Context-free grammars & finite-state automata over categories*

Noam Zeilberger

joint work with Paul-André Melliès

PSSL 109 @ Leiden

15-17 November 2024

*Paper: The categorical contours of the Chomsky-Schützenberger Representation Theorem

arXiv:2405.14703
(new version 15 Nov!)

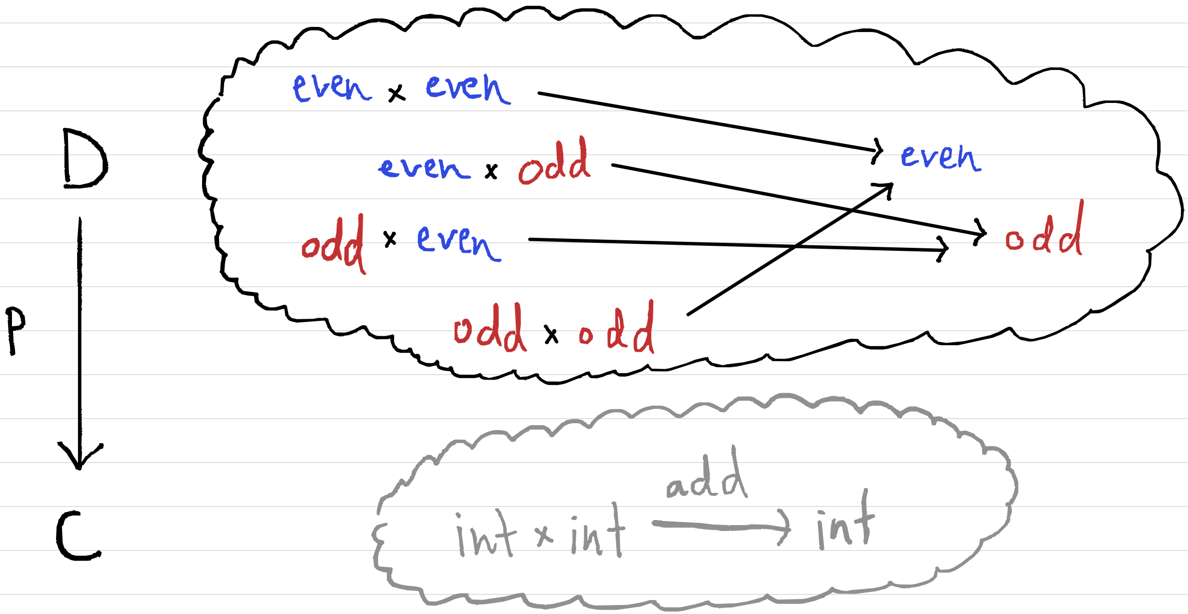
Typing as a lifting problem

Idea: model type systems fibrationally as functors that "forget" typing information.

$$\mathcal{C} \qquad \mathrm{int} \times \mathrm{int} \xrightarrow{\mathrm{add}} \mathrm{int}$$

Typing as a lifting problem

Idea: model type systems fibrationally as functors that "forget" typing information.



See "Functors are Type Refinement Systems" (POPL 2015) and other papers in series w/ PAM.

