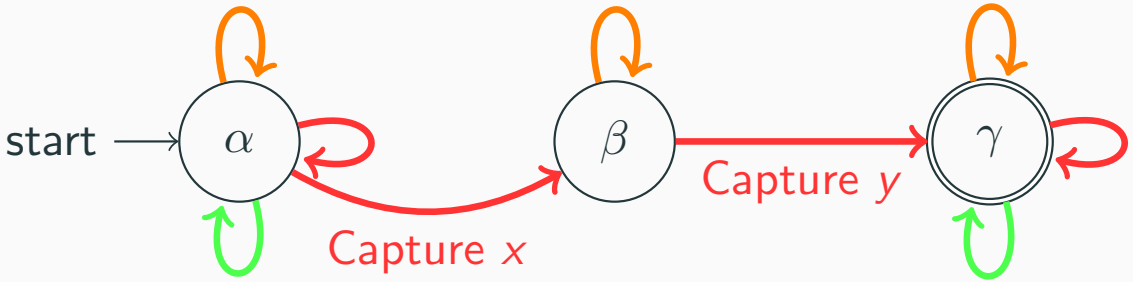
Automaton

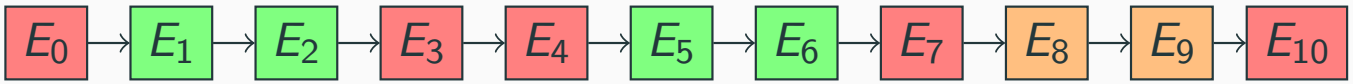


Run and traces

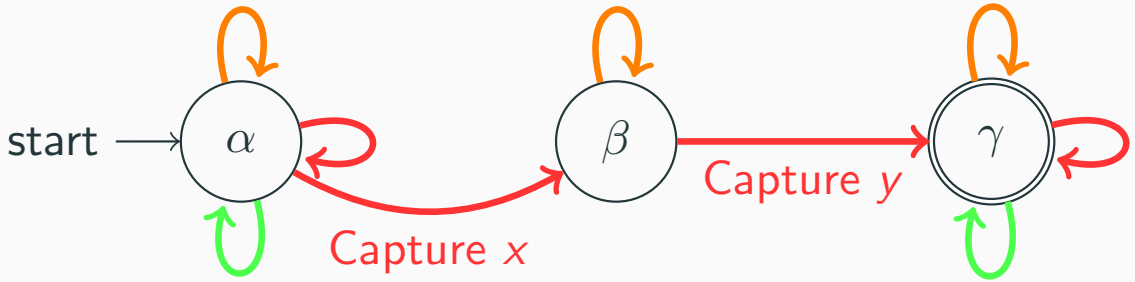


Product word - automaton





Product word - automaton



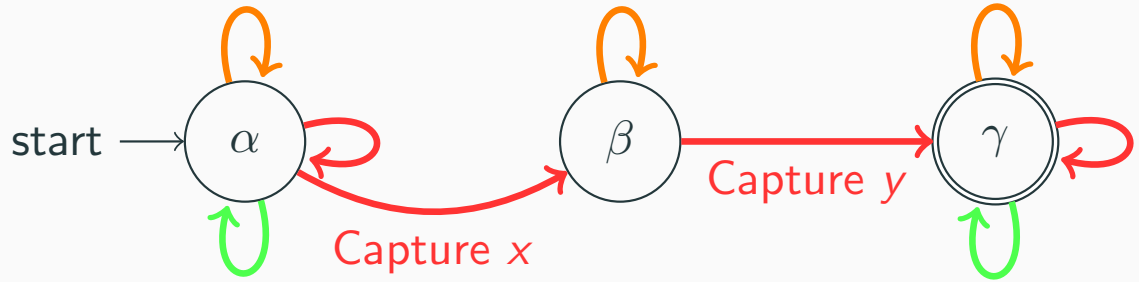
γ_0 γ_1 γ_2 γ_3 γ_4 γ_5 γ_6 γ_7 γ_8 γ_9 γ_{10} γ_{11}

