Fibrational perspectives on determinization of finite-state automata

Thea Li

Quacs, LMF, Université Paris-Saclay

ACT 2024, 17 June

Purpose and contributions

Purpose

Construct determinization in terms of universal constructions

Objectives

- Carefully investigate the universal property of the determinization construction in terms of simulations between automata.
- Generalize an existing construction to the model of automata presented by Melliès and Zeilberger
- Propose an alternative construction that is "path relevant" and has a stronger universal property

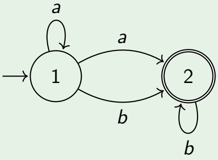
Automata

Definition

A nondeterministic automaton is $M = (\Sigma, Q, \delta : Q \times \Sigma \to \mathcal{P}(Q), q_0, Q_f)$

- Σ is an alphabet
- Q is a set of state
- δ is a transition function
- q_0 and Q_f is a state and a set of states

Example

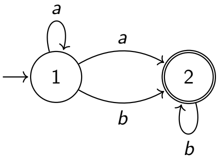


$M = (\{a, b\}, \{1, 2\}, \delta : Q \times \Sigma \to \mathcal{P}(Q), 1, \{2\})$

Automata as functors: an example

Take the automaton on the previous slide, we can see it as a functor from the free monoid over $\{a, b\}$ to **Rel** in the following way:

$$\begin{tikzcd} * \arrow[loop above, "a"] \arrow[loop below, "b"] & \rightarrow & \mathbf{Rel} \end{tikzcd}$$



$$\begin{array}{rrcl}
\text{object(s) :} & * & \mapsto & \{1, 2\} \\
\text{morphisms :} & a & \mapsto & \{(1, 1), (1, 2)\} \\
& b & \mapsto & \{(1, 2), (2, 2)\} \\
& ab & \mapsto & \{(1, 2)\} \\
& \ldots & &
\end{array}$$

Rel-automata

Definition (nondeterministic automata, (Colcombet and Petrişan 2020))

A **Rel**-automaton is a functor $A : B_\Sigma \to \mathbf{Rel}$

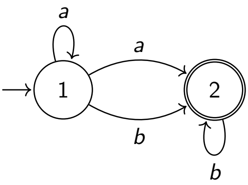
$$
\begin{tikzcd}
B_\Sigma \arrow[d, "A"] & * \arrow[loop above, "a"] \arrow[d, mapsto] \\
\mathbf{Rel} & Q \arrow[loop above, "\delta_a"]
\end{tikzcd}
$$

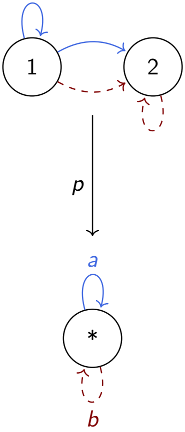
Remark

- *language oriented*
- *models well the logical aspects*
- *it only interprets transitions as ways of relating states under a label*

Example

Take our running example automaton, we can also see it as a functor over the free monoid over $\{a, b\}$ in the following way:





ULF Automata

Definition (ULF-functor)

A functor p is ULF if for each factorization of a morphism $p(\alpha) = uv$ there are unique morphisms β and γ such that $\alpha = \beta\gamma$ and $p(\beta) = u$ and $p(\gamma) = v$.

Definition (ULF-automaton, Melliès and Zeilberger 2022)

An non deterministic automaton over a category B_Σ is a tuple $(B_\Sigma, \mathcal{Q}, p : \mathcal{Q} \to B_\Sigma, q_0, Q_f)$ where:

- p is a ULF functor (with finite fibers)
- q_0 is an object of $\mathcal{Q}$
- Q_f is a set of objects in $\mathcal{Q}$

$$\begin{tikzcd} q_0 \arrow[r, dashed, "\alpha"] \arrow[d, mapsto] & q_f \arrow[d, mapsto] \\ * \arrow[r, "w"'] & * \end{tikzcd} \qquad \begin{tikzcd} Q \arrow[d, "p"] \\ B_\Sigma \end{tikzcd}$$

Span(Set)-automata

Proposition

A ULF functor into $\mathcal{C}$ corresponds by a Grothendieck-like construction to a pseudofunctor $\mathcal{C} \to$ **Span(Set)**.

