Kleene Algebra — Lecture 3

ESSLLI 2023

Last lectures

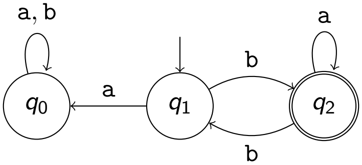
- Language semantics abstract from meaning of symbols.
- This model is equivalent to the relational semantics.
- Teased questions of *decidability* and *completeness*.

Today's lecture

- Automata as a way of representing languages.
- Decidability of language equivalence for automata.
- Translation of rational expressions to automata.
- Upshot: language equivalence of expressions is decidable.

Automata

An automaton is an *abstract machine* representing possible behaviors.



Definition (Automata)

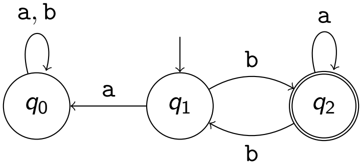
An automaton is a triple $\langle Q, \rightarrow, I, F \rangle$ where

- Q is the set of *states*, and
- $\rightarrow \subseteq Q \times \Sigma \times Q$ is the *transition relation*, and
- with $I, F \subseteq Q$ are the *initial* and *accepting states*, respectively.

When $\langle q, \mathtt{a}, q' \rangle \in \rightarrow$, we write $q \xrightarrow{\mathtt{a}} q'$.

Determinism

A *deterministic* automaton has no “ambiguity” in the transitions.



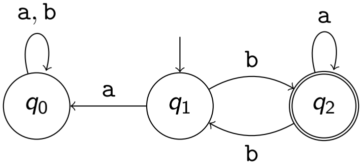
Definition (Determinism)

An automaton $\langle Q, \to, I, F \rangle$ is *deterministic* when for each $q \in Q$ and $\mathtt{a} \in \Sigma$ there exists *precisely one* $(q)_\mathtt{a} \in Q$ such that $q \xrightarrow{\mathtt{a}} (q)_\mathtt{a}$.

The example automaton is deterministic.

Languages

The *language* of a state is the set of words leading to an accepting state.



Definition (Automaton language)

The *language* of $q \in Q$, denoted $L_A(q)$, is the smallest set satisfying

$$\frac{q \in F}{\epsilon \in L_A(q)} \qquad\qquad \frac{w \in L_A(q') \qquad q \xrightarrow{\mathtt{a}} q'}{\mathtt{a}w \in L_A(q)}$$

The *language of A*, denoted $L(A)$, is $\bigcup_{q \in I} L_A(q)$.

Simulation and bisimulation

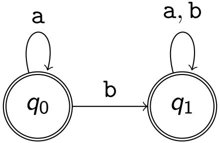
A *simulation* shows that one state can "mimic" another.

Definition (Simulation)

Let $A_i = \langle Q_i, \rightarrow_i, I_i, F_i \rangle$ for $i \in \{0, 1\}$. A *simulation* is a relation $R \subseteq Q_0 \times Q_1$ where

$$\frac{q_0 \; R \; q_1 \qquad q_0 \in F_0}{q_1 \in F_1} \qquad\qquad \frac{q_0 \; R \; q_1 \qquad q_0 \xrightarrow{\mathtt{a}} q_0'}{\exists q_1'.\; q_1 \xrightarrow{\mathtt{a}} q_1' \wedge q_0' \; R \; q_1'}$$

We call $q_0 \in Q_0$ *similar* to $q_1 \in Q_1$ when $q_0 \; R \; q_1$ for some simulation R, and $q_0 \in Q_0$ is *bisimilar* to $q_1 \in Q_1$ when q_0 is similar to q_1 and q_1 is similar to q_0.



Bisimularity versus language equivalence

Lemma

Let $A_i = \langle Q_i, \rightarrow, I_i, F_i \rangle$ for $i \in \{0, 1\}$, with $q_i \in Q_i$. The following hold:

1. *If q_0 is bisimilar to q_1, then $L_{A_0}(q_0) = L_{A_1}(q_1)$.*
2. *If $L_{A_0}(q_0) = L_{A_1}(q_1)$ and the A_i are deterministic, then q_0 is bisimilar to q_1.*

Proof of (1).

Let R be the simulation such that $q_0 \; R \; q_1$. Prove *by induction on $w \in \Sigma^*$* that *for all* $q_i \in Q_i$ we have that if $w \in L_{A_0}(q_0)$, then $w \in L_{A_1}(q_1)$.

