Tutorial: fibrational perspectives on logic and language

Part I: automata over categories

Noam Zeilberger

Ecole Polytechnique (Palaiseau, France)

RAMiCS 2026 @ Poznán, 7 April 2026

What is a logic?

What is a logic?

a <u>set</u> of true formulas $\vdash \phi$

What is a logic?

$$\begin{tikzcd}
\text{a \underline{set} of true formulas} \quad \vdash \phi \arrow[d, "\text{deductive nature of logic}"] \\
\text{a \underline{poset} of entailments} \quad \psi \vdash \phi
\end{tikzcd}$$

What is a logic?

a set of true formulas $\vdash \phi$

deductive nature of logic

a poset of entailments $\psi \vdash \phi$

constructive nature of logic

a category of proofs $\alpha : \psi \vdash \phi$

A tension in categorical logic

The objects of a category have no structure.

A tension in categorical logic

The objects of a category have no structure.

(They are just “points” related by arrows.)

A tension in categorical logic

The objects of a category have no structure.

(They are just “points” related by arrows.)

The formulas of a logic have a lot of structure.

A tension in categorical logic

The objects of a category have no structure.

(They are just “points” related by arrows.)

The formulas of a logic have a lot of structure.

(Proofs are often guided by this structure.)

A tension in categorical logic

The objects of a category have no structure.

(They are just "points" related by arrows.)

The formulas of a logic have a lot of structure.

(Proofs are often guided by this structure.)

How can this structure be made "visible" to the category?

Change of perspective

Idea: don't consider a category in a vacuum, but relative to another "base" category.

Change of perspective

Idea: don't consider a category in a vacuum, but relative to another "base" category.

$$\mathcal{D}$$

Change of perspective

Idea: don't consider a category in a vacuum, but relative to another "base" category.

$$\begin{tikzcd}
\mathcal{D} \arrow[d, "p"] \\
\mathcal{C}
\end{tikzcd}$$

Change of perspective

Idea: don't consider a category in a vacuum, but relative to another "base" category.

$$\begin{tikzcd}
\mathcal{D} \arrow[d, "p"] \\
\mathcal{C}
\end{tikzcd}$$

We can organize ourselves so that the category $\mathcal{D}$ of formulas and proofs always projects down to some simpler category $\mathcal{C}$, as witnessed by the functor p.

Change of perspective

Idea: don't consider a category in a vacuum, but relative to another "base" category.

$$\begin{tikzcd}\mathcal{D} \arrow[d, "p"] \\ \mathcal{C}\end{tikzcd}$$

We can organize ourselves so that the category $\mathcal{D}$ of formulas and proofs always projects down to some simpler category $\mathcal{C}$, as witnessed by the functor p. The objects and arrows of $\mathcal{C}$ will then give us a handle on the objects and arrows of $\mathcal{D}$.

Change of perspective

Idea: don't consider a category in a vacuum, but relative to another "base" category.

$$\begin{tikzcd}\mathcal{D} \arrow[d, "p"] \\ \mathcal{C}\end{tikzcd}$$

We can organize ourselves so that the category $\mathcal{D}$ of formulas and proofs always projects down to some simpler category $\mathcal{C}$, as witnessed by the functor p. The objects and arrows of $\mathcal{C}$ will then give us a handle on the objects and arrows of $\mathcal{D}$. Conversely, the category $\mathcal{D}$ equipped with the functor p may be considered as a "type system" for the objects and arrows of $\mathcal{C}$...

Some terminology and notation

Fix a functor $p : \mathcal{D} \to \mathcal{C}$. Given objects $R \in \mathcal{D}$ and $A \in \mathcal{C}$ such that $p(R) = A$, we write $R \sqsubset A$ and say R **refines** A.

Some terminology and notation

Fix a functor $p : \mathcal{D} \to \mathcal{C}$. Given objects $R \in \mathcal{D}$ and $A \in \mathcal{C}$ such that $p(R) = A$, we write $R \sqsubset A$ and say R **refines** A.

A triple (S, f, T) of an arrow $f : A \to B$ of $\mathcal{C}$ and a pair of objects $S \sqsubset A$ and $T \sqsubset B$ of $\mathcal{D}$ refining its domain and codomain is called a **typing judgment**.

Some terminology and notation

Fix a functor $p : \mathcal{D} \to \mathcal{C}$. Given objects $R \in \mathcal{D}$ and $A \in \mathcal{C}$ such that $p(R) = A$, we write $R \sqsubset A$ and say R **refines** A.

A triple (S, f, T) of an arrow $f : A \to B$ of $\mathcal{C}$ and a pair of objects $S \sqsubset A$ and $T \sqsubset B$ of $\mathcal{D}$ refining its domain and codomain is called a **typing judgment**.

A **derivation** of a typing judgment (S, f, T) is an arrow $\alpha : S \to T$ of $\mathcal{D}$ s.t. $p(\alpha) = f$.

We sometimes write $\alpha : S \Longrightarrow_f T$ to indicate that α is a derivation of (S, f, T).