~~Typing~~ Parsing as a lifting problem

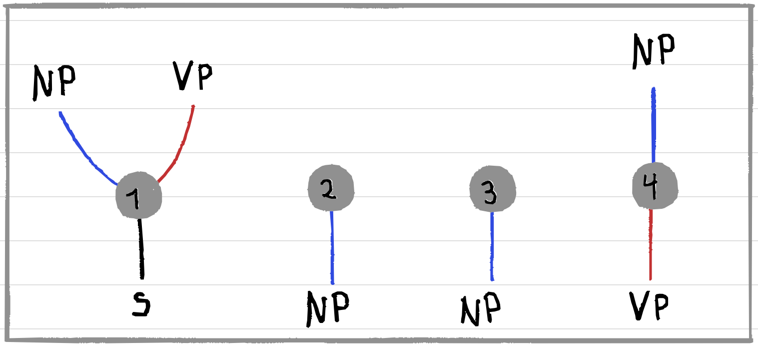
$S \to NP\ VP$

$NP \to$ Noam

$NP \to$ PSSL

$VP \to$ likes NP

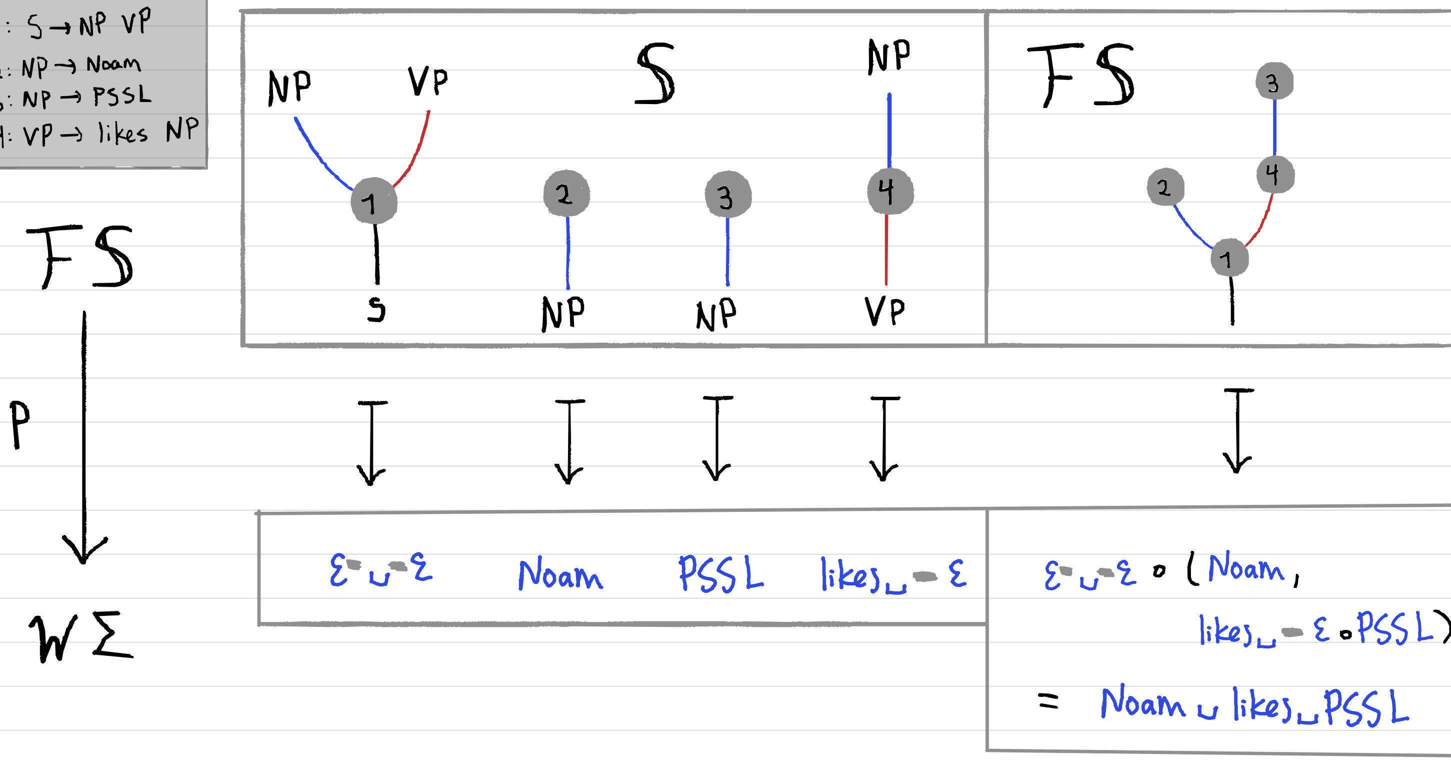
~~Typing~~ Parsing as a lifting problem



1: S → NP VP
2: NP → Noam
3: NP → PSSL
4: VP → likes NP

~~Typing~~ Parsing as a lifting problem

1: S → NP VP
2: NP → Noam
3: NP → PSSL
4: VP → likes NP

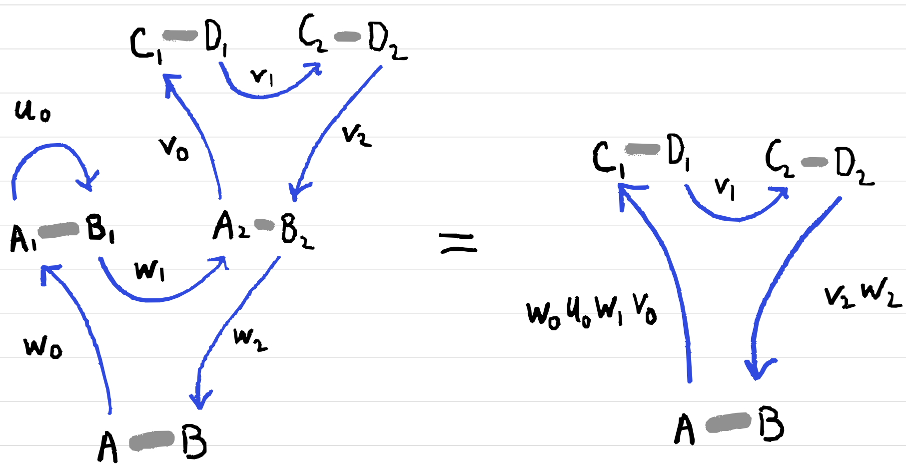


Cf. RFC Walters, "A note on context-free grammars", JPAA 62(2): 199-203, 1989.

The spliced arrows construction

Given a category $\mathcal{C}$, the operad of spliced arrows $W\mathcal{C}$ has:

- objects given by pairs (A,B) of objects $A,B \in \mathcal{C}$
- n-ary operations $f:(A_1,B_1),\dots,(A_n,B_n) \to (A,B)$ given by sequences $f = w_0 - \dots - w_n$ of $n+1$ arrows $w_i : B_i \to A_{i+1} \in \mathcal{C}$ (under convention $B_0 = A$, $A_{n+1} = B$)
- composition performed by "splicing into the gaps"...



The spliced arrows construction

Splicing extends to a functor

$$\mathrm{Cat} \xrightarrow{\mathcal{W}} \mathrm{Oper}$$

since any functor of categories $F : \mathcal{C} \to \mathcal{D}$ induces a functor of operads $\mathcal{W}F : \mathcal{W}\mathcal{C} \to \mathcal{W}\mathcal{D}$ by the mappings