Proposition (**Span(Set)**-automata, Melliès and Zeilberger 2022)

A ULF-automaton over B_Σ thus correspond (up to choice of initial and final states) to a pseudofunctor $B_\Sigma \to$ **Span(Set)** *(that factors through* **Span(Fin)***).*

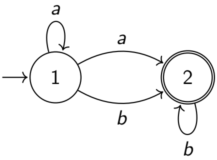
Remark

Rel *is a subcategory of* **Span(Set)**, *consequently the ULF-automata subsumes* **Rel***-automata.*

Automata as functors: an example

Take the automaton on the previous slide, we can see it as a functor from the free monoid over $\{a, b\}$ to **Span(Set)** in the following way:

$$\begin{tikzcd} * \arrow[loop above, "a"] \arrow[loop below, "b"] \arrow[r] & \textbf{Span(Set)} \end{tikzcd}$$



$$\begin{array}{rrcl} \text{object(s)}: & * & \mapsto & \{1, 2\} \\ \text{morphisms}: & a & \mapsto & \{(1,1), (1,2)\} \\ & b & \mapsto & \{(1,2), (2,2)\} \\ & ab & \mapsto & \{\{(1,1), (1,2)\}\{(1,2), (2,2)\}\} \\ & \ldots & & \end{array}$$

Deterministic ULF-automata

Definition (Melliès and Zeilberger 2023)

A deterministic automaton over a category B_Σ is a tuple $(B_\Sigma, \mathcal{Q}, p : \mathcal{Q} \to B_\Sigma, q_0, Q_f)$.

- p is a discrete opfibration (with finite fibers)
- q_0 is an object of $\mathcal{Q}$
- Q_f is a set of objects in $\mathcal{Q}$

$$\begin{tikzcd} p \arrow[r, dashed, "\exists! \ \alpha"] \arrow[d, mapsto] & p' \arrow[d, mapsto] \\ p(q) \arrow[r, "w"'] & * \end{tikzcd} \qquad \begin{tikzcd} Q \arrow[d] \\ B_\Sigma \end{tikzcd}$$

Proposition (**Set**-automata)

*By the Grothendieck construction, functors $B_\Sigma \to$ **Set** are equivalent to discrete opfibrations over B_Σ.*

Simulation

Definition

Given two labeled transition systems, $L_1 = (\Sigma, Q, \delta)$ and $L_2 = (\Sigma, s, \delta')$, a relation $R \subseteq Q \times S$ on the set of states is a *simulation* from L_1 to Ł$_2$ if

$$\forall w \in \Sigma^* \quad \begin{tikzcd} q \arrow[r, "w"] \arrow[d, squiggly, "R"] & q' \\ s & \end{tikzcd} \quad \Longrightarrow \quad \exists s' \quad \begin{tikzcd} q \arrow[r, "w"] \arrow[d, squiggly, "R"] & q' \arrow[d, squiggly, dashed, "R"] \\ s \arrow[r, dashed, "w"'] & s' \end{tikzcd}$$

Definition

Given two **Span(Set)**-automata $F, F' : \mathcal{C} \to$ **Span(Set)**, a *simulation* from F to F' is a lax natural transformation $\alpha : F' \Rightarrow F$.