Functors as type refinement systems: some intuition

$$\begin{tikzcd} \mathcal{D} \arrow[d, "p"'] & & \\ \mathcal{C} & A \arrow[r, "f"] & B \end{tikzcd}$$

(f a program taking input of type A and producing output of type B)

Functors as type refinement systems: some intuition

$$\begin{tikzcd}
\mathcal{D} \arrow[d, "p"'] \\
\mathcal{C}
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
S \arrow[d, dash, dashed] & T \arrow[d, dash, dashed] \\
A \arrow[r, "f"] & B
\end{tikzcd}$$

(S and T predicates refining A and B)

Functors as type refinement systems: some intuition

$$\begin{tikzcd}
\mathcal{D} \arrow[d, "p"'] & S \arrow[r, "\alpha"] \arrow[d, dash, dashed] & T \arrow[d, dash, dashed] \\
\mathcal{C} & A \arrow[r, "f"] & B
\end{tikzcd}$$

(α a proof that f has a more refined type $S \to T$)

Fibrational perspectives on logic and language

This idea may be applied to study a wide variety of deductive systems.

We will consider two main examples:

1. Finite-state automata (discrete (op)fibrations and ULF functors)
2. Sequent calculus (free bifibrations)

Main references:

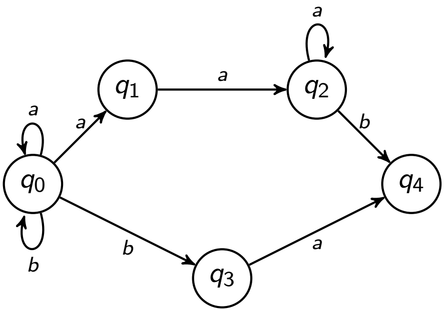
- w/P.-A. Melliès: "The categorical contours of the Chomsky-Schützenberger representation theorem", LMCS 21:2, 2025
- w/B. Clarke & G. Scherer: "The free bifibration on a functor", arxiv:2511.07314

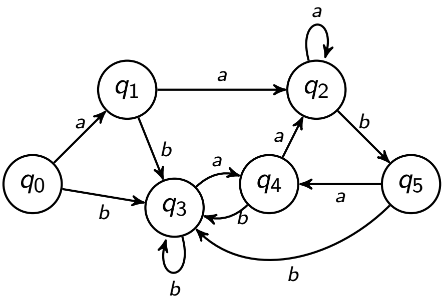
1. Finite-state automata as bundles

Automata as graph homomorphisms

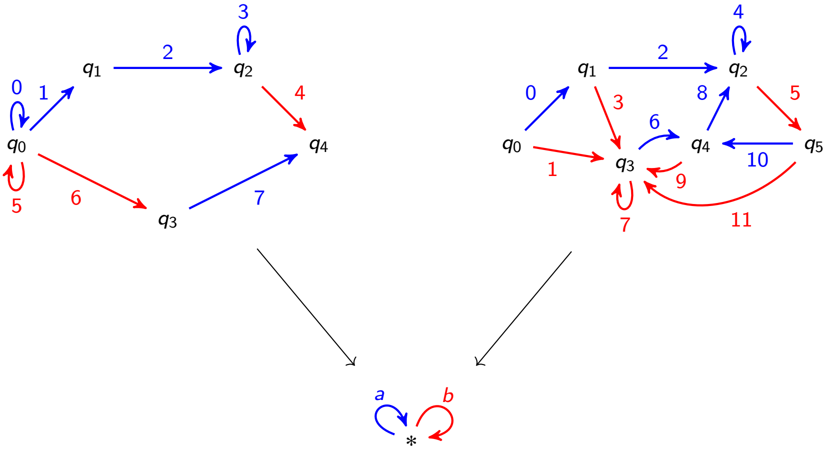
The underlying transition graph of any NFA (without ϵ-transitions) over the alphabet Σ is entirely described by a graph homomorphism $\phi : \mathbb{G} \to \mathbb{B}_\Sigma$ into the *bouquet graph* with one node $*$ and a loop $a : * \to *$ for every $a \in \Sigma$.

Automata as graph homomorphisms





Automata as graph homomorphisms



Free categories

To any graph $\mathbb{G}$ is associated a **free category** $\mathcal{F}\mathbb{G}$ whose objects are nodes and whose arrows are paths. For example, the free category over

$$\begin{tikzcd}
& & B \arrow[dr, "b"] & & & \\
\mathbb{G} \quad = & A \arrow[ur, "a"] \arrow[dr, "c"'] & & D \arrow[r, "e"] & E \arrow[r, bend left, "f"] & F \arrow[l, bend left, "g"] \\
& & C \arrow[ur, "d"'] & & &
\end{tikzcd}$$

has $\hom(A, D) = \{\, ab, cd \,\}$ and $\hom(E, E) = (fg)^*$.

Universal property of free categories: any functor $\mathcal{F}\mathbb{G} \to \mathcal{C}$ into a category $\mathcal{C}$ is uniquely determined by a graph homomorphism $\mathbb{G} \to \mathcal{C}$ into the underlying graph of $\mathcal{C}$.

