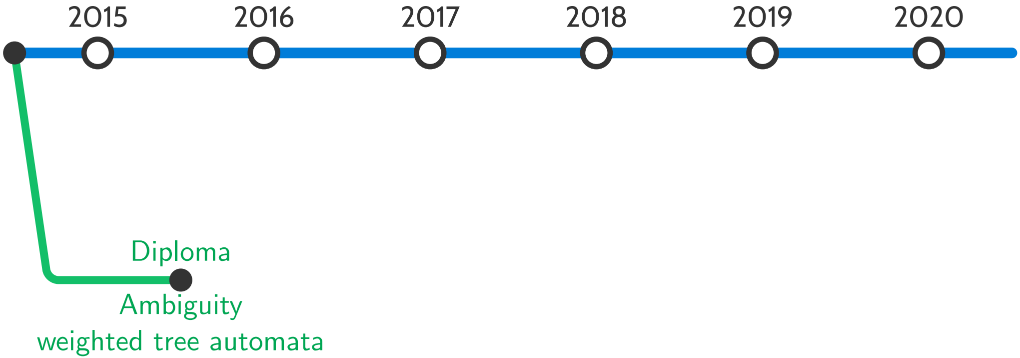
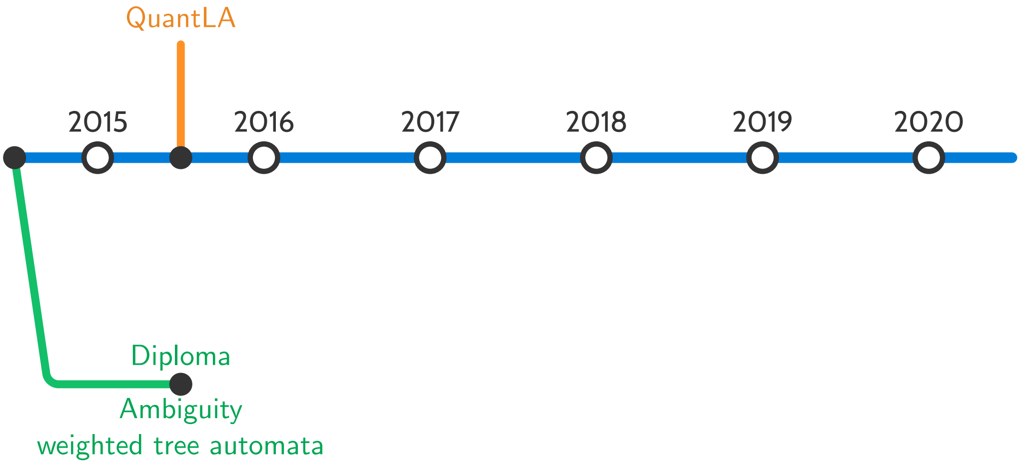
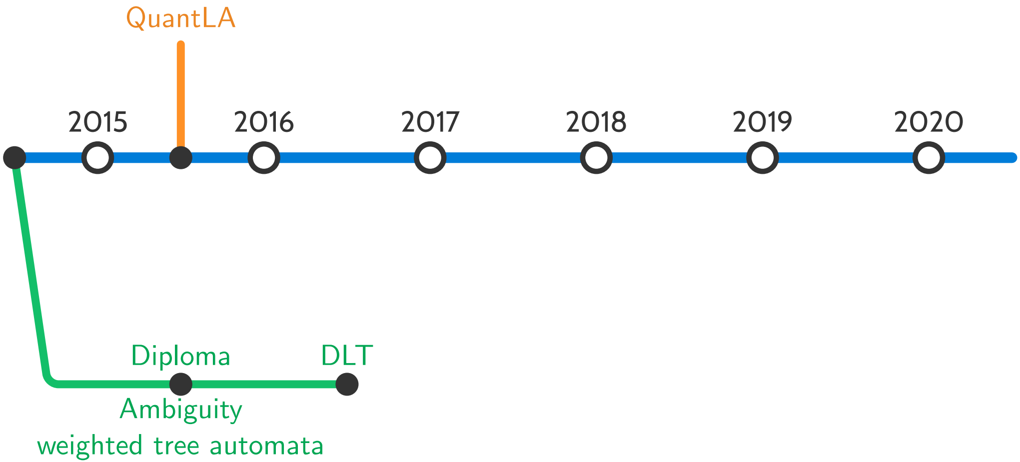
Trees, Logics, and Automata

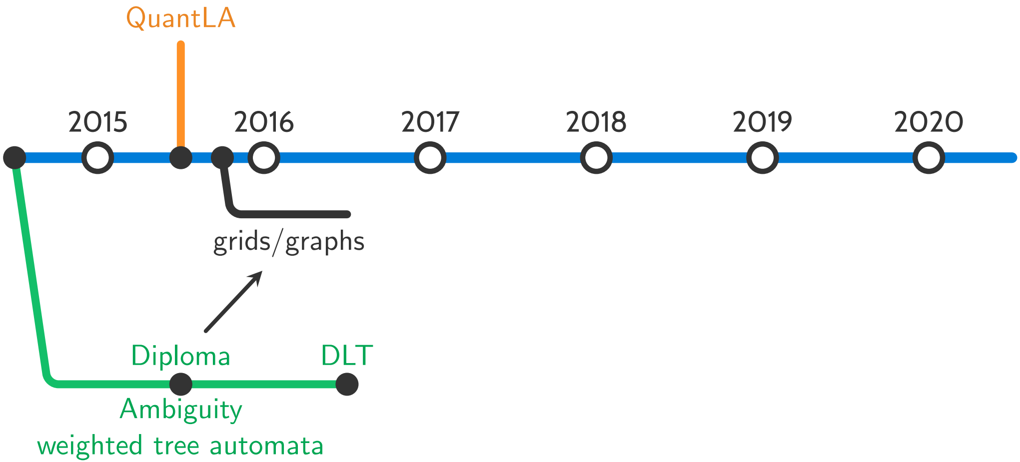
My Time in QuantLA

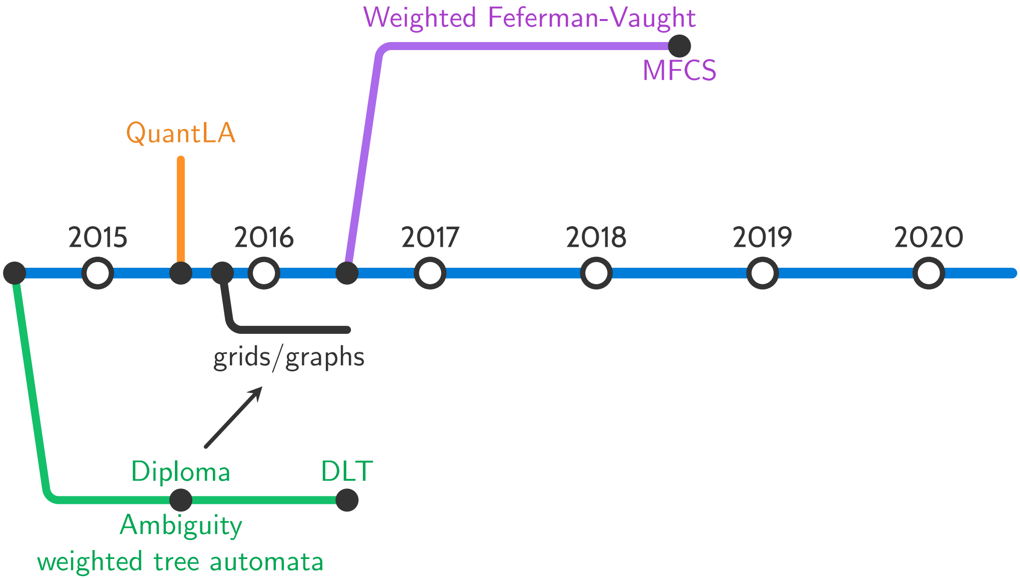
Erik Paul


2015
2016
2017
2018
2019
2020
Diploma
Ambiguity
weighted tree automata


QuantLA
2015
2016
2017
2018
2019
2020
Diploma
Ambiguity
weighted tree automata


QuantLA
2015
2016
2017
2018
2019
2020
Diploma
DLT
Ambiguity
weighted tree automata


