Goldstine Automata

characteristics of automata:

sequentiality $\downarrow$ monoid **and** non-determinism $\downarrow$ comm. monoid

Def. (monoid). A monoid is a tuple $(M, \odot, \mathbb{1})$ where M is a set, $\odot$ is a binary operation on M, $\mathbb{1} \in M$, $\odot$ is associative, and $\mathbb{1}$ is neutral w.r.t. $\odot$. A monoid is called commutative if $\odot$ is commutative.

Def. (product monoid). Let $(M_1, \odot_1, \mathbb{1}_1), \ldots, (M_k, \odot_k, \mathbb{1}_k)$ be monoids. The product of $M_1, \ldots, M_k$, denoted by $M_1 \times \ldots \times M_k$, is the monoid $(M_1 \times \ldots \times M_k, \odot, \mathbb{1})$ where $\mathbb{1} = \langle \mathbb{1}_1, \ldots, \mathbb{1}_k \rangle$ and $\langle m_1, \ldots, m_k \rangle \odot \langle m_1', \ldots, m_k' \rangle = \langle m_1 \odot_1 m_1', \ldots, m_k \odot_k m_k' \rangle$.

Ex. (free monoid). Let Σ be a set. The free monoid over Σ is the monoid $(\Sigma^*, \circ, \varepsilon)$.

Ex. (free commutative monoid) Let Σ be a set. The free comm. monoid over is the monoid $(\Sigma \to \mathbb{N} \cup \{\infty\}, \oplus, \mathbb{0})$ where $\oplus$ is the point-wise addition and $\mathbb{0}$ is the function that always returns 0.

Def. (Goldstine-automaton) Let $M_1 \times \ldots \times M_k$ be a product monoid. An $M_1 \times \ldots \times M_k$-automaton is a finite subset of $M_1 \times \ldots \times M_k$, the elements of which are called transitions.

Def. (semantics of Goldstine-automata). Let T be an $M_1 \times \ldots \times M_k \times N_1 \times \ldots \times N_\ell$-automaton, $B \subseteq M_1 \times \ldots \times M_k$, and $\varphi : N_1 \times \ldots \times N_\ell \to O$ be a homomorphism into some commutative monoid $(O, +, \mathbb{0})$. The (B, φ)-semantics of T is $[\![T]\!]_{B,\varphi} = \varphi\left(\Sigma^{\oplus}(\bar{n} \mid \langle \bar{m}, \bar{n} \rangle \in [\![T^*]\!], \bar{m} \cap B \neq \emptyset)\right)$

B.. internal behaviour

φ.. output behaviour

O.. output algebra

Goldstine Automata (examples)

Ex. (finite state automata) Let $\mathcal{M} = (Q, \Sigma, Q_i, Q_f, T)$ be a (usual) finite state automaton. Each transition has the form $\tau = (q, u, q')$ where $q, q' \in Q$ and $u \in \Sigma \cup \{\varepsilon\}$. We rewrite this transition as $\tilde{\tau} = \langle \{(q,q')\}, u \rangle$ where $\{(q,q')\}$ is an element of the monoid of binary relations on the set Q with id_Q as the neutral element and composition as the binary, associative operation on $Q \times Q$; and u is an element of the monoid $(\Sigma^*, \circ, \varepsilon)$. Furthermore, let $B = \mathcal{P}(Q_i \times Q_f) \setminus \{\emptyset\}$, $\mathcal{O} = (\mathcal{P}(\Sigma^*), \cup, \emptyset)$ and $\varphi : \Sigma^* \to \mathcal{P}(\Sigma^*)$ with $\varphi : u \mapsto \{u\}$.