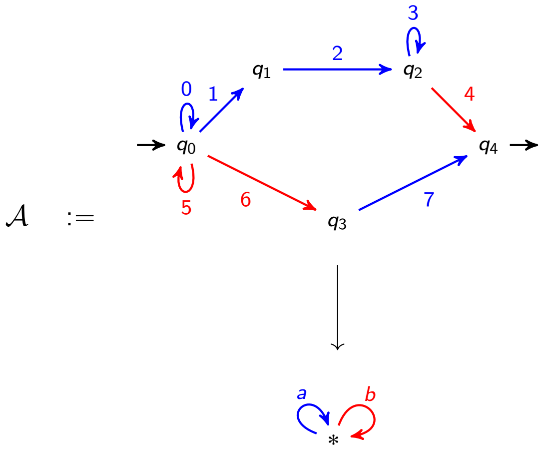
Recognition as a path lifting problem

Any *word* $w \in \Sigma^*$ corresponds to a *path* in $\mathbb{B}_\Sigma$, i.e., to an arrow $* \to *$ in $\mathcal{F}\mathbb{B}_\Sigma$.

Any graph homomorphism $\phi : \mathbb{G} \to \mathbb{H}$ induces a corresponding functor between free categories $\mathcal{F}\phi : \mathcal{F}\mathbb{G} \to \mathcal{F}\mathbb{H}$, sending paths in $\mathbb{G}$ to paths in $\mathbb{H}$ *of the same length*.

Let $\mathcal{A}$ be an NFA with transition graph $\phi : \mathbb{G} \to \mathbb{B}_\Sigma$ and associated functor $p = \mathcal{F}\phi$. Then $\mathcal{A}$ accepts $w \in \Sigma^*$ just in case there is an arrow $\alpha : q_0 \to q_f$ in $\mathcal{F}\mathbb{G}$ such that $p(\alpha) = w$, from an initial state q_0 to an accepting state q_f.

Recognition as a path lifting problem

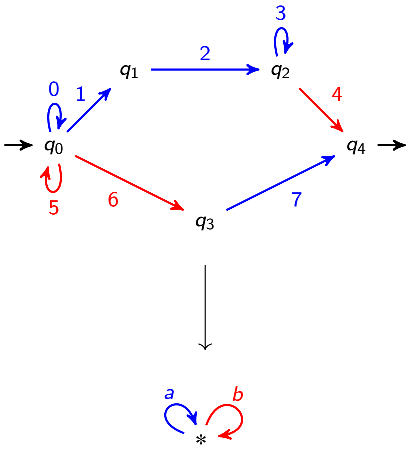


q_0 q_4

$$* \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{b} *$$

Recognition as a path lifting problem

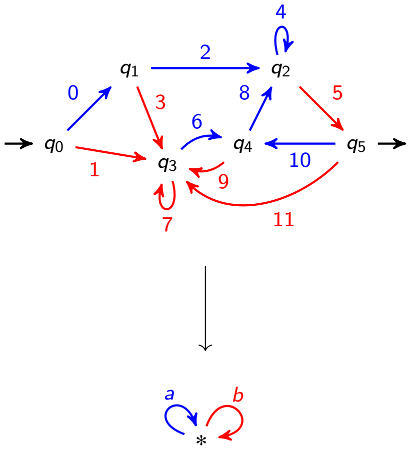
$\mathcal{A} \quad :=$



$$\begin{tikzcd}
 & q_0 \arrow[r, "1"] & q_1 \arrow[r, "2"] & q_2 \arrow[dr, "4"] & \\
q_0 \arrow[ur, "0"] \arrow[dr, "1"'] & & & & q_4 \\
 & q_1 \arrow[r, "2"'] & q_2 \arrow[r, "3"'] & q_2 \arrow[ur, "4"'] & \\
* \arrow[r, "a"] & * \arrow[r, "a"] & * \arrow[r, "a"] & * \arrow[r, "b"] & *
\end{tikzcd}$$