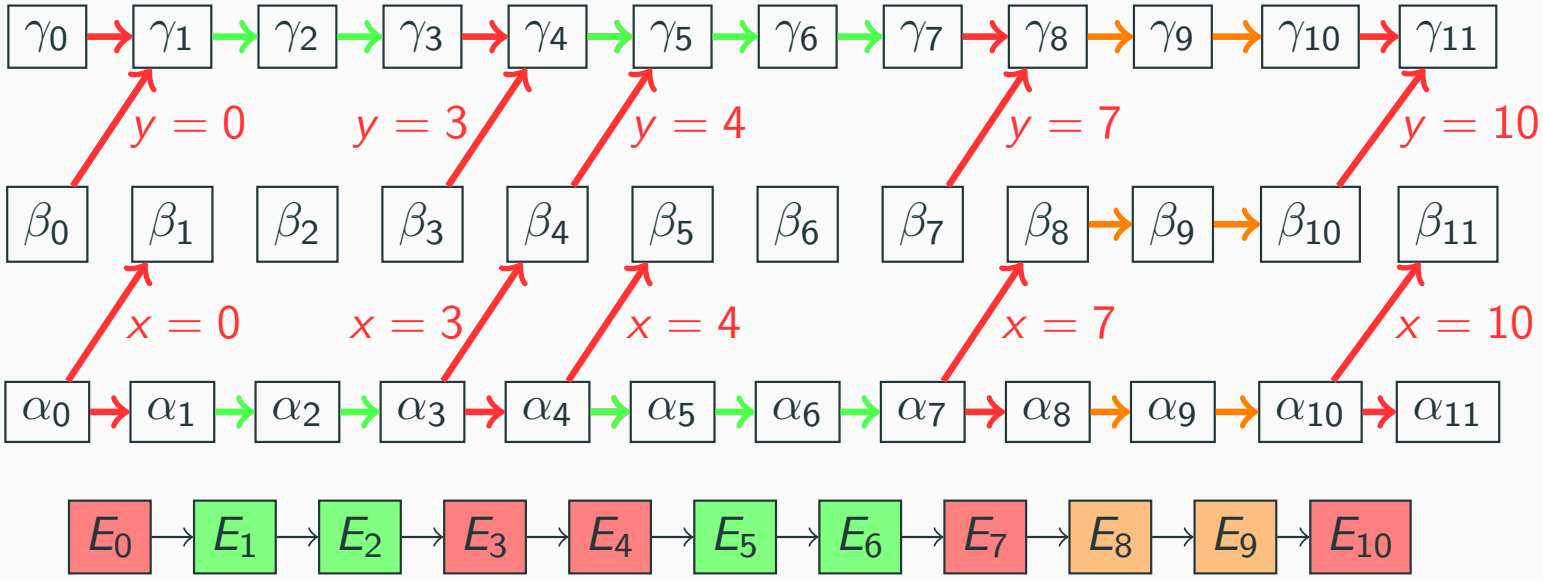
β_0 β_1 β_2 β_3 β_4 β_5 β_6 β_7 β_8 β_9 β_{10} β_{11}

