Regular Monoidal Languages

Matt Earnshaw\textsuperscript{1}

j.w.w. Paweł Sobociński\textsuperscript{1}

\textsuperscript{1}Tallinn University of Technology, Estonia

MFCS, Vienna
August 2022
Regular languages with pictures

Fact: any regular language can be pictured in this way.
Bottom-up tree languages with pictures

What about multiple wires on the left?

Fact: any bottom-up regular tree language can be pictured in this way.
Top-down tree languages with pictures

What about multiple wires on the right?

---

Fact: any top-down regular tree language can be pictured in this way.
Regular monoidal languages

What about multiple wires on the left and right?

\[
\begin{array}{ccccccccc}
\begin{array}{ccccccccc}
\uparrow & H_1 & H_1 & V_0 & H_0 & V_0 & H_0 & V_1 & H_1 & V_1 & V_1 & \downarrow \\
\downarrow & V_1 & V_1 & H_0 & V_0 & H_0 & V_1 & H_1 & V_0 & H_1 & H_1 & \uparrow \\
\end{array}
\end{array}
\]

How to define these pictures formally?
Monoidal graphs

A monoidal graph is a pair of functions \( s, t : E \rightarrow V^* \).

The components of a monoidal graph are *generators*.

When *single-sorted*, \( s \) and \( t \) give natural numbers: *arity* and *coarity*.

Morphism of monoidal graphs is a pair of functions \( E \rightarrow E', V \rightarrow V' \) commuting with \( s \) and \( t \).

All of our monoidal graphs will be finite.
Regular monoidal grammars and languages

Defines a language: the $0 \rightarrow 0$ string diagrams that can be built.

\[
\begin{align*}
&\text{\epsilon \in L(\phi)} & \text{\epsilon \notin L(\phi)}
\end{align*}
\]

Definition

Languages so definable are regular monoidal languages.
Non-deterministic monoidal automata

Definition $\Delta = (V, \Delta_\Gamma)$

- $V$, finite set
- $\Gamma$, monoidal alphabet
- $\Delta_\Gamma = \{ V^{\text{ar}(\gamma)} \xrightarrow{\Delta_{\gamma}} \mathcal{P}(V^{\text{coar}(\gamma)}) \}_{\gamma \in E_\Gamma}$, set of transition relations

String diagrams $0 \rightarrow 0$ map to a $V^0 \rightarrow \mathcal{P}(V^0)$ (accept/reject).

By restricting $\Gamma$ we recover:

- Ordinary non-deterministic automata
- Top-down tree automata
- Bottom-up tree automata
The problem of determinization

Challenge
Characterize the deterministically recognizable RMLs.

Partial answers:
- convex automata
- necessary property of deterministic language
- algebraic invariant
Partial answer I: Convex automata

A monoidal automaton is convex if its transition relations are convex.

**Theorem**

Convex automata can be determinized, by an analogue of the powerset construction. E.g. word automata and bottom-up tree automata.
Partial answer I: Convex automata

A monoidal automaton is convex if its transition relations are convex.

**Theorem**
Convex automata can be determinized, by an analogue of the powerset construction. E.g. word automata and bottom-up tree automata.
Partial answer II: Causal closure

Causal histories recombinable via equations in cartesian restriction categories

Theorem: Deterministically recognizable RMLs are causally closed.
Partial answer III: Syntactic pro

\[ \alpha \vdash \{ \text{n} \{ \text{m} \{ \beta } \}\]

\[ \gamma \equiv_L \delta \text{ if } C[\gamma] \in L \iff C[\delta] \in L, \text{ for all contexts } C \]

**Theorem**

*If* \( L \) *is an RML then its syntactic pro has finite homsets.*

**Theorem**

*If the syntactic pro of an RML has cartesian restriction category structure, then the language is deterministically recognizable.*
Future work

- Completely characterize deterministic recognizability
- Embeddings of word languages
- Diagrammatics for pushdown and Zielonka automata, transducers, etc.
- Context-free monoidal languages via a monoidal multicategory of contexts

Thanks for your attention.