Recognition as a path lifting problem

$\mathcal{A} \quad :=$

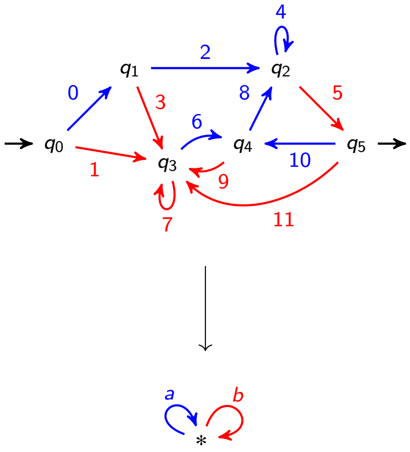


q_0

$$* \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{b} *$$

Recognition as a path lifting problem

$\mathcal{A} \quad :=$

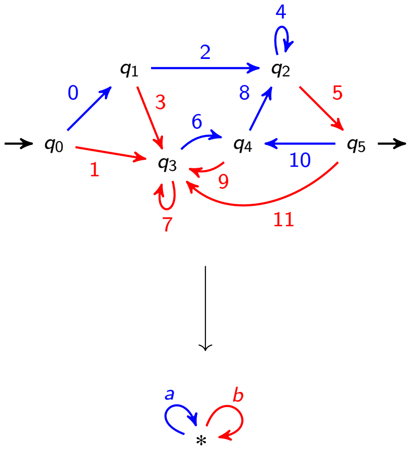


$$q_0 \xrightarrow{0} q_1$$

$$* \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{b} *$$

Recognition as a path lifting problem

$\mathcal{A} \quad :=$

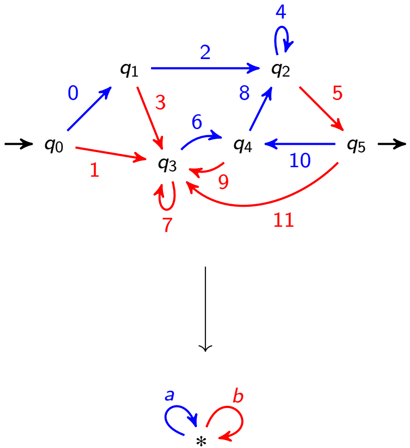


$$q_0 \xrightarrow{0} q_1 \xrightarrow{2} q_2$$

$$* \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{b} *$$

Recognition as a path lifting problem

$\mathcal{A} \quad :=$

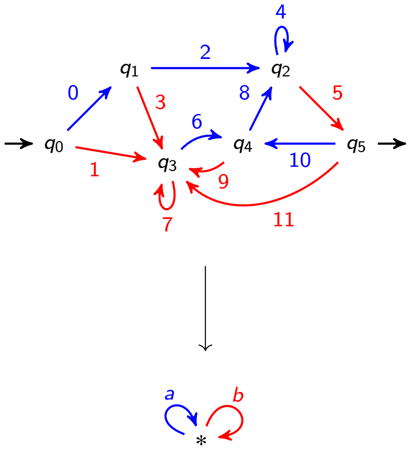


$$q_0 \xrightarrow{0} q_1 \xrightarrow{2} q_2 \xrightarrow{4} q_2$$

$$* \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{b} *$$

Recognition as a path lifting problem

$\mathcal{A} \quad :=$



$$q_0 \xrightarrow{0} q_1 \xrightarrow{2} q_2 \xrightarrow{4} q_2 \xrightarrow{5} q_5$$

$$* \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{b} *$$

Determinism

Proposition: let $\phi : \mathbb{G} \to \mathbb{B}_\Sigma$ be a homomorphism of finite graphs. TFAE:

1. ϕ is the transition graph of a complete DFA.
2. for every node $q \in \mathbb{G}$ and every loop $a : * \to * \in \mathbb{B}_\Sigma$, there is a unique node q' and unique edge $e : q \to q'$ such that $\phi(e) = a$.
3. $\mathcal{F}\phi : \mathcal{F}\mathbb{G} \to \mathcal{F}\mathbb{B}_\Sigma$ is a *discrete opfibration*.

Definition

A functor $p : \mathcal{D} \to \mathcal{C}$ is a **discrete opfibration** if for any object $S \sqsubset A$ of $\mathcal{D}$ and any arrow $f : A \to B$ of $\mathcal{C}$ there exists a unique object $T \sqsubset B$ and arrow $\alpha : S \Longrightarrow_f T$.

$$\begin{tikzcd}
\mathcal{D} \arrow[d, "p"'] & & S & \\
\mathcal{C} & & A \arrow[r, "f"] & B
\end{tikzcd}$$

Definition

A functor $p : \mathcal{D} \to \mathcal{C}$ is a **discrete opfibration** if for any object $S \sqsubset A$ of $\mathcal{D}$ and any arrow $f : A \to B$ of $\mathcal{C}$ there exists a unique object $T \sqsubset B$ and arrow $\alpha : S \Longrightarrow_f T$.

$$\begin{tikzcd}
\mathcal{D} \arrow[d, "p"'] & & S \arrow[r, "\alpha"] & T \\
\mathcal{C} & & A \arrow[r, "f"] & B
\end{tikzcd}$$

Presheaves

A **covariant presheaf** on a category $\mathcal{C}$ is a functor $G : \mathcal{C} \to \mathrm{Set}$.

Explicitly, a covariant presheaf G consists of the following:

- a set G_A for every object A in $\mathcal{C}$;
- a function $G_f : G_A \to G_B$ for every arrow $f : A \to B$ in $\mathcal{C}$;
- such that $G_{fg} = G_f \,; G_g$ and $G_{\mathrm{id}_A} = \mathrm{id}_{G_A}$.

Category of elements

Given a presheaf $G : \mathcal{C} \to \mathrm{Set}$, the **category of elements** $\int G$ is defined like so:

- Objects are pairs (A, R) of an object A in $\mathcal{C}$ and an element $R \in G_A$.
- A morphism $(A, S) \to (B, T)$ is given by a morphism $f : A \to B$ in $\mathcal{C}$ such that the function $G_f : G_A \to G_B$ maps S to T.

This category is equipped with an evident projection functor $\pi_G : \int G \to \mathcal{C}$.

Proposition: $\pi_G : \int G \to \mathcal{C}$ is a discrete opfibration.

Duality of the fibered and indexed perspectives

Conversely, to any discrete opfibration $p : \mathcal{D} \to \mathcal{C}$ is associated a **fiber functor** $G : \mathcal{C} \to \mathrm{Set}$ defined by taking $G_A = \{\, R \in \mathcal{D} \mid R \sqsubset A \,\}$ and letting $G_f : G_A \to G_B$ be the function which maps any $R \sqsubset A$ to the unique $S \sqsubset B$ such that $R \Longrightarrow_f S$.