α_0 α_1 α_2 α_3 α_4 α_5 α_6 α_7 α_8 α_9 α_{10} α_{11}

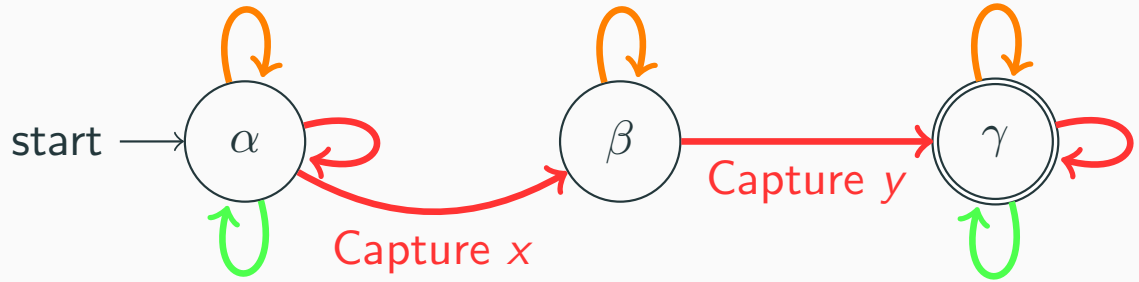
$E_0 \to E_1 \to E_2 \to E_3 \to E_4 \to E_5 \to E_6 \to E_7 \to E_8 \to E_9 \to E_{10}$