$$(A,B) \longmapsto (FA, FB)$$

$$w_0 - \ldots - w_n \longmapsto Fw_0 - \ldots - Fw_n$$

Context-free grammar over a category

Definition. A CFG over a category $\mathcal{C}$ is a pair of a pointed finite species $(\$, S \in \$)$ and a functor $p : F\$ \to W\mathcal{C}$.

The context-free language of arrows generated by a CFG $G = ((\$, S), p)$ is the set $\mathcal{L}_G = \{ p(\alpha) \mid \alpha : S \} \subseteq \mathcal{C}(A, B)$.

where $(A,B) = p(S)$

Context-free grammar over a category

Definition. A CFG over a category $\mathcal{C}$ is a pair of a pointed finite species $(\mathcal{S},\ S \in \mathcal{S})$ and a functor $p : F\mathcal{S} \to \mathcal{W}\mathcal{C}$.

The context-free language of arrows generated by a CFG $G = ((\mathcal{S}, S), p)$ is the set $\mathcal{L}_G = \{ p(\alpha) \mid \alpha : S \} \subseteq \mathcal{C}(A,B)$.

where $(A,B) = p(S)$

Proposition. $L \subseteq \Sigma^*$ is context-free in the classical sense iff it is the language of arrows of a CFG over $F\mathbb{B}_\Sigma$, where $\mathbb{B}_\Sigma = \begin{tikzcd} * \arrow[loop above, "a \in \Sigma"] \end{tikzcd}$.