Duality of the fibered and indexed perspectives

Conversely, to any discrete opfibration $p : \mathcal{D} \to \mathcal{C}$ is associated a **fiber functor** $G : \mathcal{C} \to \mathrm{Set}$ defined by taking $G_A = \{\, R \in \mathcal{D} \mid R \sqsubset A \,\}$ and letting $G_f : G_A \to G_B$ be the function which maps any $R \sqsubset A$ to the unique $S \sqsubset B$ such that $R \Longrightarrow_f S$.

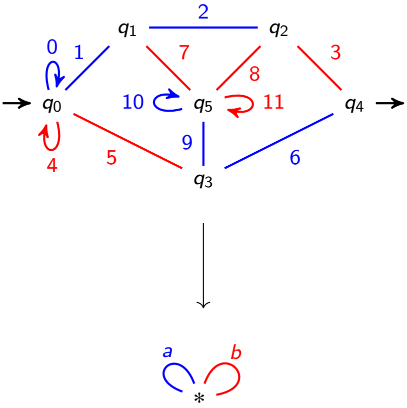
These two constructions

$$\text{discrete opfibration} \underset{\text{category of elements}}{\overset{\text{fiber functor}}{\rightleftarrows}} \text{covariant presheaf}$$

extend to an equivalence of categories between the category $\mathrm{DOpFib}(\mathcal{C})$ of discrete fibrations over $\mathcal{C}$ with commutative triangles as morphisms, and the category $[\mathcal{C}, \mathrm{Set}]$ of covariant presheaves on $\mathcal{C}$ with natural transformations as morphisms.

Running an automaton backwards

$\mathcal{A} \quad :=$

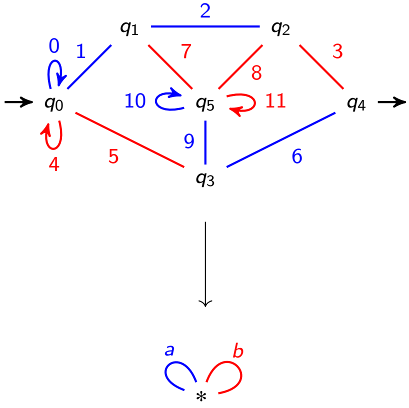


q_4

$$* \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{b} * \xrightarrow{b} *$$

Running an automaton backwards

$\mathcal{A} \quad :=$

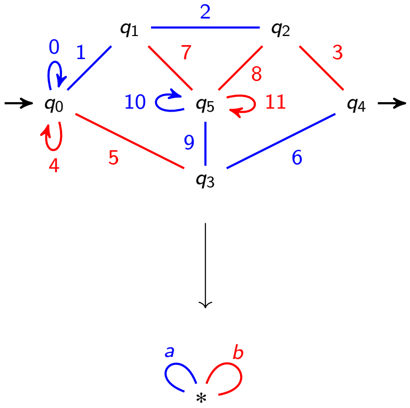


$$q_2 \xrightarrow{3} q_4$$

$$* \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{b} * \xrightarrow{b} *$$

Running an automaton backwards

$\mathcal{A} \quad :=$

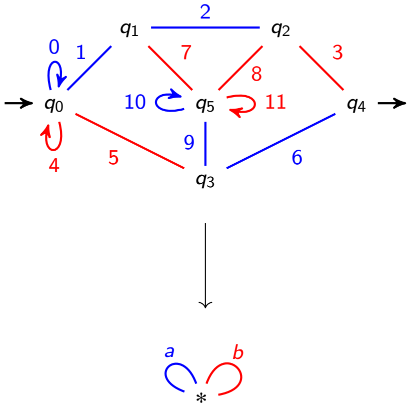


$$q_5 \xrightarrow{8} q_2 \xrightarrow{3} q_4$$

$$* \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{b} * \xrightarrow{b} *$$

Running an automaton backwards

$\mathcal{A} \quad :=$

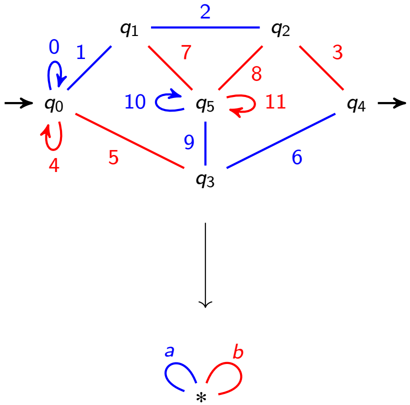


$$q_5 \xrightarrow{10} q_5 \xrightarrow{8} q_2 \xrightarrow{3} q_4$$

$$* \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{b} * \xrightarrow{b} *$$

Running an automaton backwards

$\mathcal{A} \quad :=$



$$q_5 \xrightarrow{10} q_5 \xrightarrow{10} q_5 \xrightarrow{8} q_2 \xrightarrow{3} q_4$$

$$* \xrightarrow{a} * \xrightarrow{a} * \xrightarrow{b} * \xrightarrow{b} *$$

Codeterminism

Proposition: let $\phi : \mathbb{G} \to \mathbb{B}_\Sigma$ be a homomorphism of finite graphs. TFAE:

1. ϕ is the transition graph of a complete codeterministic automaton.
2. for every node $q \in \mathbb{G}$ and every loop $a : * \to * \in \mathbb{B}_\Sigma$, there is a unique node q' and unique edge $e : q' \to q$ such that $\phi(e) = a$.
3. $\mathcal{F}\phi : \mathcal{F}\mathbb{G} \to \mathcal{F}\mathbb{B}_\Sigma$ is a *discrete fibration*.