The (B, φ)-semantics of the $(Q \times Q) \times \Sigma^*$-automaton $\tilde{T} = \{\tilde{\tau} \mid \tau \in T\}$ is

$$\begin{aligned}
[\![\tilde{T}]\!]_{B,\varphi} &= \varphi\left(\Sigma^{\oplus}(w) \mid \langle r, w \rangle \in [\![\tilde{T}^*]\!], r \cap (Q_i \times Q_f) \neq \emptyset \right) \\
&= \bigcup \left(\{w\} \mid \langle r, w \rangle \in [\![\tilde{T}^*]\!], r \cap (Q_i \times Q_f) \neq \emptyset \right) \\
&= \bigcup \left(\{u_1 \cdots u_k\} \mid \langle \{(q_0,q_1)\}, u_1 \rangle, \ldots, \langle \{(q_{k-1}, q_k)\}, u_k \rangle \in \tilde{T}, \right. \\
&\qquad\qquad \left. q_0 \in Q_i, q_k \in Q_f \right) \\
&= L(\mathcal{M})
\end{aligned}$$

exactly the language of $\mathcal{M}$.

Ex. (weighted finite-state automaton)

- $\mathcal{M} = (Q, \Sigma, \mu_i, \mu_f, \mu_\tau)$, $\mu_i, \mu_f : Q \to (\mathcal{A}, \oplus, \odot, \mathbb{0}, \mathbb{1})$
- $\mu_\tau : Q \times (\Sigma \cup \{\varepsilon\}) \times Q \to \mathcal{A}$, $\mathcal{A}$ $\oplus$-commutative bimonoid
- $\tilde{T} = \{ \langle \{(q,q')\}, u, a \rangle \mid (q, u, q') \in Q \times (\Sigma \cup \{\varepsilon\}) \times Q,$
  $a = \mu_\tau((q,u,q')) \}$
- $B = \mathcal{P}(Q_i \times Q_f) \setminus \{\emptyset\}$
- $\mathcal{O} = (\Sigma^* \to \mathcal{A}, \hat{\oplus}, \text{const } \mathbb{0})$
  $(f \mathbin{\hat{\oplus}} g)(w) = f(w) \oplus g(w)$
- $\varphi(\langle u, a \rangle) = \{(u, a)\} \cup \{(u', \mathbb{0}) \mid u' \in \Sigma^* \setminus \{u\}\}$

Goldstine automata (examples)

Ex. (string transducers)

- $\mathcal{M} = (Q, \Sigma, \Delta, Q_i, Q_f, T)$
- $T \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Delta^* \times Q$ finite
- $\tilde{T} = \{\langle \{(q,q')\}, u, v\rangle \mid (q,u,v,q') \in T\}$
- $B = \mathcal{P}(Q_i \times Q_f) \setminus \{\emptyset\}$
- $O = (\Sigma^* \to \mathcal{P}(\Delta^*), \circledcup, \text{const } \emptyset)$ where

  $(f \circledcup g)(w) = f(w) \cup g(w) \quad \forall w \in \Sigma^*$
- $\varphi(\langle u, v\rangle) = \{(u,v)\}$

$$
\begin{aligned}
[\![\tilde{T}]\!]_{B,\varphi} &= \varphi\left(\Sigma^{\oplus}(\langle u,v\rangle) \mid \langle r,u,v\rangle \in [\![\tilde{T}^*]\!],\ r \cap (Q_i \times Q_f) \neq \emptyset\right) \\
&= \circledcup\left(\{(u,v)\} \mid \langle r,u,v\rangle \in [\![\tilde{T}^*]\!],\ r \cap (Q_i \times Q_f) \neq \emptyset\right) \\
&= \{(u, vs) \mid u \in \Sigma^*, \\
&\qquad vs = \{v \mid \langle r,u,v\rangle \in [\![\tilde{T}^*]\!],\ r \cap (Q_i \times Q_f) \neq \emptyset\}, \\
&\qquad vs \neq \emptyset\} \\
&= \tau(\mathcal{M})
\end{aligned}
$$

Ex. (automata with storage)