Simulations

Definition (Simulation)

Let F and F' be **Span(Set)**- automata, given a lax natural transformation $\alpha : F' \Rightarrow F$ then:

$$\begin{tikzcd} F'(*) \arrow[r, "F'(w)"] \arrow[d, "\alpha_*"'] & F'(*) \arrow[d, "\alpha_*"] \\ F(*) \arrow[r, "F(w)"'] \arrow[ur, Rightarrow, "\gamma"] & F(*) \end{tikzcd}$$

Automata translation: for each (q', s) with a lift in $F(w) \circ \alpha_*$

$$\begin{tikzcd} q \arrow[r, "w"] \arrow[d, squiggly, "\alpha"] & q' \\ s & \end{tikzcd} \implies \exists s' \begin{tikzcd} q \arrow[r, "w"] \arrow[d, squiggly, "\alpha"] & q' \arrow[d, squiggly, dashed, "\alpha"] \\ s \arrow[r, dashed, "w"'] & s' \end{tikzcd}$$

with s' given by γ.

Simulations

Definition (Simulation)

Given a pseudo natural transformation $\alpha : F' \Rightarrow F$ then:

$$
\begin{tikzcd}
F'(*) \arrow[r, "F'(w)"] \arrow[d, "\alpha_*"'] & F'(*) \arrow[d, "\alpha_*"] \\
F(*) \arrow[r, "F(w)"'] \arrow[ur, Rightarrow, "\sim"] & F(*)
\end{tikzcd}
$$

Automata translation: for each (q', s) with a lift in $F(w) \circ \alpha_*$

$$
\begin{tikzcd}
 & q' \arrow[d, squiggly, "\alpha"] \\
s \arrow[r, "w"'] & s'
\end{tikzcd}
\implies \exists q
\begin{tikzcd}
q \arrow[r, dashed, "w"] \arrow[d, squiggly, dashed, "\alpha"] & q' \arrow[d, squiggly, "\alpha"] \\
s \arrow[r, "w"'] & s'
\end{tikzcd}
$$

(and vice-versa as the transformation is strict)

Simulations

Definition (Bisimulation)

Given a lax natural transformation $\alpha : F' \Rightarrow F$, such that $(\alpha)^\dagger : F \Rightarrow F'$ then:

$$\begin{tikzcd}
F'(*) \arrow[r, "F'(w)"] \arrow[d, leftrightarrow, "\alpha_*"'] & F'(*) \arrow[d, leftrightarrow, "\alpha_*"] \\
F(*) \arrow[r, "F(w)"'] & F(*)
\end{tikzcd}$$

Automata translation: for each (s', q) with a lift in $F(w) \circ (\alpha)^\dagger_*$

$$\begin{tikzcd}
q \arrow[d, leftrightarrow, squiggly, "\alpha"] & \\
s \arrow[r, "w"'] & s'
\end{tikzcd}
\implies \exists q'
\begin{tikzcd}
q \arrow[r, dashed, "w"] \arrow[d, leftrightarrow, squiggly, "\alpha"] & q' \arrow[d, leftrightarrow, squiggly, dashed, "\alpha"] \\
s \arrow[r, "w"'] & s'
\end{tikzcd}$$

Determinization

Rel-automaton determinization (Colcombet and Petrișan 2020)

- Postcomposing by a functor from **Rel** into **Set**.
- **Rel** is is the kleisli category of the powerset monad on Set, thus there is a natural choice
- There is a canonical simulation of a **Rel**-automaton by its determinization (counit)
- We get a unique factorization of simulations from a deterministic automaton to a nondeterministic one via the canonical simulation.

$$
\begin{tikzcd}
B_\Sigma \arrow[rr, "F"] \arrow[dr, "G"'] & & \mathbf{Rel} \\
& \mathbf{Set} \arrow[ur, "i"'] \arrow[u, phantom, "\Uparrow\alpha"] &
\end{tikzcd}
=
\begin{tikzcd}
\mathcal{C} \arrow[rr, "F"] \arrow[dr, "G"'] & & \mathbf{Rel} \arrow[rr, equal] \arrow[dr, "\mathbb{P}"] & & \mathbf{Rel} \\
& \mathbf{Set} \arrow[ur, "i"'] \arrow[rr, equal] \arrow[u, phantom, "\Uparrow\alpha"] & \arrow[u, phantom, "\Uparrow\eta"] & \mathbf{Set} \arrow[ur, "i"'] \arrow[u, phantom, "\Uparrow\varepsilon"] &
\end{tikzcd}
$$