Definition

A functor $p : \mathcal{D} \to \mathcal{C}$ is a **discrete fibration** if for any object $T \sqsubset B$ of $\mathcal{D}$ and any arrow $f : A \to B$ of $\mathcal{C}$ there exists a unique object $S \sqsubset A$ and arrow $\alpha : S \Longrightarrow_f T$.

$$\begin{tikzcd}
\mathcal{D} \arrow[d, "p"'] \\
\mathcal{C}
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
 & T \\
A \arrow[r, "f"] & B
\end{tikzcd}$$

Definition

A functor $p : \mathcal{D} \to \mathcal{C}$ is a **discrete fibration** if for any object $T \sqsubset B$ of $\mathcal{D}$ and any arrow $f : A \to B$ of $\mathcal{C}$ there exists a unique object $S \sqsubset A$ and arrow $\alpha : S \Longrightarrow_f T$.

$$\begin{tikzcd}
\mathcal{D} \arrow[d, "p"'] \\
\mathcal{C}
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
S \arrow[r, "\alpha"] & T \\
A \arrow[r, "f"] & B
\end{tikzcd}$$

Discrete fibrations as contravariant presheaves

A **contravariant presheaf** on a category $\mathcal{C}$ is a functor $G : \mathcal{C} \to \mathrm{Set}^{\mathrm{op}}$.

Explicitly, a contravariant presheaf G consists of the following:

- a set G_A for every object A in $\mathcal{C}$;
- a function $G_f : G_B \to G_A$ for every arrow $f : A \to B$ in $\mathcal{C}$;
- such that $G_{fg} = G_f \circ G_g$ and $G_{\mathrm{id}_A} = \mathrm{id}_{G_A}$.

Symmetric versions of the category of elements and fiber functor constructions realize an equivalence of categories $\mathrm{DFib}(\mathcal{C}) \cong [\mathcal{C}, \mathrm{Set}^{\mathrm{op}}]^{\mathrm{op}}$.

2. NFAs over categories

A key property of automata

Deterministic and codeterministic automata are represented by discrete opfibrations and discrete fibration, respectively.

A key property of automata

Deterministic and codeterministic automata are represented by discrete opfibrations and discrete fibration, respectively.

But more generally, when does a functor $p : \mathcal{D} \to \mathcal{F}\mathbb{B}_\Sigma$ represent an NFA?

A key property of automata

Deterministic and codeterministic automata are represented by discrete opfibrations and discrete fibration, respectively.

But more generally, when does a functor $p : \mathcal{D} \to \mathcal{F}\mathbb{B}_\Sigma$ represent an NFA?

Proposition. The following are equivalent:

1. $\mathcal{D} = \mathcal{F}\mathbb{G}$ and $p = \mathcal{F}\phi$ where $\phi : \mathbb{G} \to \mathbb{B}_\Sigma$ is the transition graph of a NFA.
2. p has finite fibers and the **unique lifting of factorizations** property.

Unique lifting of factorizations

ULF property: let α be an arrow in $\mathcal{D}$ with image $p(\alpha) = w$ in $\mathcal{C}$; if w factors as uv, then there exist unique β and γ in $\mathcal{D}$ such that $\alpha = \beta\gamma$ and $p(\beta) = u$ and $p(\gamma) = v$.

Unique lifting of factorizations

ULF property: let α be an arrow in $\mathcal{D}$ with image $p(\alpha) = w$ in $\mathcal{C}$; if w factors as uv, then there exist unique β and γ in $\mathcal{D}$ such that $\alpha = \beta\gamma$ and $p(\beta) = u$ and $p(\gamma) = v$.

$$\begin{tikzcd}
\mathcal{D} \arrow[dd, "p"'] & S \arrow[rr, "\alpha"] & & T \\
& A \arrow[rr, "w"] \arrow[dr, "u"'] & & B \\
\mathcal{C} & & X \arrow[ur, "v"'] &
\end{tikzcd}$$

Unique lifting of factorizations

ULF property: let α be an arrow in $\mathcal{D}$ with image $p(\alpha) = w$ in $\mathcal{C}$; if w factors as uv, then there exist unique β and γ in $\mathcal{D}$ such that $\alpha = \beta\gamma$ and $p(\beta) = u$ and $p(\gamma) = v$.

$$\begin{tikzcd}
\mathcal{D} \arrow[d, "p"'] \\
\mathcal{C}
\end{tikzcd}$$

$$\begin{tikzcd}
S \arrow[rr, "\alpha"] \arrow[dr, "\beta"'] & & T \\
& U \arrow[ur, "\gamma"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
A \arrow[rr, "w"] \arrow[dr, "u"'] & & B \\
& X \arrow[ur, "v"'] &
\end{tikzcd}$$

Another equivalent view of ULF functors (aka discrete Conduché fibrations)

Let Span be the bicategory whose objects are sets, whose 1-cells are spans $A \leftarrow X \rightarrow B$, and whose 2-cells are morphisms of spans

$$\begin{tikzcd}
 & X \arrow[r, red] \arrow[dl] \arrow[drr] & Y \arrow[dll] \arrow[dr] & \\
A & & & B
\end{tikzcd}$$

.