Context-free grammar over a category

Definition. A CFG over a category $\mathcal{C}$ is a pair of a pointed finite species $(\mathcal{S}, S \in \mathcal{S})$ and a functor $p : F\mathcal{S} \to \mathcal{W}\mathcal{C}$.
The context-free language of arrows generated by a CFG $G = ((\mathcal{S}, S), p)$ is the set $\mathcal{L}_G = \{ p(\alpha) \mid \alpha : S \} \subseteq \mathcal{C}(A,B)$.
where $(A,B) = p(S)$

Proposition. $L \subseteq \Sigma^*$ is context-free in the classical sense iff it is the language of arrows of a CFG over $F\mathbb{B}_\Sigma$, where
$$\mathbb{B}_\Sigma = \begin{tikzcd} * \arrow[loop above, "a \in \Sigma"] \end{tikzcd}.$$

Example #2: Let
$$\mathbb{B}_\Sigma^{\wedge\$} = \begin{tikzcd} \bot \arrow[r, "\wedge"] & * \arrow[loop above, "a \in \Sigma"] \arrow[r, "\$"] & \top \end{tikzcd}.$$
A CFG over $F\mathbb{B}_\Sigma^{\wedge\$}$ can have productions that are only applicable at beginning/end of string.

$$S \to E\,\$ \qquad \text{(cf. Knuth 1965)}$$

Some closure properties of CFLs

union

- If $L_1 \subseteq \mathcal{C}(A,B)$ and $L_2 \subseteq \mathcal{C}(A,B)$ are context-free then so is $L_1 \cup L_2 \subseteq \mathcal{C}(A,B)$

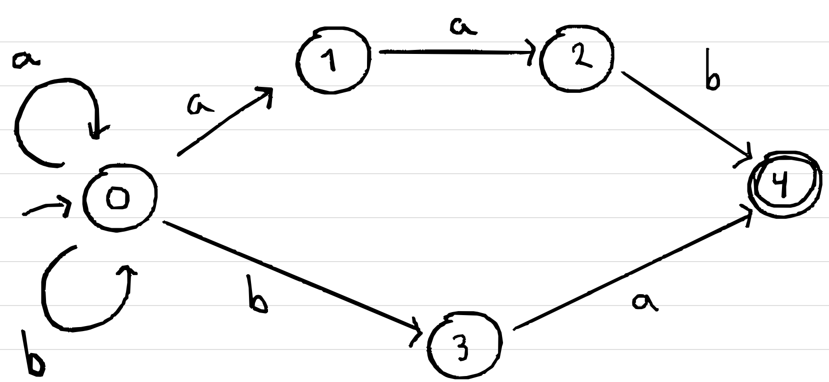
concatenation

- If $L_1 \subseteq \mathcal{C}(A_1,B_1), \ldots, L_n \subseteq \mathcal{C}(A_n,B_n)$ are context-free then so is $w_0 L_1 w_1 \cdots L_n w_n \subseteq \mathcal{C}(A,B)$ for any $w_0 - \cdots - w_n : (A_1,B_1),\ldots,(A_n,B_n) \to (A,B) \in \mathcal{W}\mathcal{C}$