QuantLA
2015
2016
2017
2018
2019
2020
grids/graphs
Diploma
DLT
Ambiguity
weighted tree automata



Weighted Feferman-Vaught
MFCS
QuantLA
2015
2016
2017
2018
2019
2020
grids/graphs
Diploma
DLT
Ambiguity
weighted tree automata

formula β $\xleftrightarrow{\text{satisfaction}}$ structure $\mathcal{A}$

formula β $\xleftrightarrow{\text{satisfaction}}$ structure $\mathcal{A}$

Feferman-Vaught theorem

question about union of structures $\mathcal{A} \sqcup \mathcal{B}$

formula β $\xleftrightarrow{\text{satisfaction}}$ structure $\mathcal{A}$

Feferman-Vaught theorem

question about union of structures $\mathcal{A} \sqcup \mathcal{B}$

questions about $\mathcal{A}$

questions about $\mathcal{B}$

formula β $\xleftrightarrow{\text{satisfaction}}$ structure $\mathcal{A}$

Feferman-Vaught theorem

question about union of structures $\mathcal{A} \sqcup \mathcal{B}$

$\Uparrow$

combine answers

$\nearrow$ $\nwarrow$

questions about $\mathcal{A}$ questions about $\mathcal{B}$

formula β $\xleftrightarrow{\text{satisfaction}}$ structure $\mathcal{A}$

Feferman-Vaught theorem

question about union of structures $\mathcal{A} \sqcup \mathcal{B}$

$\Uparrow$

combine answers

$\nearrow$ $\nwarrow$

questions about $\mathcal{A}$ questions about $\mathcal{B}$

---

weighted logic: $\beta \colon \mathcal{A} \mapsto$ weight (Booleans, numbers, semiring elements)

formula β $\xleftrightarrow{\text{satisfaction}}$ structure $\mathcal{A}$

Feferman-Vaught theorem

question about union of structures $\mathcal{A} \sqcup \mathcal{B}$

$\Uparrow$

combine answers

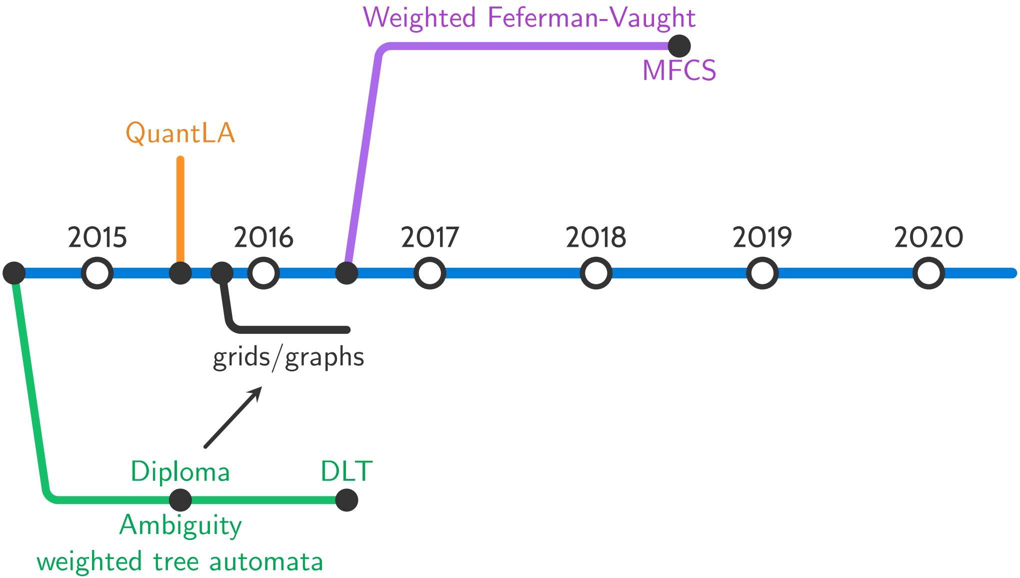
$\nearrow$ $\nwarrow$

questions about $\mathcal{A}$ questions about $\mathcal{B}$

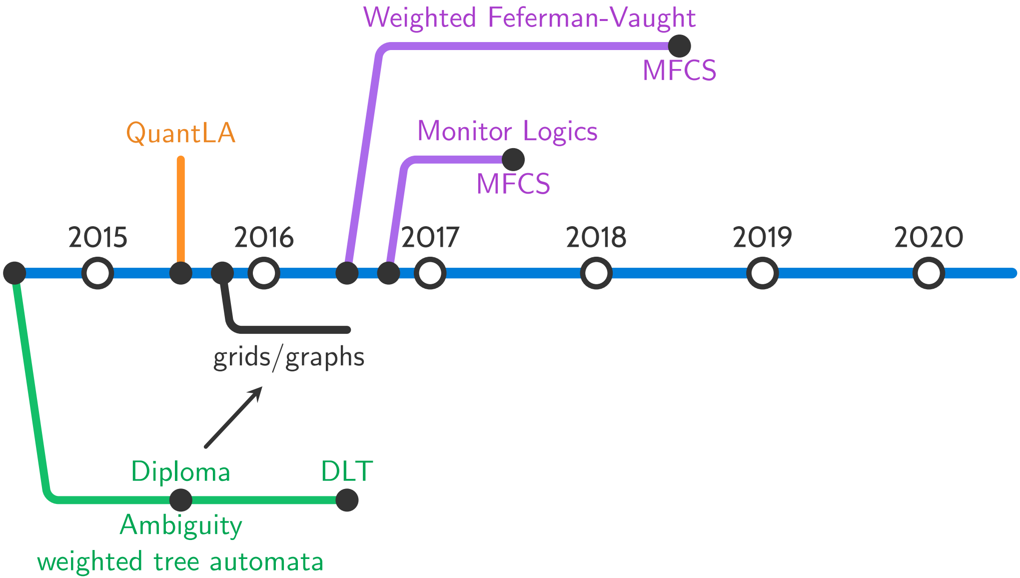
---

weighted logic: $\beta \colon \mathcal{A} \mapsto$ weight (Booleans, numbers, semiring elements)

Example $\bigoplus_x \text{label}_a(x)$ graph $\mapsto$ number of a-nodes



Weighted Feferman-Vaught
MFCS
QuantLA
2015
2016
2017
2018
2019
2020
grids/graphs
Diploma
DLT
Ambiguity
weighted tree automata


Weighted Feferman-Vaught
MFCS
QuantLA
Monitor Logics
MFCS
2015
2016
2017
2018
2019
2020
grids/graphs
Diploma
DLT
Ambiguity
weighted tree automata