Another equivalent view of ULF functors (aka discrete Conduché fibrations)

Let Span be the bicategory whose objects are sets, whose 1-cells are spans $A \leftarrow X \rightarrow B$, and whose 2-cells are morphisms of spans

$$\begin{tikzcd}
 & X \arrow[r, red] \arrow[dl] \arrow[drr] & Y \arrow[dll] \arrow[dr] & \\
A & & & B
\end{tikzcd}$$

.

To any functor whatsoever $p : \mathcal{D} \rightarrow \mathcal{C}$ is associated a *lax fiber functor* $G : \mathcal{C} \rightarrow \mathrm{Span}$ that sends every arrow $f : A \rightarrow B$ of $\mathcal{C}$ to the following span of sets:

$$p^{-1}(A) = \{\, S \mid S \sqsubset A \,\} \longleftarrow p^{-1}(f) = \{\, \alpha \mid \alpha : S \Longrightarrow_f T \,\} \longrightarrow p^{-1}(B) = \{\, T \mid T \sqsubset B \,\}$$

Another equivalent view of ULF functors (aka discrete Conduché fibrations)

Let Span be the bicategory whose objects are sets, whose 1-cells are spans $A \leftarrow X \rightarrow B$, and whose 2-cells are morphisms of spans

$$\begin{tikzcd} & X \arrow[r, red] \arrow[dl] \arrow[drr] & Y \arrow[dll] \arrow[dr] & \\ A & & & B \end{tikzcd}$$

.

To any functor whatsoever $p : \mathcal{D} \to \mathcal{C}$ is associated a *lax fiber functor* $G : \mathcal{C} \to \mathrm{Span}$ that sends every arrow $f : A \to B$ of $\mathcal{C}$ to the following span of sets:

$$p^{-1}(A) = \{\, S \mid S \sqsubset A \,\} \longleftarrow p^{-1}(f) = \{\, \alpha \mid \alpha : S \Longrightarrow_f T \,\} \longrightarrow p^{-1}(B) = \{\, T \mid T \sqsubset B \,\}$$

In general this fiber functor is only a lax functor, equipped with structure maps $\textsc{comp} : G_f \, G_g \Rightarrow G_{fg}$ and $\textsc{id} : \mathrm{id}_{G_A} \Rightarrow G_{\mathrm{id}_A}$ that are not necessarily invertible.

Proposition: p is ULF iff $\textsc{comp}$ and $\textsc{id}$ are invertible, i.e., iff G is a pseudofunctor.

Definition: NFA over a category $\mathcal{C}$

A tuple $\mathcal{A} = (\mathcal{Q}, p, q_0, q_f)$ consisting of

- a category $\mathcal{Q}$
- a ULF functor $p : \mathcal{Q} \to \mathcal{C}$ with finite fibers
- a pair of objects $q_0, q_f \in \mathcal{Q}$

$$\begin{tikzcd}
\mathcal{Q} \arrow[d, "p"] & q_0 & q_f \\
\mathcal{C} & &
\end{tikzcd}$$

Definition: NFA over a category $\mathcal{C}$

A tuple $\mathcal{A} = (\mathcal{Q}, p, q_0, q_f)$ consisting of

- a category $\mathcal{Q}$
- a ULF functor $p : \mathcal{Q} \to \mathcal{C}$ with finite fibers
- a pair of objects $q_0, q_f \in \mathcal{Q}$

$$\begin{tikzcd}
\mathcal{Q} \arrow[d, "p"] & q_0 \arrow[r, dashed, "\alpha"] & q_f \\
\mathcal{C} & A \arrow[r, "w"] & B
\end{tikzcd}$$

Such an automaton recognizes a **regular language of arrows** $\mathcal{L}_{\mathcal{A}}$ defined as the image of the homset $\mathcal{Q}(q_0, q_f)$ along p, that is,

$$\mathcal{L}_{\mathcal{A}} \overset{\text{def}}{=} \{\, p(\alpha) \mid \alpha : q_0 \to q_f \,\} \subseteq \mathcal{C}(A, B)$$

where $p(q_0) = A$, $p(q_f) = B$.

ULF $\Rightarrow$ discrete fibers

Proposition: if $p : \mathcal{D} \to \mathcal{C}$ is ULF then it has discrete fibers.

Proof: Suppose that $\alpha : R \to R'$ were a non-identity arrow such that $p(\alpha) = \mathrm{id}_A$.
Then (id_R, α) and $(\alpha, \mathrm{id}_{R'})$ would be two liftings of the factorization $\mathrm{id}_A = \mathrm{id}_A \mathrm{id}_A$.