- $\mathcal{M} = (Q, \Sigma, DS, Q_i, Q_f, T)$
- $DS = (C, I, C_i, C_f)$
- $T \subseteq Q \times I \times (\Sigma \cup \{\varepsilon\}) \times Q$ finite
- $\tilde{T} = \{\langle \{(q,q')\}, i, u\rangle \mid (q,i,u,q') \in T\}$
- $B = (\mathcal{P}(Q_i \times Q_f) \setminus \{\emptyset\}) \times (\mathcal{P}(C_i \times C_f) \setminus \{\emptyset\})$
- $O = (\mathcal{P}(\Sigma^*), \cup, \emptyset)$
- $\varphi(\langle u\rangle) = \{u\}$

Goldstine automata (genealogy)

Parameter I: internal behaviour

- automaton exhibits state behaviour (over finite set Q)
  - $\rightarrow$ state transition $q \to q'$ becomes a singleton binary relation $\{\langle q,q'\rangle\} \subseteq Q \times Q$

state beh. is just spec. storage beh.

- automaton exhibits storage behaviour (over set C)
  - $\rightarrow$ storage instructions can be directly taken over
  - $\rightarrow$ storage predicates become partial identities

predicates are superfluous

Parameter II: output behaviour / Parameter III: weights

- automaton models a language ($\subseteq \Sigma^*$)
  - $\rightarrow$ $O = (\mathcal{P}(\Sigma^*), \cup, \emptyset)$
  - $\rightarrow$ $\varphi(\langle u\rangle) = \{u\}$
- automaton models a transduction ($\subseteq \Sigma^* \times \Delta^*$)
  - $\rightarrow$ $O = (\mathcal{P}(\Sigma^* \times \Delta^*), \cup, \emptyset)$
  - $\rightarrow$ $\varphi(\langle u,v\rangle) = \{(u,v)\}$
- automaton models a weighted language ($\in \Sigma^* \to \mathcal{A}$) $(\mathcal{A}, \oplus, \odot, \mathbb{0}, \mathbb{1})$
  - $\rightarrow$ $O = (\Sigma^* \to \mathcal{A}, \oplus, \text{const } \mathbb{0})$
    $(s_1 \oplus s_2)(w) = s_1(w) + s_2(w)$
  - $\rightarrow$ $\varphi(\langle u,a\rangle) = (\text{const } \mathbb{0})[u/a]$
- automaton models a weighted transduction ($\in \Sigma^* \times \Delta^* \to \mathcal{A}$)
  - $\rightarrow$ $O = (\Sigma^* \times \Delta^* \to \mathcal{A}, \oplus, \text{const } \mathbb{0})$
    $(s_1 \oplus s_2)(u,v) = s_1(u,v) + s_2(u,v)$
  - $\rightarrow$ $\varphi(\langle u,v,a\rangle) = (\text{const } \mathbb{0})[(u,v)/a]$

Transitions:

$$T \subseteq \underbrace{M_1 \times \dots \times M_k}_{\text{internal behaviour}} \times \underbrace{N_{in}}_{\text{input}} \times \underbrace{N_{out}}_{\text{output}} \times \underbrace{N_{wt}}_{\text{weight}}$$

(present if relevant)

Goldstine automata (outlook)

- demonstrate usefulness (by exhibiting proofs)
  - → proofs of "algebraic" properties should benefit
  - → usual proof-techniques could be translated
  - → hopefully find new proof-techniques
- extend to the tree case

  only bottom-up?

  weighted/unweighted tree automata/transducers

  with/without storage

  characteristics: branching and non-determinism

  $\Sigma^{(k)} \to \mathrm{Ops}_k(\mathcal{A})$ Σ-algebra — comm. monoid
- extend to the unranked tree case

  characteristics: branching and non-determinism

  $\Sigma \to \mathrm{Ops}^*(\mathcal{A})$ unranked Σ-algebra — comm. monoid

simplicity is a "use"

start with closure properties

maybe start from $[\![\cdot]\!]$-def.

what about top-down tree transducers?

What about associativity?

relation to M-monoids?

Problems/questions (from the discussion)

→ $M \times N$ enough for syntax of Goldstine automata?

→ $[\![T^*]\!] = \langle T, \odot, 1\!\!1 \rangle$ (submonoid generated by T)

→ maybe even one monoid is enough

→ is this more than group theory?