Quantitative Monitor Automata [Chatterjee, Henzinger, Otop '16]

- infinite words
- words $\to \mathbb{R}$
- passive ("monitor") counters

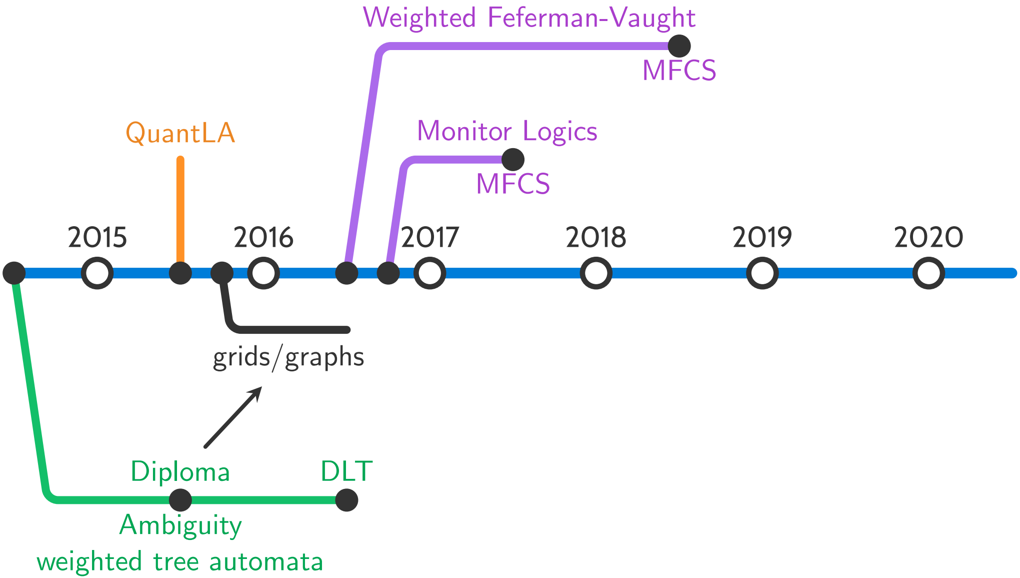
Monitor Logics

Quantitative Monitor Automata [Chatterjee, Henzinger, Otop '16]

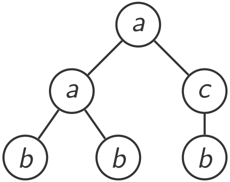
- infinite words
- words $\rightarrow \mathbb{R}$
- passive ("monitor") counters

Logical characterization

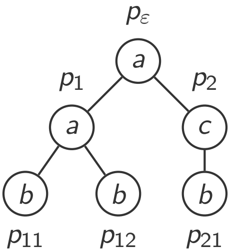
- Quantitative Monitor Automata $\leftrightarrow$ Monitor Logic

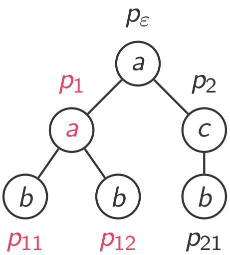

Weighted Feferman-Vaught
MFCS
QuantLA
Monitor Logics
MFCS
2015
2016
2017
2018
2019
2020
grids/graphs
Diploma
DLT
Ambiguity
weighted tree automata

Max-Plus Tree Automata

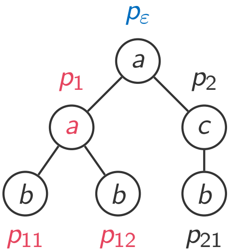


Max-Plus Tree Automata





$$(p_{11}, p_{12}, a, p_1)$$

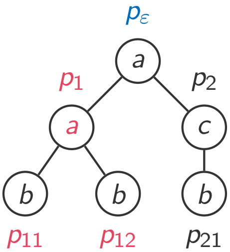


final weights

transition weights

in $\mathbb{R} \cup \{-\infty\}$

(p_{11}, p_{12}, a, p_1)



final weights

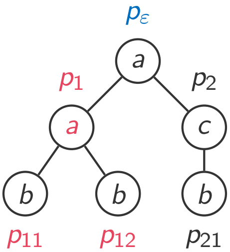
transition weights

in $\mathbb{R} \cup \{-\infty\}$

(p_{11}, p_{12}, a, p_1)

weight of run $=$

transition weights $+$ final weight

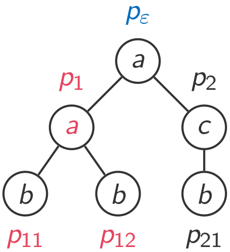


final weights
transition weights
in $\mathbb{R} \cup \{-\infty\}$

(p_{11}, p_{12}, a, p_1)

weight of run =
transition weights + final weight

weight of tree =
maximum over runs on tree



final weights
transition weights
in $\mathbb{R} \cup \{-\infty\}$

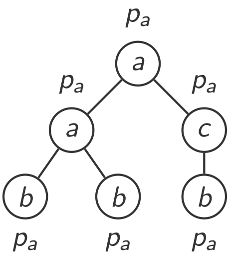
(p_{11}, p_{12}, a, p_1)

weight of run $=$
transition weights $+$ final weight

weight of tree $=$
maximum over runs on tree

Example tree $\mapsto$ $\max\{a\text{-nodes},\ b\text{-nodes}\}$

Max-Plus Tree Automata



final weights
transition weights
in $\mathbb{R} \cup \{-\infty\}$

(p_{11}, p_{12}, a, p_1)

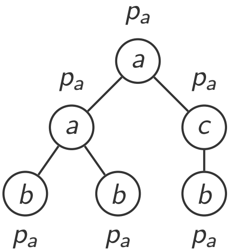
weight of run $=$
transition weights $+$ final weight

weight of tree $=$
maximum over runs on tree

Example

tree $\mapsto$ $\max\{a\text{-nodes},\ b\text{-nodes}\}$
states $\{p_a, p_b\}$

run valid $\leftrightarrow$ only p_a or only p_b



final weights
transition weights
in $\mathbb{R} \cup \{-\infty\}$

(p_{11}, p_{12}, a, p_1)

weight of run $=$
transition weights $+$ final weight

weight of tree $=$
maximum over runs on tree

Example

tree $\mapsto$ max$\{a\text{-nodes},\ b\text{-nodes}\}$
states $\{p_a, p_b\}$

run valid $\leftrightarrow$ only p_a or only p_b

$(p_a, p_a, a, p_a) \mapsto 1$ $\quad$ $(b, p_b) \mapsto 1$ $\quad$ rest $\mapsto \{0, -\infty\}$

Ambiguity

sequential / deterministic

bottom-up determinism

$(p_1, \ldots, p_m, a, q)$

Ambiguity

sequential / deterministic

bottom-up determinism

$(p_1, \ldots, p_m, a, q)$

$$\mathrm{Run}(t) = \{\text{Runs } r \text{ on } t \text{ with weight}(r) \neq -\infty\}$$

