Automata on Graph Alphabets

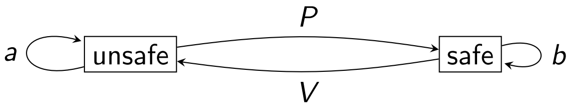
Hugo Bazille[1] Uli Fahrenberg[2]

EPITA Research Lab (LRE), Rennes/Paris, France
LMF, Université Paris-Saclay, France

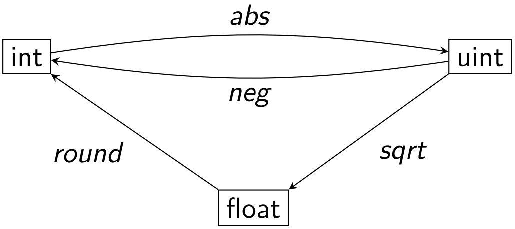
RAMiCS 2026

LMF — Laboratoire Méthodes Formelles

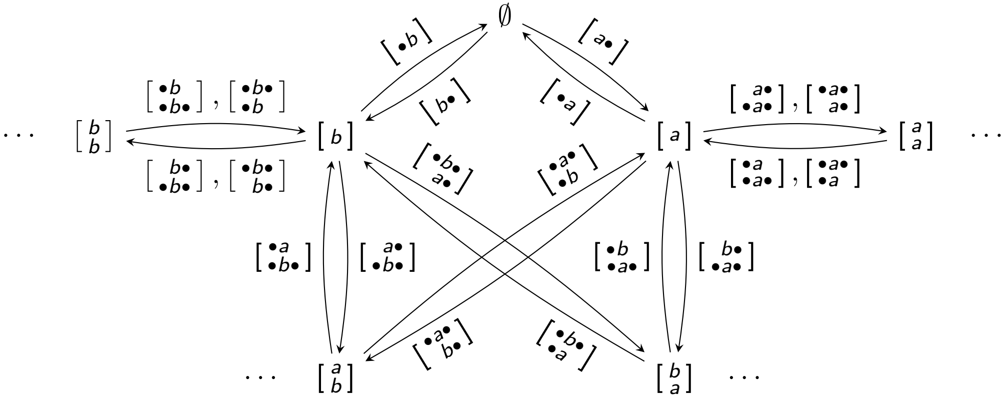
Some Structured Alphabets



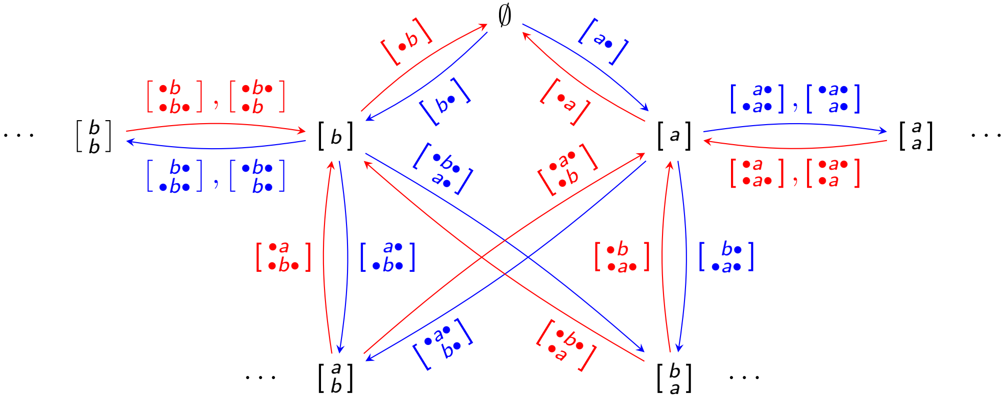
Some Structured Alphabets



Some Structured Alphabets



Some Structured Alphabets



Automata on Graph Alphabet

Definition: (directed multi)graph alphabet (V, Σ, d_0, d_1)

- vertices V; edges Σ; $d_0, d_1 : \Sigma \to V$ sources / targets (will omit d_0, d_1)

Definition: automaton $(Q, I, F, E, s, t, \mu, \lambda)$ over alphabet (V, Σ)

- states Q; $I, F \subseteq Q$ initial / accepting; transitions E; $s, t : E \to Q$ sources / targets
- $\mu : Q \to V$, $\lambda : E \to \Sigma$ labelings
- such that $\mu(s(e)) = d_0(\lambda(e))$ and $\mu(t(e)) = d_1(\lambda(e))$ for every $e \in E$

Automata on Graph Alphabet

Definition: (directed multi)graph alphabet (V, Σ, d_0, d_1)

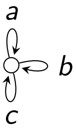
- vertices V; edges Σ; $d_0, d_1 : \Sigma \to V$ sources / targets (will omit d_0, d_1)

Definition: automaton $(Q, I, F, E, s, t, \mu, \lambda)$ over alphabet (V, Σ)

- states Q; $I, F \subseteq Q$ initial / accepting; transitions E; $s, t : E \to Q$ sources / targets
- $\mu : Q \to V$, $\lambda : E \to \Sigma$ labelings
- such that $\mu(s(e)) = d_0(\lambda(e))$ and $\mu(t(e)) = d_1(\lambda(e))$ for every $e \in E$