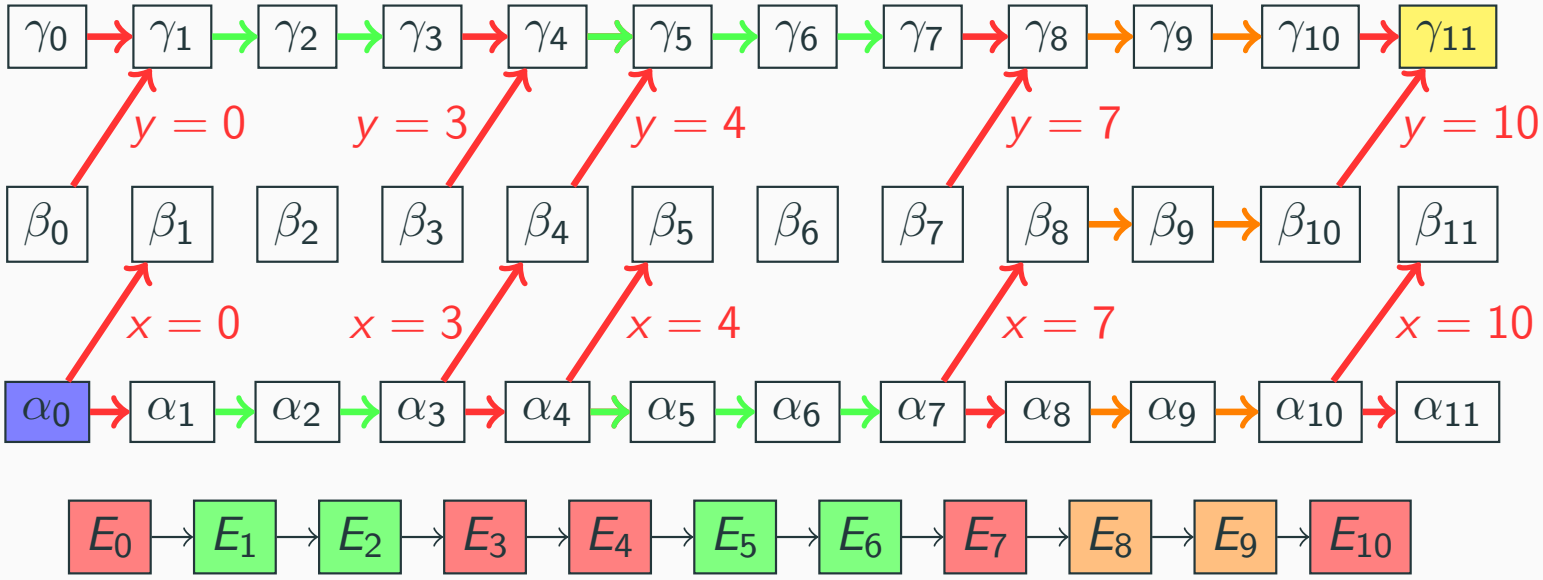
Product word - automaton





Product word - automaton





Enumeration of traces

boils down to

enumeration of

paths in a DAG

Enumeration order depends on

the algorithm used

and not on the

relevance of results

Cost MSO and

Ranked Enumeration of MSO

Logic on Words

Cost MSO Logic on Words

Idea

Each solution comes with a weight

Cost MSO Logic on Words

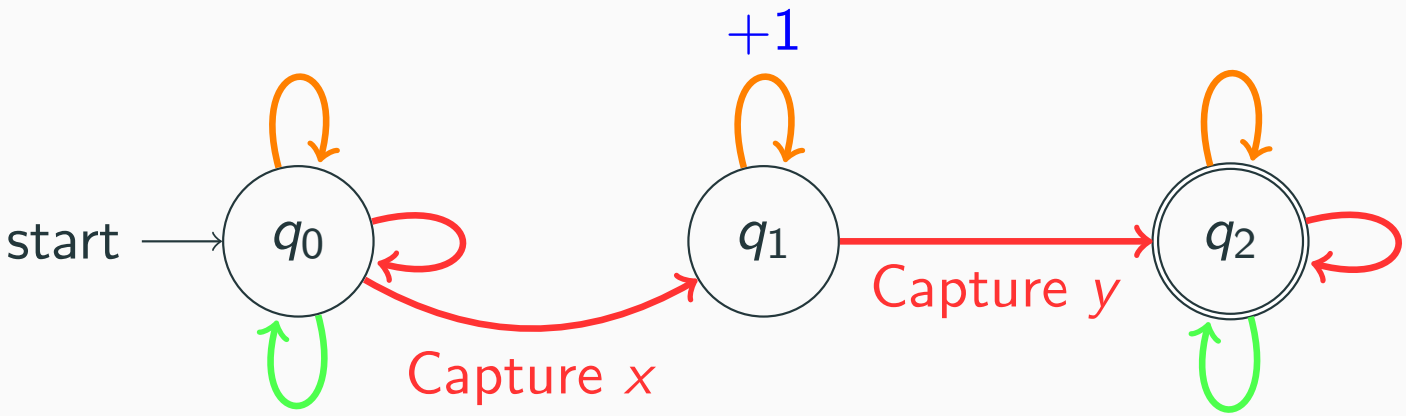
Idea

Each solution comes with a weight

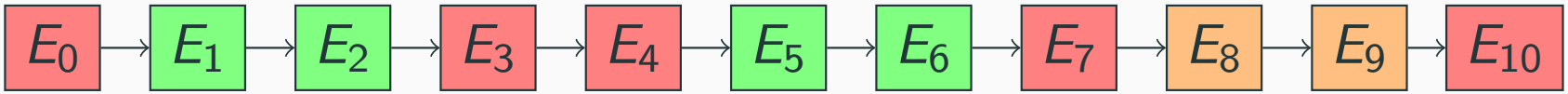
Example

$Weight(x, y) = |\{z \mid x < z < y\}|$

Weighted Automaton with $Weight(x, y) = |\{z \mid x < z < y\}|$