Ambiguity

sequential / deterministic — bottom-up determinism

$(p_1, \ldots, p_m, a, q)$

$$\text{Run}(t) = \{\text{Runs } r \text{ on } t \text{ with weight}(r) \neq -\infty\}$$

unambiguous — $|\text{Run}(t)| \leq 1$

Ambiguity

sequential / deterministic	bottom-up determinism $(p_1, \dots, p_m, a, q)$

$$\mathrm{Run}(t) = \{\text{Runs } r \text{ on } t \text{ with weight}(r) \neq -\infty\}$$

unambiguous	$\lvert\mathrm{Run}(t)\rvert \leq 1$
finitely ambiguous	$\lvert\mathrm{Run}(t)\rvert \leq M$

sequential / deterministic — bottom-up determinism $(p_1, \ldots, p_m, a, q)$

$$\text{Run}(t) = \{\text{Runs } r \text{ on } t \text{ with weight}(r) \neq -\infty\}$$

unambiguous	$\lvert\text{Run}(t)\rvert \leq 1$
finitely ambiguous	$\lvert\text{Run}(t)\rvert \leq M$
polynomially ambiguous	$\lvert\text{Run}(t)\rvert \leq P(\lvert t\rvert)$

Ambiguity

sequential / deterministic — bottom-up determinism $(p_1, \ldots, p_m, a, q)$

$$\text{Run}(t) = \{\text{Runs } r \text{ on } t \text{ with weight}(r) \neq -\infty\}$$

unambiguous	$\lvert\text{Run}(t)\rvert \leq 1$
finitely ambiguous	$\lvert\text{Run}(t)\rvert \leq M$
polynomially ambiguous	$\lvert\text{Run}(t)\rvert \leq P(\lvert t\rvert)$

Diploma

- finitely ambiguous = union of unambiguous

AMBIGUITY

sequential / deterministic

bottom-up determinism

$(p_1, \dots, p_m, a, q)$

$$\text{Run}(t) = \{\text{Runs } r \text{ on } t \text{ with weight}(r) \neq -\infty\}$$

unambiguous	$\lvert\text{Run}(t)\rvert \leq 1$
finitely ambiguous	$\lvert\text{Run}(t)\rvert \leq M$
polynomially ambiguous	$\lvert\text{Run}(t)\rvert \leq P(\lvert t\rvert)$

Diploma

- finitely ambiguous = union of unambiguous
- structure of polynomially ambiguous tree automata

sequential / deterministic | bottom-up determinism

$(p_1, \ldots, p_m, a, q)$

$$\text{Run}(t) = \{\text{Runs } r \text{ on } t \text{ with weight}(r) \neq -\infty\}$$

unambiguous	$\lvert\text{Run}(t)\rvert \leq 1$
finitely ambiguous	$\lvert\text{Run}(t)\rvert \leq M$
polynomially ambiguous	$\lvert\text{Run}(t)\rvert \leq P(\lvert t\rvert)$

Diploma

- finitely ambiguous = union of unambiguous
- structure of polynomially ambiguous tree automata
- logical characterization of ambiguity

Four Decision Problems

unambiguous	$\lvert\mathrm{Run}(t)\rvert \leq 1$
finitely ambiguous	$\lvert\mathrm{Run}(t)\rvert \leq M$
polynomially ambiguous	$\lvert\mathrm{Run}(t)\rvert \leq P(\lvert t\rvert)$

Equivalence problem

Given $\mathcal{A}_1, \mathcal{A}_2$ — Is $[\![\mathcal{A}_1]\!](t) = [\![\mathcal{A}_2]\!](t)$ for all t?

unambiguous	$\lvert \mathrm{Run}(t) \rvert \leq 1$
finitely ambiguous	$\lvert \mathrm{Run}(t) \rvert \leq M$
polynomially ambiguous	$\lvert \mathrm{Run}(t) \rvert \leq P(\lvert t \rvert)$

Equivalence problem

Given $\mathcal{A}_1, \mathcal{A}_2$ Is $[\![\mathcal{A}_1]\!](t) = [\![\mathcal{A}_2]\!](t)$ for all t?

Unambiguity problem

Given $\mathcal{A}$ Is $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ for some unamb $\mathcal{A}'$?

Four Decision Problems

unambiguous	$\lvert \mathrm{Run}(t) \rvert \leq 1$
finitely ambiguous	$\lvert \mathrm{Run}(t) \rvert \leq M$
polynomially ambiguous	$\lvert \mathrm{Run}(t) \rvert \leq P(\lvert t \rvert)$

Equivalence problem

Given $\mathcal{A}_1, \mathcal{A}_2$ — Is $[\![\mathcal{A}_1]\!](t) = [\![\mathcal{A}_2]\!](t)$ for all t?

Unambiguity problem

Given $\mathcal{A}$ — Is $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ for some unamb $\mathcal{A}'$?

Sequentiality problem

Given $\mathcal{A}$ — Is $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ for some determ $\mathcal{A}'$?

unambiguous	$\lvert\mathrm{Run}(t)\rvert \leq 1$
finitely ambiguous	$\lvert\mathrm{Run}(t)\rvert \leq M$
polynomially ambiguous	$\lvert\mathrm{Run}(t)\rvert \leq P(\lvert t\rvert)$

Equivalence problem

Given $\mathcal{A}_1, \mathcal{A}_2$ — Is $[\![\mathcal{A}_1]\!](t) = [\![\mathcal{A}_2]\!](t)$ for all t?

Unambiguity problem

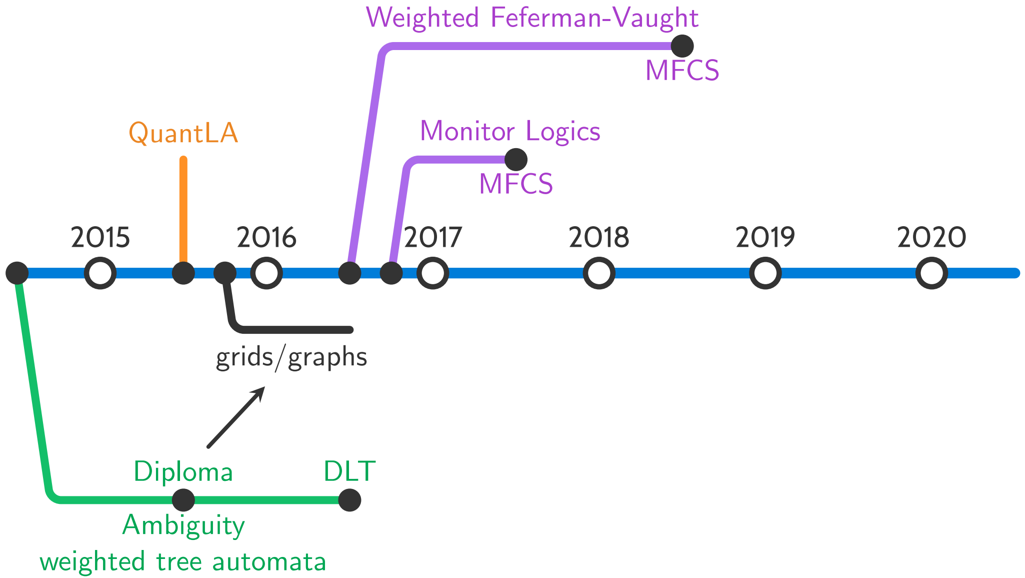
Given $\mathcal{A}$ — Is $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ for some unamb $\mathcal{A}'$?