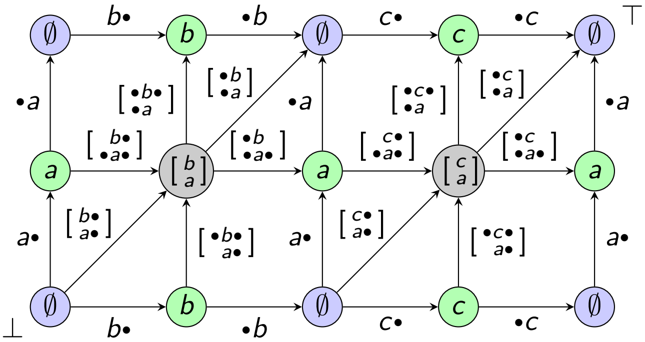
- automaton $\triangleq$ graph + homomorphism into (V, Σ)
- or (if you wish) a covariant presheaf on (V, Σ)

Example



- usual alphabets are graph alphabets

Example



- ST-automaton accepting $a \parallel bc$

Languages

Let $A = (Q, I, F, E, s, t, \mu, \lambda)$ be an automaton over alphabet (V, Σ).

- a path in A: alternating sequence $\pi = (q_0, e_1, q_1, \dots, e_n, q_n) \in Q(EQ)^*$ such that $s(e_i) = q_{i-1}$ and $t(e_i) = q_i$ for all i (as usual)
- the label of π: $\lambda(\pi) = (\mu(q_0), \lambda(e_1) \cdots \lambda(e_n), \mu(q_n))$ (keeping source and target information)
- (no empty paths!)
- the language of A: $L(A) = \{\lambda(\pi) \mid \pi \text{ path in } A, s(\pi) \in I, t(\pi) \in F\}$
- Lemma: $\{\omega \in \Sigma^* \mid \exists u, v \in V : (u, \omega, v) \in L(A)\}$ is regular.

Languages

Let $A = (Q, I, F, E, s, t, \mu, \lambda)$ be an automaton over alphabet (V, Σ).

- a path in A: alternating sequence $\pi = (q_0, e_1, q_1, \dots, e_n, q_n) \in Q(EQ)^*$ such that $s(e_i) = q_{i-1}$ and $t(e_i) = q_i$ for all i (as usual)
- the label of π: $\lambda(\pi) = (\mu(q_0), \lambda(e_1) \cdots \lambda(e_n), \mu(q_n))$ (keeping source and target information)
- (no empty paths!)
- the language of A: $L(A) = \{\lambda(\pi) \mid \pi \text{ path in } A, s(\pi) \in I, t(\pi) \in F\}$
- Lemma: $\{\omega \in \Sigma^* \mid \exists u, v \in V : (u, \omega, v) \in L(A)\}$ is regular.

- $L(A)$ is a set of morphisms in $(V, \Sigma)^*$, the free category generated by (V, Σ)
- (not necessarily a subcategory)

Results

- Kleene theorem ✓
- Myhill-Nerode theorem ✓
- determinization, minimization ✓

Kleene theorem

- basic sets: $\emptyset$ and $\{(d_0(a), a, d_1(a))\}$ for each $a \in \Sigma$
- rational sets:
  - basic sets;
  - $X \cup Y$ for X, Y rational
  - XY for X, Y rational
  - X^+ for X rational

Theorem

X is rational iff $X = L(A)$ for some automaton A.

Proof.

All the standard constructions still work. □

Myhill-Nerode theorem

- for $X \subseteq (V,\Sigma)^*$ and $w \in (V,\Sigma)^*$ define:
  - $w^{-1}X = \{u \in (V,\Sigma)^* \mid d_0(u) = d_1(w), wu \in X\}$
  - $\text{suff}(X) = \{w^{-1}X \mid w \in (V,\Sigma)^*\}$

Theorem

X is rational iff $\text{suff}(X)$ is finite.

Proof.

Nothing special, but a new (?) proof for $\Longleftarrow$ (Thanks to Krzysztof Ziemiański!):

1. Define universal tree U:
   - states $Q = (V,\Sigma)^*$, edges $E = \{(x, a, xa) \mid x \in Q, a \in \Sigma, d_1(x) = d_0(a)\}$
   - (the unfolding of (V,Σ))
2. Restrict U so that $L(U_{\restriction X}) = X$
3. Quotient restriction by $x \sim_X y \iff x^{-1}X = y^{-1}X$

□

Determinization, minimization

- standard subset construction still works for determinization
- and Myhill-Nerode construction gives minimal deterministic automaton

Conclusion?

- When generalizing alphabets to (directed multi)graphs, everything still works!
- Paper rejected at ICALP (too simple and lack of motivation (?!))
- Out motivation: ST-automata (and their relationship with higher-dimensional automata)
- Generalizations (work in progress):
  - alphabets which are simplicial sets
    - induce an eqivalence relation on words
    - everything still seems to work
  - alphabets which are directed hypergraphs
    - "polymorphic types"
    - more difficult!
  - directed hypergraphs with (simplicial?) equivalence relation
    - finally allow for $ab \sim ba$
    - $\rightsquigarrow$ trace theory! known territory; known to be difficult