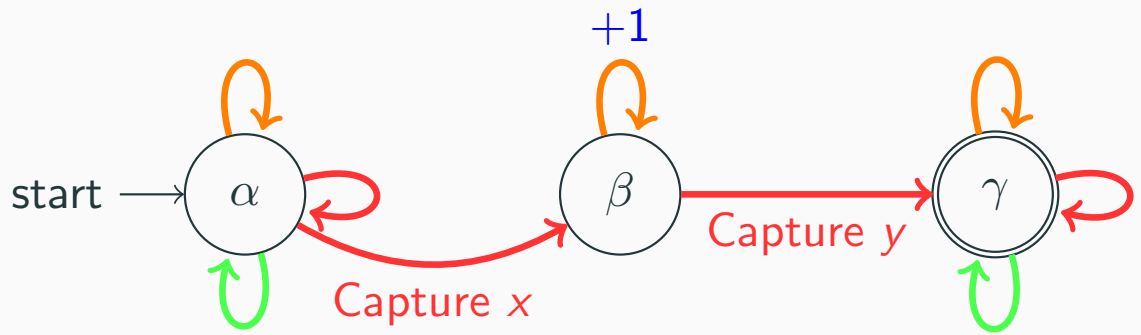
Run and traces

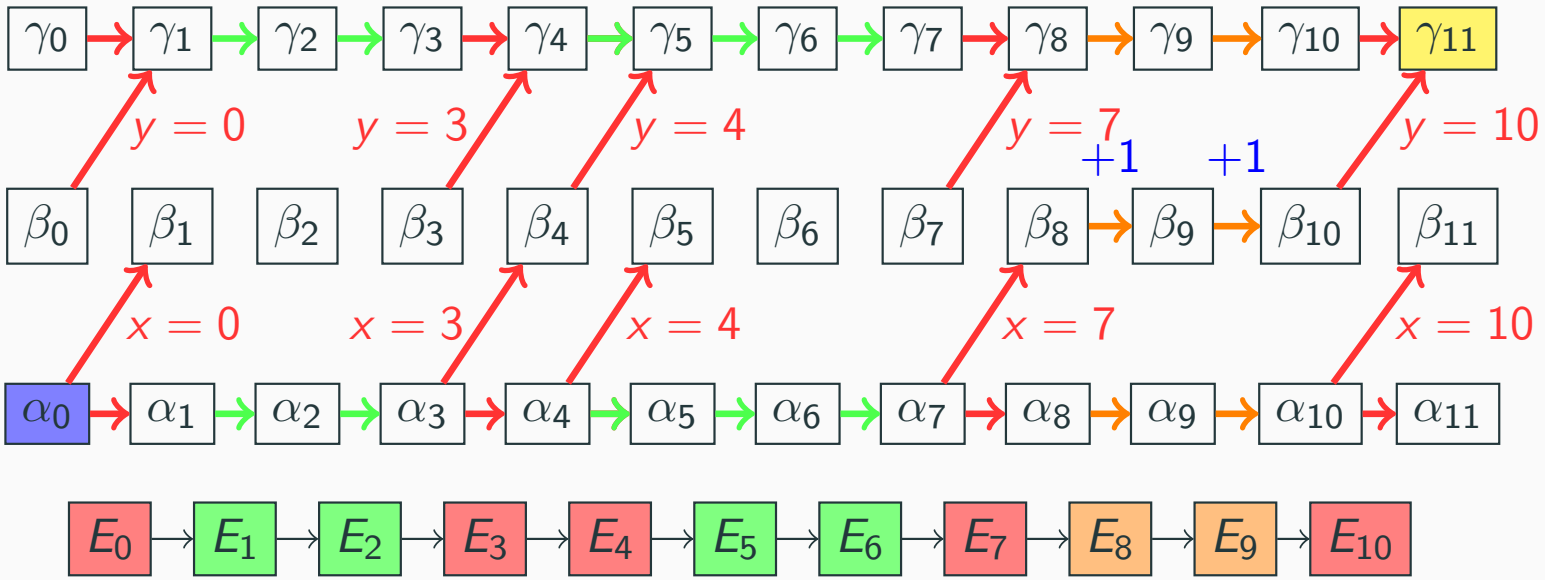


Trace of a run

The trace of a run $q_0 \dots q_n$ has weight $\sum w(q_i, w_i, q_{i+1})$.

Product word - automaton





The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add
- union

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add
- union
- findMin

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add
- union
- findMin
- deleteMin

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add
- union
- findMin
- deleteMin
- increaseBy

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add
- union
- findMin
- deleteMin
- increaseBy
- extendBy