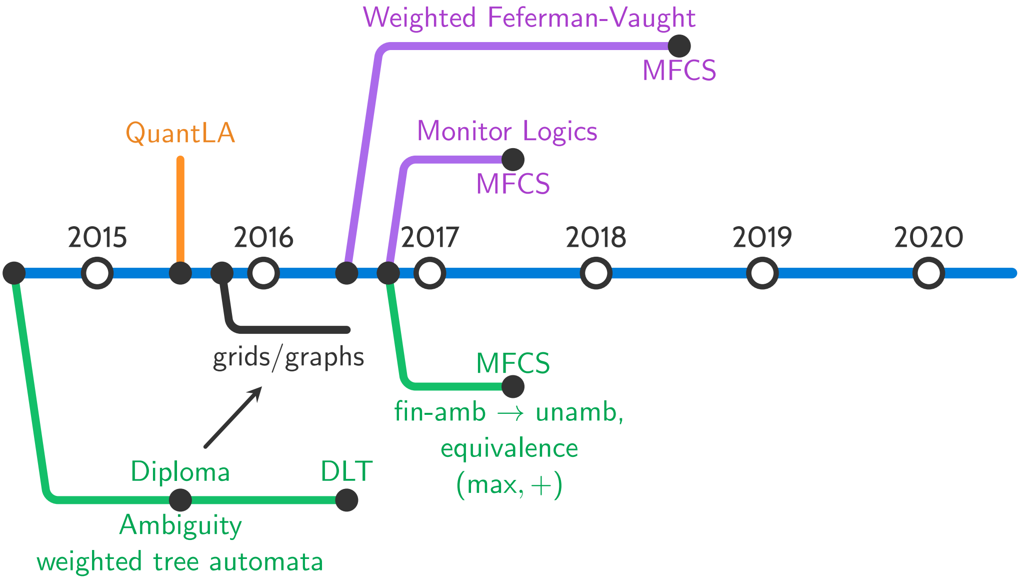
Sequentiality problem

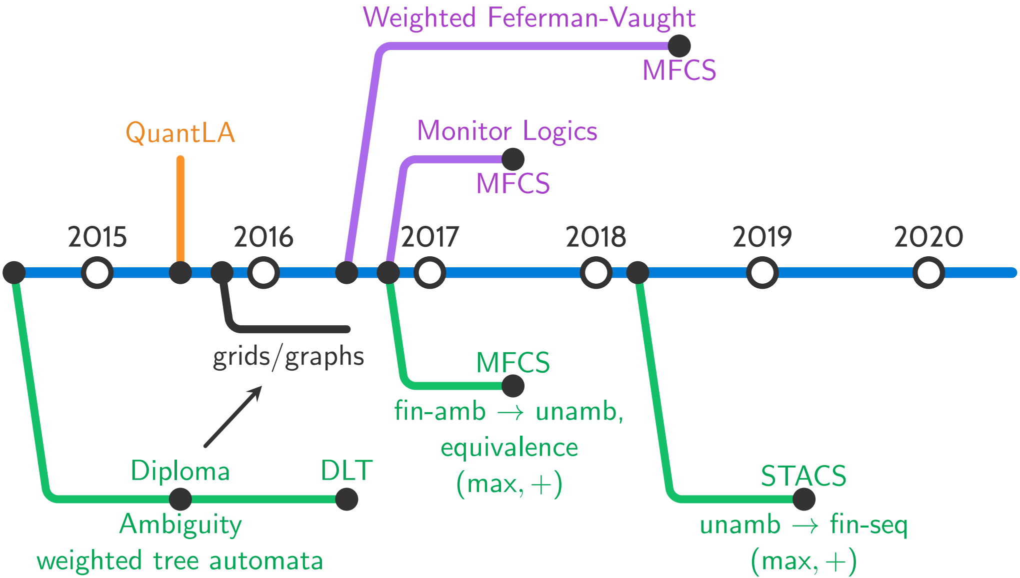
Given $\mathcal{A}$ — Is $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ for some determ $\mathcal{A}'$?

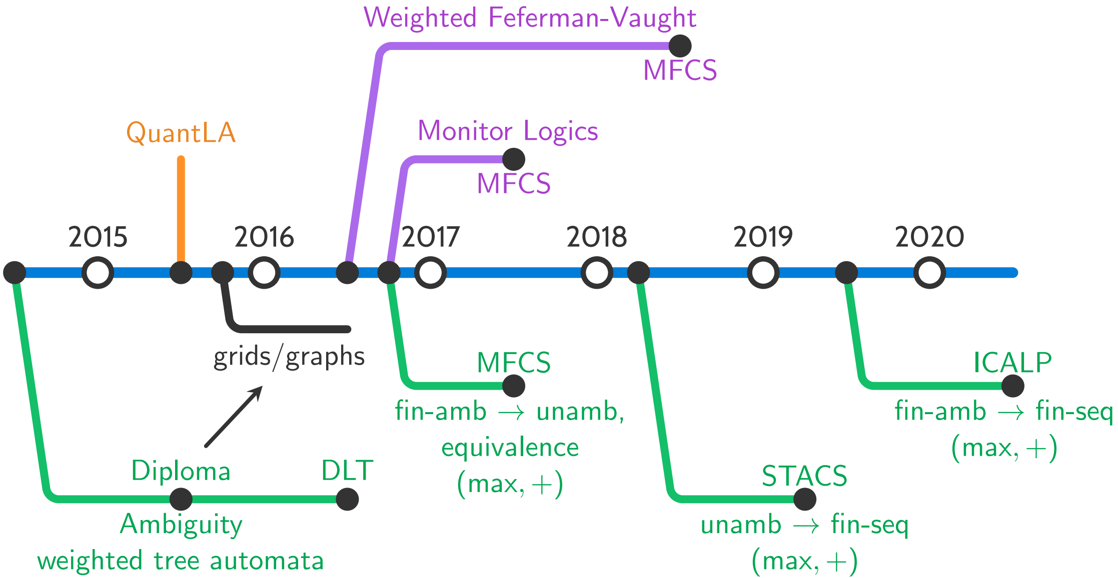
Finite Sequentiality problem

Given $\mathcal{A}$ — Is $[\![\mathcal{A}]\!] = \max_{i=1}^{n} [\![\mathcal{A}_i]\!]$ for some determ $\mathcal{A}_i$?


Weighted Feferman-Vaught
MFCS
QuantLA
Monitor Logics
MFCS
2015
2016
2017
2018
2019
2020
grids/graphs
Diploma
DLT
Ambiguity
weighted tree automata


Weighted Feferman-Vaught
MFCS
QuantLA
Monitor Logics
MFCS
2015
2016
2017
2018
2019
2020
grids/graphs
MFCS
fin-amb → unamb,
equivalence
(max, +)
Diploma
DLT
Ambiguity
weighted tree automata


Weighted Feferman-Vaught
MFCS
Monitor Logics
MFCS
QuantLA
2015
2016
2017
2018
2019
2020
grids/graphs
MFCS
fin-amb → unamb,
equivalence
(max, +)
Diploma
DLT
Ambiguity
weighted tree automata
STACS
unamb → fin-seq
(max, +)


Weighted Feferman-Vaught
MFCS
QuantLA
Monitor Logics
MFCS
2015
2016
2017
2018
2019
2020
grids/graphs
MFCS
fin-amb → unamb,
equivalence
(max, +)
ICALP
fin-amb → fin-seq
(max, +)
Diploma
DLT
Ambiguity
weighted tree automata
STACS
unamb → fin-seq
(max, +)

	Equivalence	Unambiguity	Sequentiality	Finite Sequentiality
fin-amb	**yes**	**yes**	**yes**	**yes**
poly-amb	**no**	**?**	**?**	**?**
general	**no**	**?**	**?**	**?**

Equivalence problem

Given $\mathcal{A}_1, \mathcal{A}_2$ — Is $[\![\mathcal{A}_1]\!](t) = [\![\mathcal{A}_2]\!](t)$ for all t?