Hence these automata have no ϵ-transitions!

Example: automata over free categories

Let $\mathbb{B}_\Sigma^{\triangleright\triangleleft}$ be the graph obtained from $\mathbb{B}_\Sigma$ by adjoining a pair of initial and final nodes:

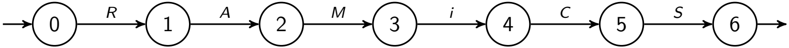
$$\begin{tikzcd} \bot \arrow[r, "\triangleright"] & * \arrow[loop above, "a\in\Sigma"] \arrow[r, "\triangleleft"] & \top \end{tikzcd}$$

An automaton over $\mathcal{F}\mathbb{B}_\Sigma^{\triangleright\triangleleft}$ processes a word with explicit begin/end markers.

More generally, an automaton over the free category $\mathcal{F}\mathbb{H}$ on an arbitrary graph $\mathbb{H}$ recognizes a language of paths, which may be considered as “typed words”.

Example: singleton automaton

For any $w \in \Sigma^*$ s.t. $|w| = n$, there is an $(n+1)$-state automaton recognizing $\{ w \}$:



This is a special case of the following general construction...

Example: singleton automaton

Let $w : A \to B$ be an arrow of a category $\mathcal{C}$. The category Fact_w has objects given by triples $(X,\ A \overset{u}{\to} X \overset{v}{\to} B\ )$ such that $w = uv$, and arrows $(X, u, v) \to (Y, u', v')$ given by arrows $x : X \to Y$ making the diagram commute:

$$\begin{tikzcd}
& X \arrow[drr, "v"] \arrow[ddr, "x"] & & \\
A \arrow[ur, "u"] \arrow[rrr, "w"] \arrow[drr, "u'"'] & & & B \\
& & Y \arrow[ur, "v'"'] &
\end{tikzcd}$$

Let $p_w : \mathrm{Fact}_w \to \mathcal{C}$ be the middle projection $(X, u, v) \mapsto X$. Then p_w is ULF.

Thus $(\mathcal{Q}, p, q_0, q_f) := (\mathrm{Fact}_w, p_w, (\mathrm{id}_A, w), (w, \mathrm{id}_B))$ defines an automaton recognizing the singleton language $\{\, w \,\} \subset \mathcal{C}(A, B)$. (Note this automaton is not always finite state; we say $\mathcal{C}$ has *finitary factorizations* if p_w has finite fibers for all $w : A \to B \in \mathcal{C}$.)

Example: pullback automaton

Proposition: Finitary ULF functors are preserved by pullback along arbitrary functors.

$$\begin{tikzcd}
\mathcal{E} \times_{\mathcal{C}} \mathcal{Q} \arrow[r] \arrow[d, swap, "F^* p \text{ finULF}"] \arrow[dr, phantom, very near start, "\lrcorner"] & \mathcal{Q} \arrow[d, "p \text{ finULF}"] \\
\mathcal{E} \arrow[r, swap, "F"] & \mathcal{C}
\end{tikzcd}$$

(Easy way to see this: if $G : \mathcal{C} \to \mathrm{FinSpan}$ is the fiber functor of p, then $G \circ F$ is the fiber functor of $F^* p$.)

Corollary: Regular languages of arrows are closed under inverse image along functors, and under intersection.

Classifying automata by their fibrational properties

We say that a categorical automaton is unambiguous/deterministic/codeterministic if p is faithful/opfibration/fibration, or equivalently, just in case its fiber functor factors as follows...

Classifying automata by their fibrational properties

$$\begin{tikzcd}\mathcal{C} \arrow[rr, "\text{general NFA}"] & & \mathrm{FinSpan}\end{tikzcd}$$

Classifying automata by their fibrational properties

$$\begin{tikzcd}
\mathcal{C} \arrow[r, "\text{general NFA}"] \arrow[dr, dashed, "\text{unambiguous}"'] & \mathrm{FinSpan} \\
 & \mathrm{FinRel} \arrow[u, hook]
\end{tikzcd}$$

Classifying automata by their fibrational properties

$$\begin{tikzcd}
\mathcal{C} \arrow[rr, "\text{general NFA}"] \arrow[drr, dashed, "\text{unambiguous}"'] \arrow[ddr, dashed, "\text{deterministic}"'] & & \mathrm{FinSpan} \\
& & \mathrm{FinRel} \arrow[u, hook] \\
& \mathrm{FinSet} \arrow[ur, hook] &
\end{tikzcd}$$

Classifying automata by their fibrational properties

$$\begin{tikzcd}
\mathcal{C} \arrow[rr, "\text{general NFA}"] \arrow[drr, dashed, "\text{unambiguous}"] \arrow[ddr, dashed, swap, "\text{deterministic}"] \arrow[ddrrr, dashed, swap, "\text{codeterministic}"] & & \mathrm{FinSpan} & \\
& & \mathrm{FinRel} \arrow[u, hook] & \\
& \mathrm{FinSet} \arrow[ur, hook] & & \mathrm{FinSet}^{\mathrm{op}} \arrow[ul, hook']
\end{tikzcd}$$