Local adjunction between **Span(Set)** and **Rel**

Definition (local adjunction)

Let $\mathcal{C}$ and $\mathcal{D}$ be bicategories, a local adjunction between $\mathcal{C}$ and $\mathcal{D}$ is comprised of two functors $L : \mathcal{C} \to \mathcal{D}$ and $R : \mathcal{D} \to \mathcal{C}$, inducing a family of adjunctions

$$
\begin{tikzcd}
\mathcal{D}(LA, B) \arrow[rr, bend left] \arrow[rr, phantom, "\perp"] & & \mathcal{C}(A, RB) \arrow[ll, bend left]
\end{tikzcd}
$$

natural in A and B.

Proposition

There is a local adjunction between **Span(Set)** *and* **Rel**, *induced by* i : **Rel** $\to$ **Span(Set)** *and the "forgetful" functor* im : **Span(Set)** $\to$ **Rel**.

Determinization

Construction

Given a **Span(Set)**-*automaton* $F : B_\Sigma \to \textbf{Span(Set)}$, *its determinization is given by*

$$Det(F) := \; B_\Sigma \xrightarrow{F} \textbf{Span(Set)} \xrightarrow{Im} \textbf{Rel} \xrightarrow{\mathbb{P}} \textbf{Set}$$

Proposition (Canonical simulation)

Let $F : B_\Sigma \to \textbf{Span(Set)}$ *be an automaton*

$$\begin{tikzcd}
\mathcal{C} \arrow[r, "F"] & \textbf{Span(Set)} \arrow[rr, equal] \arrow[dr, "Im"'] & & \textbf{Span(Set)} \arrow[rr, equal] & & \textbf{Span(Set)} \\
& & \textbf{Rel} \arrow[ur, "i"'] \arrow[rr, equal] \arrow[dr, "\mathbb{P}"] \arrow[u, phantom, "\Uparrow \text{``}\eta\text{''}"] & & \textbf{Rel} \arrow[ur, "i"'] & \\
& & & \textbf{Set} \arrow[ur, "i"'] \arrow[u, phantom, "\Uparrow \varepsilon"] & &
\end{tikzcd}$$

Determinization

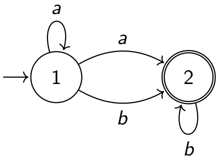
Proposition

Let $G : B_\Sigma \to \mathbf{Set}$ *be some deterministic automaton and* $F : B_\Sigma \to \mathbf{Span(Set)}$, *then there is a factorization:*

$$
\begin{tikzcd}
B_\Sigma \arrow[rrr, "F"] \arrow[ddr, bend right, "G"'] & & & \mathbf{Span(Set)} \\
& & \mathbf{Rel} \arrow[ur, "i"] & \\
& \mathbf{Set} \arrow[ur, "i"] \arrow[uu, phantom, "\Uparrow \alpha"] & &
\end{tikzcd}
\qquad =
$$

$$
\begin{tikzcd}
B_\Sigma \arrow[rrr, "F"] \arrow[ddr, bend right, "G"'] & & & \mathbf{Span(Set)} \arrow[dr, "Im"] \arrow[d, phantom, "\Uparrow \text{"}\varepsilon\text{"}"] \arrow[rr, equal] & \arrow[d, phantom, "\Uparrow \text{"}\eta\text{"}"] & \mathbf{Span(Set)} \arrow[rr, equal] & & \mathbf{Span(Set)} \\
& & \mathbf{Rel} \arrow[ur, "i"] \arrow[rr, equal] \arrow[dr, "\mathbb{P}"] \arrow[d, phantom, "\Uparrow \eta"] & & \mathbf{Rel} \arrow[ur, "i"'] \arrow[rr, equal] \arrow[dr, "\mathbb{P}"] & \arrow[d, phantom, "\Uparrow \varepsilon"] & \mathbf{Rel} \arrow[ur, "i"'] & \\
& \mathbf{Set} \arrow[ur, "i"] \arrow[uu, phantom, "\Uparrow \alpha"] \arrow[rr, equal] & & \mathbf{Set} \arrow[rr, equal] & & \mathbf{Set} \arrow[ur, "i"'] & &
\end{tikzcd}
$$