Unambiguity problem

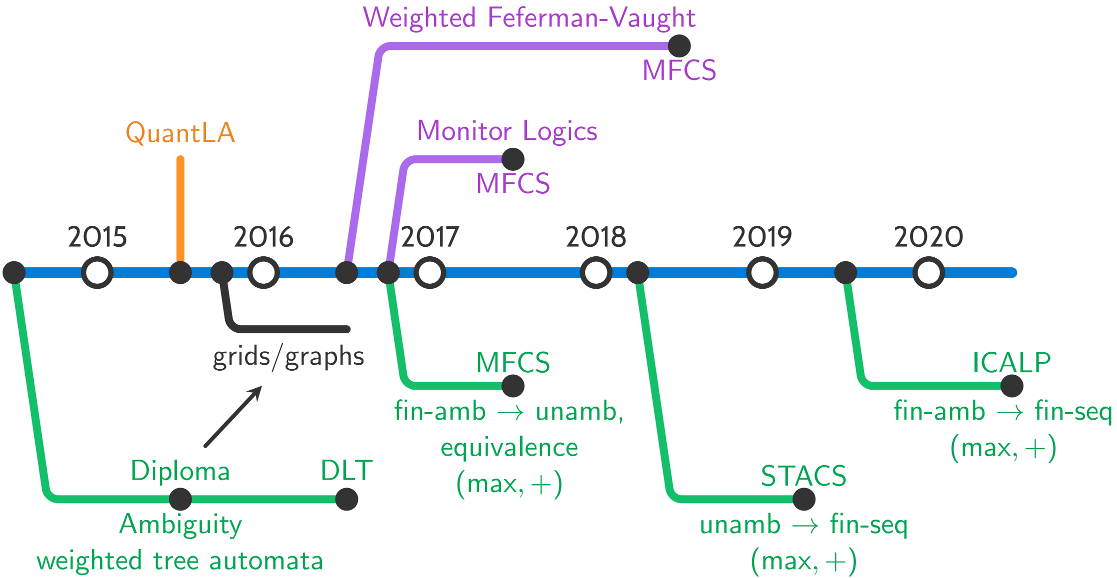
Given $\mathcal{A}$ — Is $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ for some unamb $\mathcal{A}'$?

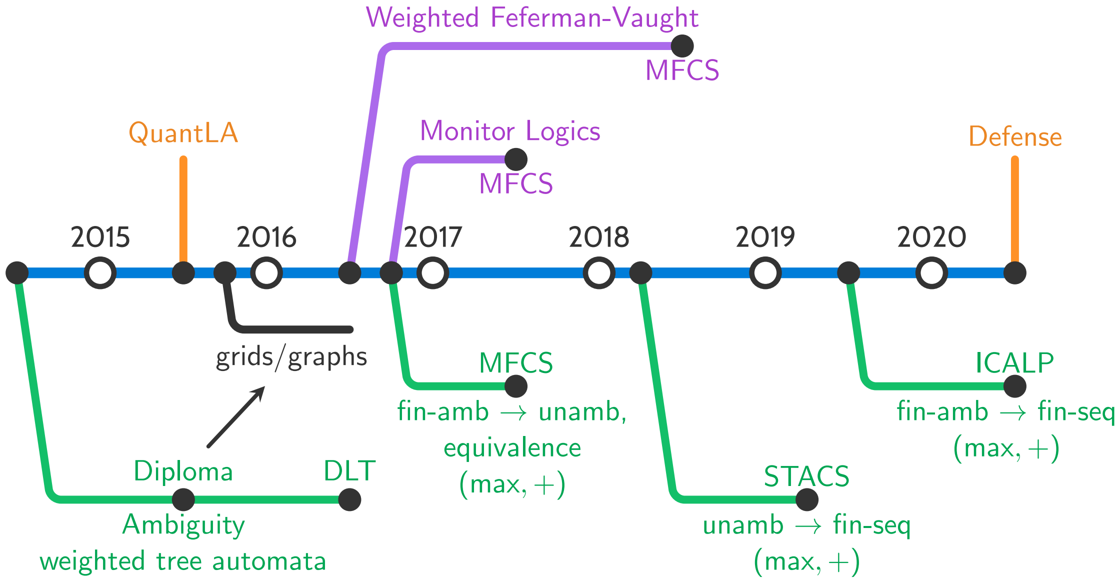
Sequentiality problem

Given $\mathcal{A}$ — Is $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ for some determ $\mathcal{A}'$?

Finite Sequentiality problem

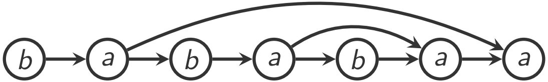
Given $\mathcal{A}$ — Is $[\![\mathcal{A}]\!] = \max_{i=1}^{n} [\![\mathcal{A}_i]\!]$ for some determ $\mathcal{A}_i$?


Weighted Feferman-Vaught
MFCS
QuantLA
Monitor Logics
MFCS
2015
2016
2017
2018
2019
2020
grids/graphs
MFCS
fin-amb → unamb,
equivalence
(max, +)
ICALP
fin-amb → fin-seq
(max, +)
Diploma
DLT
Ambiguity
weighted tree automata
STACS
unamb → fin-seq
(max, +)

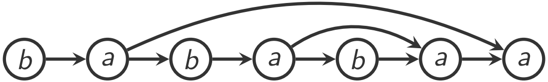

Weighted Feferman-Vaught
MFCS
QuantLA
Monitor Logics
MFCS
Defense
2015
2016
2017
2018
2019
2020
grids/graphs
MFCS
ICALP
fin-amb → unamb,
equivalence
(max, +)
fin-amb → fin-seq
(max, +)
Diploma
DLT
STACS
Ambiguity
weighted tree automata
unamb → fin-seq
(max, +)