homomorphic image

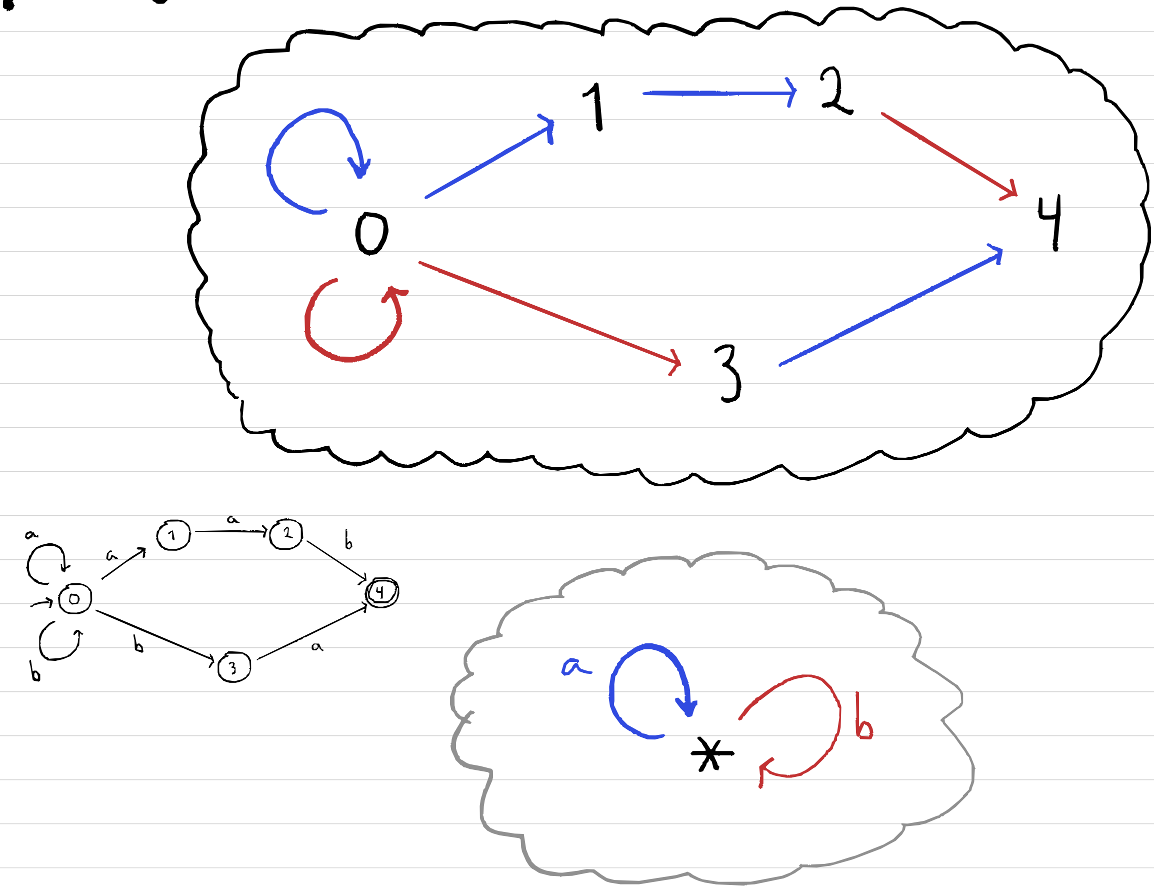
- If $L \subseteq \mathcal{C}(A,B)$ is context-free then so is $F(L) \subseteq \mathcal{D}(FA,FB)$ for any functor $F : \mathcal{C} \to \mathcal{D}$

~~Typing~~ Recognition as a lifting problem



~~Typing~~ Recognition as a lifting problem

$$\begin{tikzcd} F\mathbb{Q} \arrow[d] \\ F\mathbb{B}_\Sigma \end{tikzcd}$$



ULF and finitary functors

Let $p : D \to C$ be a functor of categories.

- p has unique liftings of factorizations (ULF aka "discrete Conduché") if for all arrows $\alpha \in D$ s.t. $p(\alpha) = uv$, $\exists! \beta, \gamma$. such that $\alpha = \beta\gamma$ and $p(\beta) = u$ and $p(\gamma) = v$.

$$
\begin{tikzcd}
D \arrow[d] & R \arrow[rr, "\alpha"] \arrow[dr, dashed, "\beta"'] & & S \\
 & & T \arrow[ur, dashed, "\gamma"'] & \\
C & A \arrow[rr, "w"] \arrow[dr, "u"'] & & B \\
 & & X \arrow[ur, "v"'] &
\end{tikzcd}
$$

ULF and finitary functors

Let $p : D \to C$ be a functor of categories.

- p has unique liftings of factorizations (ULF aka "discrete Conduché")
  if for all arrows $\alpha \in D$ s.t. $p(\alpha) = uv$, $\exists!\, \beta, \gamma$.
  such that $\alpha = \beta\gamma$ and $p(\beta) = u$ and $p(\gamma) = v$.

$$\begin{tikzcd}
D \arrow[d] & R \arrow[rr, "\alpha"] \arrow[dr, dashed, "\beta"'] & & S \\
& & T \arrow[ur, dashed, "\gamma"'] & \\
C & A \arrow[rr, "w"] \arrow[dr, "u"'] & & B \\
& & X \arrow[ur, "v"'] &
\end{tikzcd}$$

- p is finitary if the fibers $p^{-1}(A)$ and $p^{-1}(w)$ of every object $A \in C$ and every arrow $w : A \to B$ are finite.