Example



$$\begin{tikzcd} * \arrow[loop above, "a"] \arrow[loop below, "b"] \end{tikzcd} \quad \to \quad \mathbf{Span(Set)} \quad \to \quad \mathbf{Rel} \quad \to \quad \mathbf{Set}$$

$$\begin{array}{cccccccc}
* & \mapsto & \{1,2\} & \mapsto & \dots & & \mathcal{P}(\{1,2\}) \\
 & & & & & & \\
a & \mapsto & \dots & \mapsto & \delta_a & \mapsto & S \mapsto \{q \mid \exists q' \in S(q \in \delta_a(q'))\} \\
w & \mapsto & \dots & \mapsto & \dots & \mapsto & S \mapsto \{q \mid \exists q' \in S(q \in \delta_{l_n} \dots \delta_{l_1}(q'))\} \\
\dots & & & & & &
\end{array}$$

Classical determinization

Remark

This determinization construction recovers the classical determinization algorithm!

Remark

Let $F : B_\Sigma \to \mathbf{Span(Set)}$ *be some automaton, then the projection* $\Pi : \int \mathbb{P} \circ ImF \to B_\Sigma$ *gives us an automaton with states that are subsets of the states of* F *and transitions labelled* $w : A \to B \in Mor(B_\Sigma)$ *between two states* (A, S) *and* (B, U) *if for* $\mathbb{P} \circ Im(w)(S) = U$*, i.e.* $\{v \in F(B) \mid \exists s \exists \alpha \in F(w)(\alpha : s \to v)\} = V$.

Determinization

Computaional content?

This construction identifies similarily labeled paths between states

Idea

- Path relevance, consider multisets instead of subsets
- A span $A \leftarrow S \rightarrow B$ of finite sets gives a function $A \rightarrow \mathbb{N}^B$

Proposition

Multisets form a relative monad from **Fin** *to* **Set**, *defined for all* A, B : **Fin** *and* $\varphi : A \rightarrow \mathbb{N}^B$ *by*

$$\begin{array}{rcl} M : \mathbf{Fin} & \rightarrow & \mathbf{Set} \\ A & \mapsto & \mathbb{N}^A \end{array} \qquad \begin{array}{rcl} \eta_A : A & \rightarrow & \mathbb{N}^A \\ a & \mapsto & \Sigma_{a' \in A} \chi_a a' \end{array} \qquad \begin{array}{rcl} \varphi^* : \mathbb{N}^A & \rightarrow & \mathbb{N}^B \\ f & \mapsto & \Sigma_{a \in A} f(a) \cdot \varphi(a, -) \end{array}$$

Multiset determinization

Construction

Given a **Span(Fin)***-automaton* $F : B_\Sigma \to \mathbf{Span(Fin)}$*, its multiset determinization is given by the following, where* $\mathbf{Fin}_M$ *is the relative Kleisli-category associated to* M*.*

$$MDet(F) := \; B_\Sigma \xrightarrow{F} \mathbf{Span(Fin)} \xrightarrow{U} \mathbf{Fin}_M \xrightarrow{M} \mathbf{Set}$$