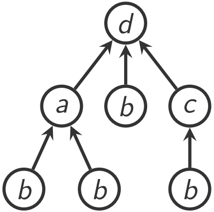

a
b
b
a
a


a
b
b
a
a

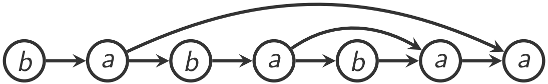

b
a
b
a
b
a
a

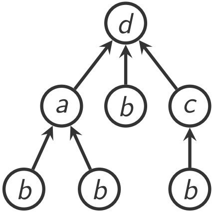

a b b a a

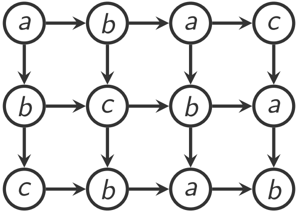

b a b a b a a


d
a b c
b b b

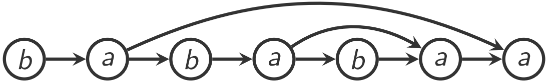

a b b a a

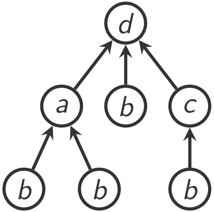

b a b a b a a

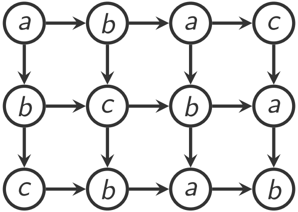

d
a b c
b b b

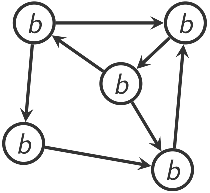

a b a c
b c b a
c b a b

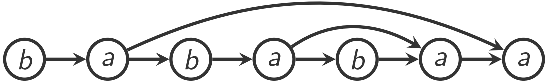

a b b a a

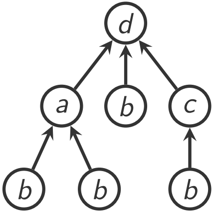

b a b a b a a

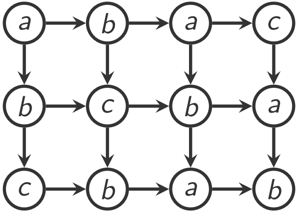

d
a b c
b b b

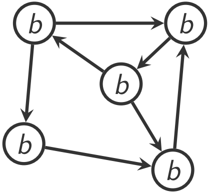

a b a c
b c b a
c b a b


b b
b
b
b

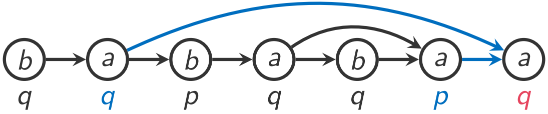


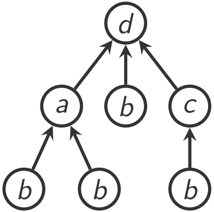


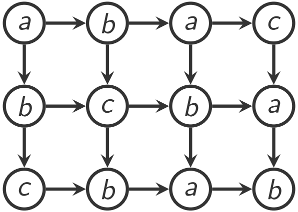


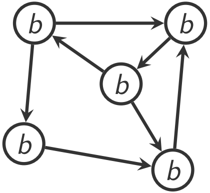


vertex marking automaton: states on nodes $(q_1, \ldots, q_0) \xrightarrow{a} q$

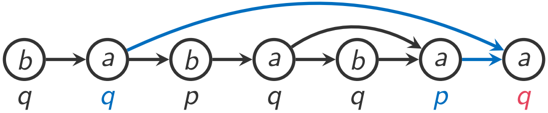


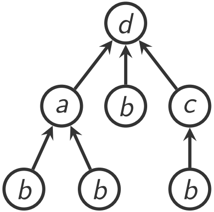


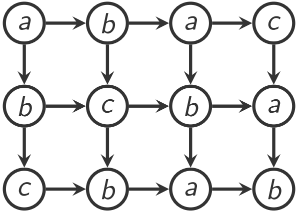


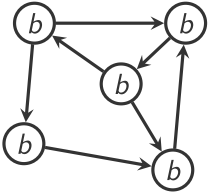


vertex marking automaton: states on nodes $(q_1, \ldots, q_0) \xrightarrow{a} q$



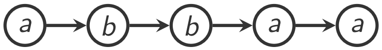


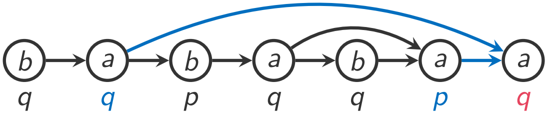


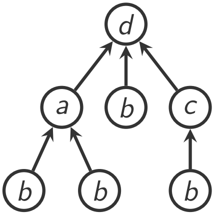


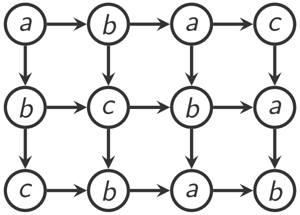
vertex marking automaton: states on nodes $(q_1, \dots, q_0) \xrightarrow{a} q$

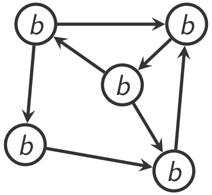
determinize? powerset construction what states could I be in?







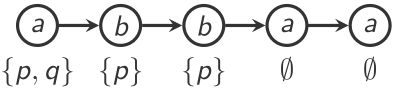


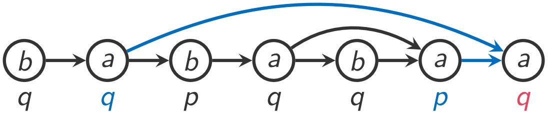


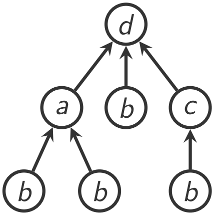
vertex marking automaton: states on nodes $(q_1, \ldots, q_0) \xrightarrow{a} q$

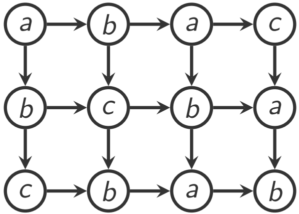
determinize? powerset construction what states could I be in?

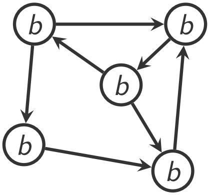
word automaton: $\xrightarrow{a} p$ $\xrightarrow{a} q$ $p \xrightarrow{b} p$







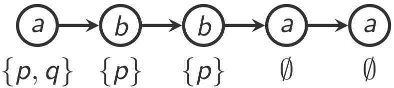


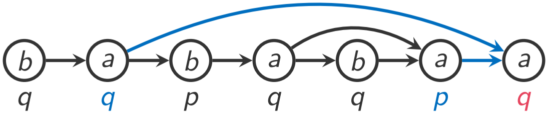


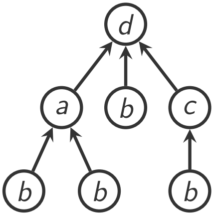
vertex marking automaton: states on nodes $(q_1, \ldots, q_0) \xrightarrow{a} q$

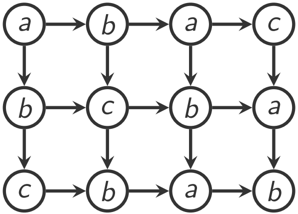
determinize? powerset construction what states could I be in?

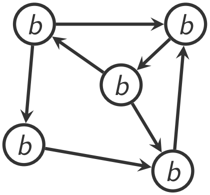
word automaton: $\xrightarrow{a} p$ $\xrightarrow{a} q$ $p \xrightarrow{b} p$









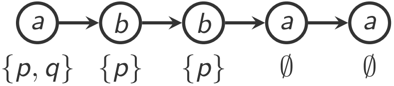


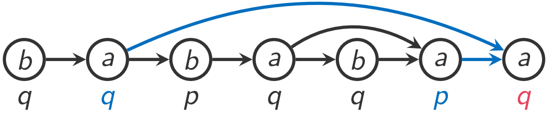
vertex marking automaton: states on nodes $(q_1, \ldots, q_0) \xrightarrow{a} q$

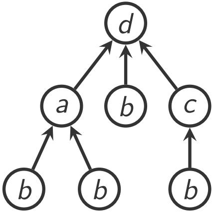
determinize? powerset construction what states could I be in?