ULF and finitary functors

Let $p : D \to C$ be a functor of categories.

- p has unique liftings of factorizations (ULF aka "discrete Conduché") if for all arrows $\alpha \in D$ s.t. $p(\alpha) = uv$, $\exists! \beta, \gamma$. such that $\alpha = \beta\gamma$ and $p(\beta) = u$ and $p(\gamma) = v$.

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

- p is finitary if the fibers $p^{-1}(A)$ and $p^{-1}(w)$ of every object $A \in C$ and every arrow $w : A \to B$ are finite.

Proposition. Let $F : C \to \mathrm{Span}(\mathrm{Set})$ be the lax functor canonically representing the functor $p : D \to C$. (As a "displayed category".)

- p is ULF iff F is a pseudofunctor
- p is finitary iff F factors via $\mathrm{Span}(\mathrm{FinSet})$

ULF and finitary functors

Proposition (Street 1996, cf. Guetta 2020). Let $p : \mathcal{D} \to \mathcal{C}$ be a functor into a category $\mathcal{C} \simeq F\mathbb{G}$ freely generated by some graph $\mathbb{G}$. Then p is ULF iff $\mathcal{D} \simeq F\mathbb{H}$ and $p = F\phi$ for some graph $\mathbb{H}$ and homomorphism $\phi : \mathbb{H} \to \mathbb{G}$.

Proposition. Let $\phi : \mathbb{H} \to \mathbb{G}$ be a homomorphism into a finite graph $\mathbb{G}$. Then ϕ is finitary iff $\mathbb{H}$ is finite.

Corollary. A functor $p : Q \to F\mathbb{B}_\Sigma$ represents the underlying bare automaton of an NFA over Σ iff p is ULF and finitary.

Finite-state automaton over a category

Definition. A NFA over a category $\mathcal{C}$ is a pair of a bipointed category $(Q, q_0 \in Q, q_f \in Q)$ and a finitary ULF functor $p : Q \to \mathcal{C}$. The regular language of arrows recognized by an NFA $M = ((Q, q_0, q_f), p)$ is the set $\mathcal{L}_M = \{ p(\alpha) \mid \alpha : q_0 \to q_f \} \subseteq \mathcal{C}(A, B)$.
where $(A, B) = (p(q_0), p(q_f))$

Finite-state automaton over a category

Definition. A NFA over a category C is a pair of a bipointed category $(Q, q_0 \in Q, q_f \in Q)$ and a finitary ULF functor $p : Q \to C$. The regular language of arrows recognized by an NFA $M = ((Q, q_0, q_f), p)$ is the set $\mathcal{L}_M = \{ p(\alpha) \mid \alpha : q_0 \to q_f \} \subseteq C(A,B)$.

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

Proposition. $L \subseteq \Sigma^*$ is regular in the classical sense iff it is the language of arrows of a NFA over $FB_\Sigma^{1\$}$.

Finite-state automaton over a category

Definition. A NFA over a category C is a pair of a bipointed category $(Q, q_0 \in Q, q_f \in Q)$ and a finitary ULF functor $p : Q \to C$. The regular language of arrows recognized by an NFA $M = ((Q, q_0, q_f), p)$ is the set $\mathcal{L}_M = \{ p(\alpha) \mid \alpha : q_0 \to q_f \} \subseteq C(A,B)$.

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

Proposition. $L \subseteq \Sigma^*$ is regular in the classical sense iff it is the language of arrows of a NFA over $FB_\Sigma^{1\$}$.