Proposition

Let $F : \mathcal{C} \to \mathbf{Span(Set)}$ *be an automaton, then there is a canonical forward-backward simulation from* F *to* $MDet(F)$*.*

$$\begin{tikzcd}
\mathcal{C} \arrow[r, "F"] & \mathbf{Span(Fin)} \arrow[rr, equal] \arrow[dr, "U"'] & & \mathbf{Span(Fin)} \arrow[rr, "i"] & & \mathbf{Span(Set)} \\
& & \mathbf{Fin}_M \arrow[u, phantom, "\Uparrow \text{''}\eta\text{''}"] \arrow[ur, "\amalg"'] \arrow[rr, "i"] \arrow[dr, "M"'] & & \mathbf{Set}_M \arrow[ur, "i"'] & \\
& & & \mathbf{Set} \arrow[u, phantom, "\Uparrow \beta"] \arrow[ur, "i"'] & &
\end{tikzcd}$$

Universal Property

Proposition

*Let $G : B_\Sigma \to$ **Set** be some deterministic automaton, then any simulation from F to G factors through the canonical forward-backward simulation to MDet(F) and a bisimulation between MDet(F) and G.*

$$
\begin{tikzcd}
B_\Sigma \arrow[rrr, "i\circ F_p"] \arrow[ddr, bend right, "G"'] & & & \mathbf{Span(Set)} \\
& & \mathbf{Fin}_M \arrow[ur, "i\circ\Pi"] & \\
& \mathbf{Fin} \arrow[ur, "i"] \arrow[uu, phantom, "\Uparrow\alpha"] & &
\end{tikzcd}
\qquad = \qquad
$$

$$
\begin{tikzcd}
B_\Sigma \arrow[rrr, "F_p"] \arrow[ddr, bend right, "G"'] \arrow[dr, phantom, "\Uparrow\alpha"] & & & \mathbf{Span(Set)} \arrow[rr, equal] \arrow[dr, "U"] \arrow[rr, phantom, "\Uparrow\text{"}\eta\text{"}"'] & & \mathbf{Span(Fin)} \arrow[rr, "i"] & & \mathbf{Span(Set)} \\
& & \mathbf{Fin}_M \arrow[ur, "\Pi"] \arrow[rr, equal] \arrow[dr, "M"] & & \mathbf{Fin}_M \arrow[ur, "\Pi"'] \arrow[rr, "i"] \arrow[dr, "M"] \arrow[rr, phantom, "\Uparrow\beta"'] & & \mathbf{Set}_M \arrow[ur, "i"'] & \\
& \mathbf{Fin} \arrow[ur, "i"] \arrow[rr, "i"'] \arrow[rr, phantom, "\Uparrow\eta"] & & \mathbf{Set} \arrow[rr, equal] & & \mathbf{Set} \arrow[ur, "i"'] & &
\end{tikzcd}
$$

Conclusions & Takeaways

- Determinization of nondeterministic automata can be expressed as a universal construction
- In particular, we can recover the classical determinization algorithm via such a construction using the powerset monad
  - This relies on a local adjunction between **Span(Set)** and **Rel**
- If one instead would like to have a notion of path relevance, there is a determinization construction using the multiset relative monad.
  - This gives a stronger universal property

Colcombet, Thomas and Daniela Petrişan (Mar. 2020). "Automata Minimization: a Functorial approach". In: *Logical Methods in Computer Science*, pp. 1–32. DOI: 10.23638/LMCS-16(1:32)2020. URL: https://hal.science/hal-03105616.

Melliès, Paul-André and Noam Zeilberger (July 2022). "Parsing as a lifting problem and the Chomsky-Schützenberger representation theorem". In: *MFPS 2022 - 38th conference on Mathematical Foundations for Programming Semantics*. Ithaca, NY, United States. URL: https://hal.science/hal-03702762.

— (Dec. 2023). "The categorical contours of the Chomsky-Schützenberger representation theorem". This is a thoroughly revised and expanded version of a paper with a similar title (hal-03702762, arXiv:2212.09060) presented at the 38th Conference on the Mathematical Foundations of Programming Semantics (MFPS 2022). 62 pages, including a 13 page Addendum on "gCFLs as initial models of gCFGs", and a table of contents. URL: https://hal.science/hal-04399404.