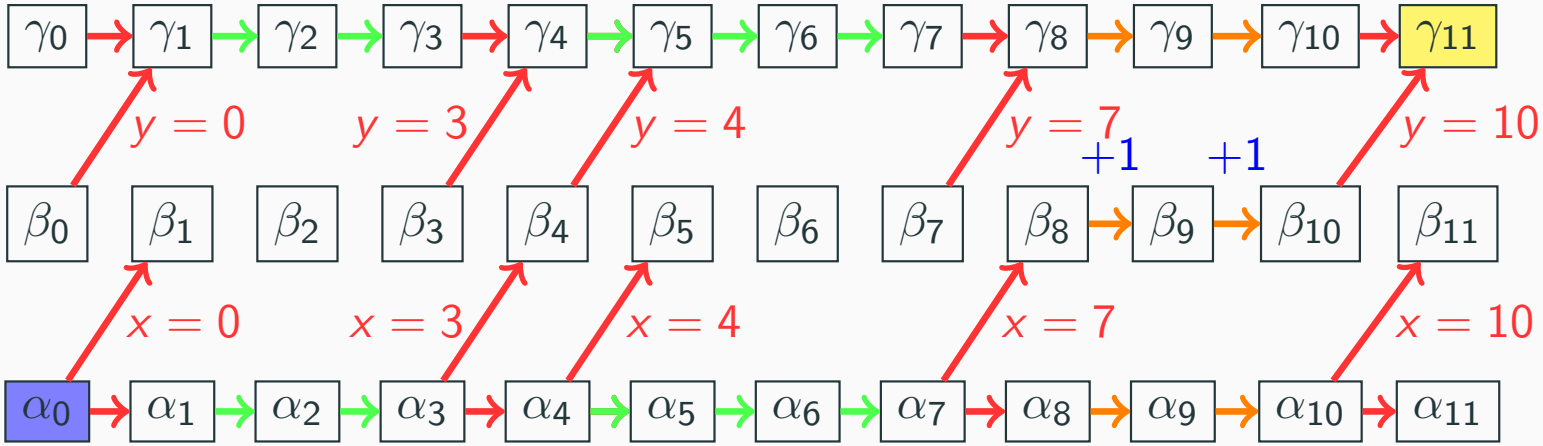
The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add
- union
- findMin
- deleteMin
- increaseBy
- extendBy

We also ask that the datastructure is persistent.