A NFA M is (total) deterministic iff p is a discrete opfibration. It is codeterministic iff p is a discrete fibration.

cf. Colcombet & Petrişan 2020

Some examples of categorical NFA

Product automaton $M \times M' :=$

$$\begin{array}{c} Q \times Q' \\ \Big\downarrow {\scriptstyle p \times p'} \\ C \times C' \end{array}$$

w/ initial state (q_0, q_0')

final state (q_f, q_f')

$$\mathcal{L}_{M \times M'} = \mathcal{L}_M \times \mathcal{L}_{M'}$$

Some examples of categorical NFA

Product automaton $M \times M' :=$
$$\begin{array}{c} Q \times Q' \\ \Big\downarrow{\scriptstyle p \times p'} \\ C \times C' \end{array}$$
w/ initial state (q_0, q_0')
final state (q_f, q_f')

$$\mathcal{L}_{M \times M'} = \mathcal{L}_M \times \mathcal{L}_{M'}$$

Total automaton $M_{C(A,B)} :=$
$$\begin{array}{c} C \\ \Big\downarrow{\scriptstyle id} \\ C \end{array}$$
w/ initial state A
final state B

$$\mathcal{L}_{M_{C(A,B)}} = C(A,B)$$

Some examples of categorical NFA

Product automaton $M \times M' :=$ $\begin{array}{c} Q \times Q' \\ \big\downarrow{\scriptstyle p \times p'} \\ C \times C' \end{array}$ w/ initial state (q_0, q_0')
final state (q_f, q_f')

$\mathcal{L}_{M \times M'} = \mathcal{L}_M \times \mathcal{L}_{M'}$

Total automaton $M_{C(A,B)} :=$ $\begin{array}{c} C \\ \big\downarrow{\scriptstyle id} \\ C \end{array}$ w/ initial state A
final state B

$\mathcal{L}_{M_{C(A,B)}} = C(A,B)$

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

Singleton automaton $M_w :=$ $\begin{array}{c} \mathrm{Fact}_w \\ \big\downarrow{\scriptstyle p} \\ C \end{array}$ w/ initial state (id_A, w)
final state (w, id_B)

$\mathcal{L}_{M_w} = \{ w \}$

Requirement: C has finitary factorizations

An aside on ε-transitions

Naturally modelled as arrows $\alpha : q \to q'$ such that $p(\alpha) = id$.

... but ULF implies no such arrows! $(p(id_q\,\alpha) = p(\alpha\, id_{q'}) = id)$

So to model ε-transitions it seems we need to weaken ULF.

... but the general Conduché property seems too weak.
(wrong version of "ε-removal")

Automata over operads

Definition. NFA over an operad $\mathcal{O}$ is a pair $M = (\dot{Q}, p)$ of a pointed operad $\dot{Q} = (Q, q_r \in Q)$ + a finitary ULF functor $p : Q \to \mathcal{O}$, recognizing a regular language of constants $\mathcal{L}_M = \{ p(\alpha) \mid \alpha : q_r \} \subseteq \mathcal{O}(A)$

where $p(q_r) = A$

Automata over operads

Definition. NFA over an operad $\mathcal{O}$ is a pair $M = (\dot{Q}, p)$ of a pointed operad $\dot{Q} = (Q, q_r \in Q)$ + a finitary ULF functor $p : Q \to \mathcal{O}$, recognizing a regular language of constants $\mathcal{L}_M = \{ p(\alpha) \mid \alpha : q_r \} \subseteq \mathcal{O}(A)$

where $p(q_r) = A$

Regular tree language $\Leftrightarrow$ recognized by NFA over free operad.

Automata over operads

Definition. NFA over an operad $\mathcal{O}$ is a pair $M = (\dot{Q}, p)$ of a pointed operad $\dot{Q} = (Q, q_r \in Q)$ + a finitary ULF functor $p : Q \to \mathcal{O}$, recognizing a regular language of constants $\mathcal{L}_M = \{p(\alpha) \mid \alpha : q_r\} \subseteq \mathcal{O}(A)$
where $p(q_r) = A$

Regular tree language $\Leftrightarrow$ recognized by NFA over free operad.

Proposition. If $p : D \to C$ is ULF (resp. finitary) then so is $Wp : WD \to WC$

Hence any categorical NFA $M = ((Q, q_0, q_f), p)$ induces an operadic NFA $WM = ((WQ, (q_0, q_f)), Wp)$ with

$$\mathcal{L}_{WM} = \mathcal{L}_M .$$

The Chomsky-Schützenberger Repr Thm (1963)

"A language is context-free iff it is a homomorphic image of a Dyck language with a regular language."