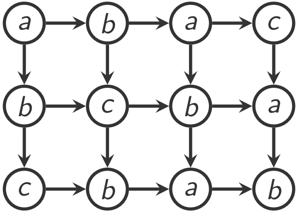
word automaton: $\xrightarrow{a} p$ $\xrightarrow{a} q$ $p \xrightarrow{b} p$

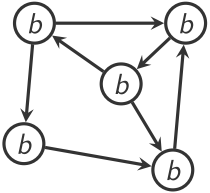
→ fails on top-down trees, nested words, pictures synchronization after split









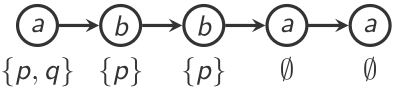


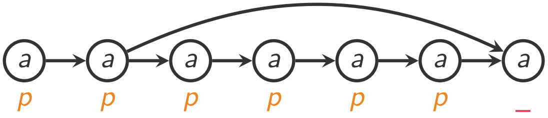
vertex marking automaton: states on nodes $(q_1, \ldots, q_0) \xrightarrow{a} q$

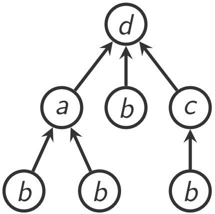
determinize? powerset construction what states could I be in?

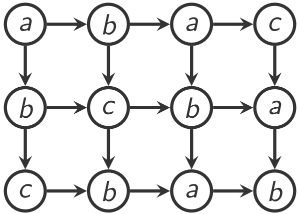
nested word automaton: $\xrightarrow{a} p$ $p \xrightarrow{a} p$ $\xrightarrow{a} q$ $q \xrightarrow{a} q$ $(p, q) \xrightarrow{a} p$

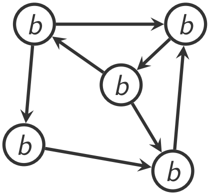
$\rightarrow$ fails on top-down trees, nested words, pictures synchronization after split









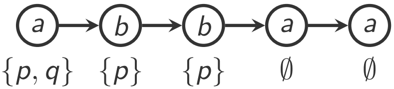


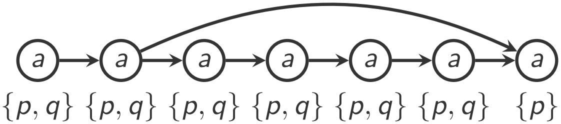
vertex marking automaton: states on nodes $(q_1, \ldots, q_0) \xrightarrow{a} q$

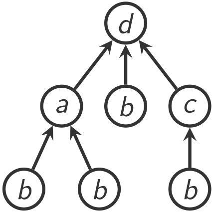
determinize? powerset construction what states could I be in?

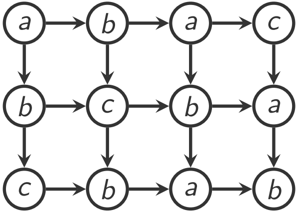
nested word automaton: $\xrightarrow{a} p \quad p \xrightarrow{a} p \quad \xrightarrow{a} q \quad q \xrightarrow{a} q \quad (p, q) \xrightarrow{a} p$

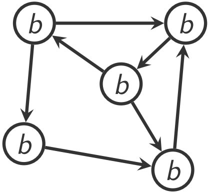
→ fails on top-down trees, nested words, pictures synchronization after split









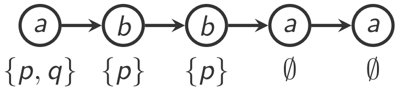


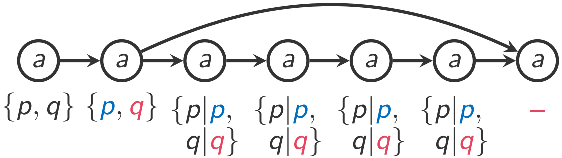
vertex marking automaton: states on nodes $(q_1, \ldots, q_0) \xrightarrow{a} q$

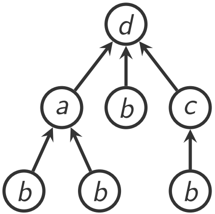
determinize? powerset construction what states could I be in?

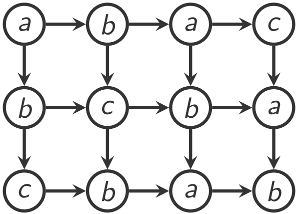
nested word automaton: $\xrightarrow{a} p$ $p \xrightarrow{a} p$ $\xrightarrow{a} q$ $q \xrightarrow{a} q$ $(p, q) \xrightarrow{a} p$

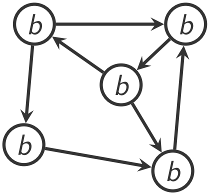
$\rightarrow$ fails on top-down trees, nested words, pictures synchronization after split









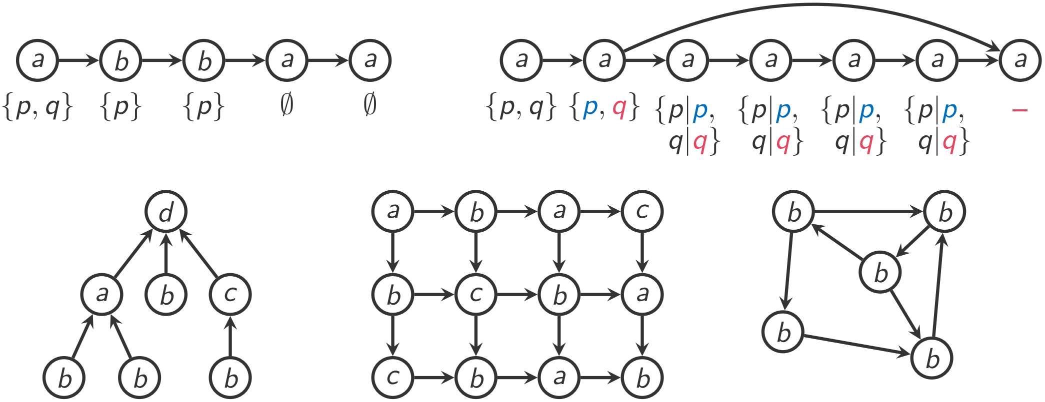


vertex marking automaton: states on nodes $(q_1, \ldots, q_0) \xrightarrow{a} q$

determinize? powerset construction what states could I be in?

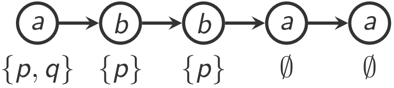
nested word automaton: $\xrightarrow{a} p$ $p \xrightarrow{a} p$ $\xrightarrow{a} q$ $q \xrightarrow{a} q$ $(p, q) \xrightarrow{a} p$

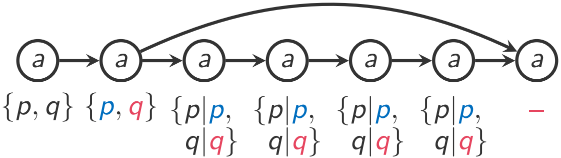
→ fails on top-down trees, nested words, pictures synchronization after split

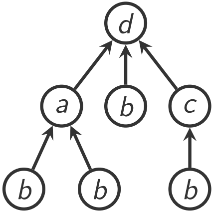


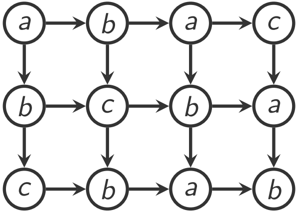
When does modified powerset construction work?

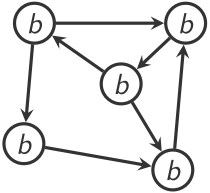
Characterization of determinizable graph classes?











When does modified powerset construction work?

Characterization of determinizable graph classes?

- DAGs with single sink
- Tree width?