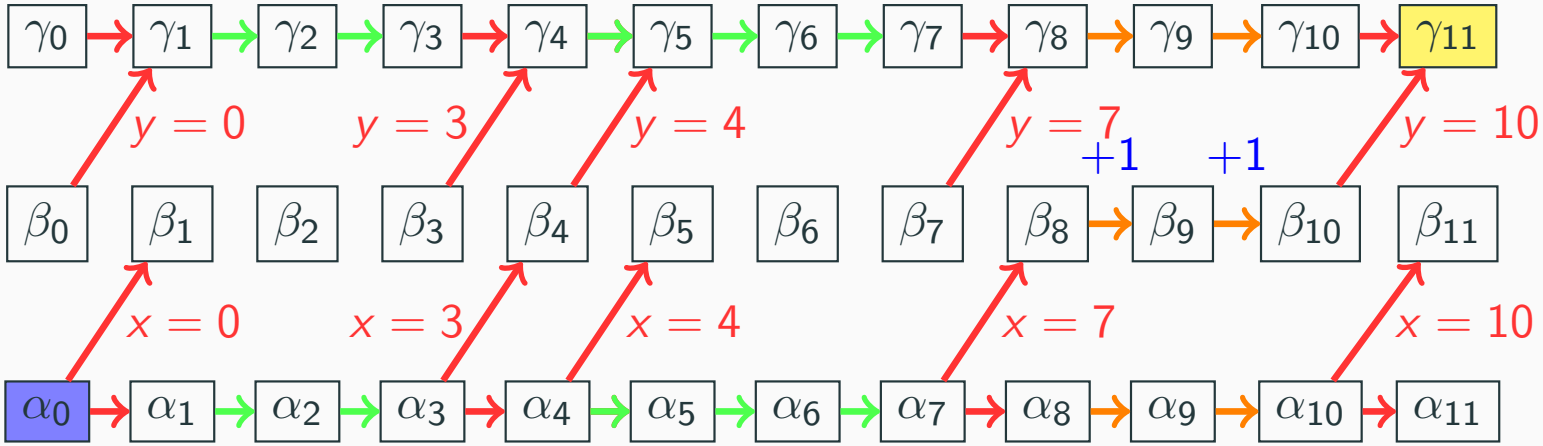
Product word - automaton



For a node n

$\text{Paths}(n) = \text{Union}(\text{Paths}(n, n'))$

Product word - automaton



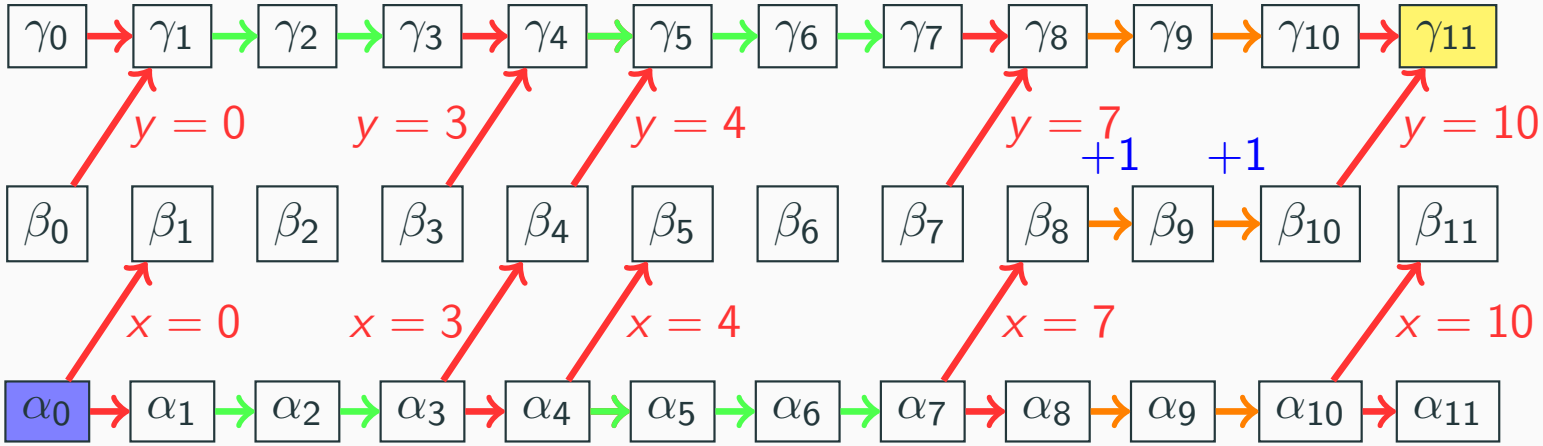
For a node n

$\text{Paths}(n) = \text{Union}(\text{Paths}(n,n'))$

For an edge (u, v) **with a weight** w

$\text{Paths}((u, v)) = \textsc{increaseBy}(\text{Paths}(v),\text{w})$

Product word - automaton



For a node n

$\text{Paths}(n) = \text{Union}(\text{Paths}(n,n'))$

For an edge (u, v) **with a weight** w

$\text{Paths}((u, v)) = \textsc{increaseBy}(\text{Paths}(v),\text{w})$

For an edge (u, v) **with a label** l **and a weight** w

$\text{Paths}((u, v)) = \textsc{increaseBy}(\textsc{extendBy}(\text{Paths}(v),l),w)$

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add
- union

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add
- union
- findMin

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add
- union
- findMin
- deleteMin

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add
- union
- findMin
- deleteMin
- increaseBy

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add
- union
- findMin
- deleteMin
- increaseBy
- extendBy

The Heap of Words datastructure

We suppose that we have a datastructure capable of storing words with priorities equipped with the following operations

- isEmpty
- add
- union
- findMin
- deleteMin
- increaseBy
- extendBy

We also asks that the datastructure is persistent.

Questions?

The Brodal Queue

The Heap of Words datastructure