Key observation: the functor $\mathcal{W}$ has a left adjoint!

$$\begin{tikzcd} \mathrm{Oper} \arrow[rr, bend left, "C"] \arrow[rr, phantom, "\perp"] & & \mathrm{Cat} \arrow[ll, "\mathcal{W}"] \end{tikzcd}$$

See our paper for the contour category construction, and for the proof of a generalized C-S rep theorem.

One ingredient: closure of CFLs under intersection w/ RLs...

Pulling back a CFG along an NFA

Lemma. The pullback of a functor of operads $p : F\mathcal{S} \to \mathcal{O}$ along a ULF functor of operads $p_{\mathcal{Q}} : \mathcal{Q} \to \mathcal{O}$ is obtained from a pullback of $\phi : \mathcal{S} \to \mathcal{O}$ along $p_{\mathcal{Q}}$ in species.

$$\begin{tikzcd}
\mathcal{S}' \arrow[r, "\gamma'"] \arrow[d, "\phi' = p_{\mathcal{Q}}^* \phi"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathcal{S} \arrow[d, "\phi"] \\
\mathcal{Q} \arrow[r, "p_{\mathcal{Q}}"'] & \mathcal{O}
\end{tikzcd}
\qquad
\begin{tikzcd}
F\mathcal{S}' \arrow[r, "F\gamma'"] \arrow[d, "p' = p_{\mathcal{Q}}^* p"'] \arrow[dr, phantom, "\lrcorner", very near start] & F\mathcal{S} \arrow[d, "p"] \\
\mathcal{Q} \arrow[r, "p_{\mathcal{Q}}"'] & \mathcal{O}
\end{tikzcd}$$

Spec   Oper

Moreover, if $\mathcal{S}$ is finite and $p_{\mathcal{Q}}$ is finitary then $\mathcal{S}'$ is finite.

Pulling back a CFG along an NFA

Let $G = ((\mathcal{S}, S), p_G)$ be a CFG and $M = ((Q, q_0, q_f), p_M)$ a NFA over the same category w/ $p_G(S) = (p_M(q_0), p_M(q_f))$. Take the pullback:

$$
\begin{tikzcd}
F\mathcal{S}' \arrow[r, "F\pi'"] \arrow[d, "p_G' = W p_M^* p_G"'] \arrow[dr, phantom, "\lrcorner", very near start] & F\mathcal{S} \arrow[d, "p_G"] & Q \arrow[d, "p_M"] \\
WQ \arrow[r, "W p_M"'] & WC & C
\end{tikzcd}
$$

Then $M^*G := ((\mathcal{S}', (S, (q_0, q_f))), p_G')$ is a CFG generating

$$\mathcal{L}_{M^*G} = p_M^{-1}(\mathcal{L}_G) \cap Q(q_0, q_f).$$

CFL of runs of the NFA M!

Corollary: CFLs closed under intersection with RLs.

Conclusion

See paper (arXiv: 2405.14703) for more on:

- Translations between CFGs

$$\begin{tikzcd} FS_1 \arrow[rr] \arrow[dr] & & FS_2 \arrow[dl] \\ & WC & \end{tikzcd}$$

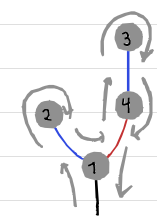
- Generalized CFGs over operads

$$a^n \# b^n \#' c^n \quad t \quad \begin{tikzcd} FS \arrow[d] \\ !_{aff} WC \end{tikzcd}$$

- More properties of finitary ULF functors, and closure properties of regular languages

$$\begin{tikzcd} & & \cdot \arrow[dl, "\mathrm{fin\,ULF}"'] \arrow[dr] & & \\ & Q \arrow[dl, "\mathrm{fin\,ULF}"'] \arrow[dr] & & Q' \arrow[dl, "\mathrm{fin\,ULF}"'] \arrow[dr] & \\ X & & Y & & Z \end{tikzcd}$$

- The contour category construction and the universal CFG of a pointed finite species



Long term goals:

- Categorify more of automata theory & parsing theory
- Tranfer knowledge back to type theory and category theory?