Base: if $w = \epsilon$ and $w \in L_{A_0}(q_0)$, then $q_0 \in F_0$, so $q_1 \in F_1$, hence $w = \epsilon \in L_{A_1}(q_1)$.

Inductive step: if $\mathtt{a}w \in L_{A_0}(q_0)$, then $q_0 \xrightarrow{\mathtt{a}} q_0'$ and $w \in L_{A_0}(q_0')$. There exists $q_1' \in Q_1$ such that $q_1 \xrightarrow{\mathtt{a}} q_1'$ and $q_0' \; R \; q_1'$. By induction, $w \in L_{A_1}(q_1')$, so $\mathtt{a}w \in L_{A_1}(q_1)$. □

Bisimularity versus language equivalence

Lemma

Let $A_i = \langle Q_i, \rightarrow, I_i, F_i \rangle$ for $i \in \{0, 1\}$, with $q_i \in Q_i$. The following hold:

1. *If q_0 is bisimilar to q_1, then $L_{A_0}(q_0) = L_{A_1}(q_1)$.*
2. *If $L_{A_0}(q_0) = L_{A_1}(q_1)$ and the A_i are deterministic, then q_0 is bisimilar to q_1.*

Proof of (2).

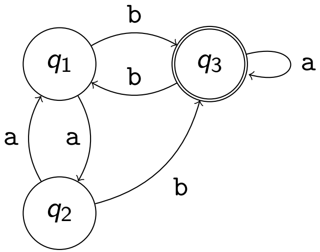
Let $R = \{\langle q_0', q_1' \rangle \in Q_0 \times Q_1 : L_{A_0}(q_0') = L_{A_1}(q_1')\}$. We claim that R is a simulation.

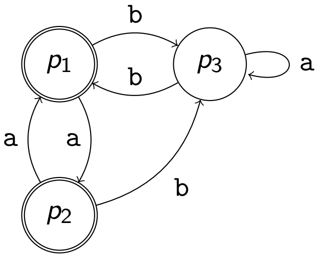
First rule: Let $q_0' \; R \; q_1'$ and $q_0' \in F_0$. Then $\epsilon \in L_{A_0}(q_0') = L_{A_1}(q_1')$, so $q_1' \in F_1$.

Second rule: Let $q_0' \; R \; q_1'$ and $q_0' \xrightarrow{\mathtt{a}} q_0''$. Because A_0 is deterministic, $q_0'' = (q_0')_{\mathtt{a}}$. We should find q_1'' such that $q_1' \xrightarrow{\mathtt{a}} q_1''$ and $(q_0')_{\mathtt{a}} \; R \; q_1''$. We choose $q_1'' = (q_1')_{\mathtt{a}}$. A quick proof shows that $L_{A_0}((q_0')_{\mathtt{a}}) = L_{A_1}((q_1')_{\mathtt{a}})$, and so $(q_0')_{\mathtt{a}} \; R \; (q_1')_{\mathtt{a}}$.

Analogously $R' = \{\langle q_1', q_0' \rangle \in Q_1 \times Q_0 : L_{A_1}(q_1') = L_{A_0}(q_0')\}$ is a simulation. □

Deciding bisimilarity





Deciding bisimilarity

```
Data: det. automata ⟨Q_i, F_i, δ_i⟩ with state q_i ∈ Q_i, for i ∈ {1, 2}.
Result: true if q_1 is similar to q_2, false otherwise.
R ← ∅;  T ← {⟨q_1, q_2⟩};
while T ≠ ∅ do
    pop ⟨q'_1, q'_2⟩ from T;
    if ⟨q'_1, q'_2⟩ ∉ R then
        if q'_1 ∈ F_1 ⟹ q'_2 ∈ F_2 then
            add ⟨q'_1, q'_2⟩ to R;
            add ⟨(q'_1)_a, (q'_2)_a⟩ to T for all a ∈ Σ;
        else
            return false;
return true;
```

Enforcing determinism

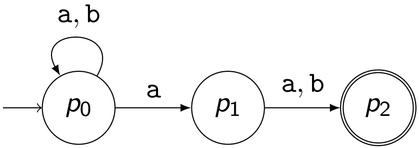
Definition (Powerset automata)

Let $A = \langle Q, \rightarrow, I, F \rangle$ be an automaton. The *powerset automaton* of A is the *deterministic* automaton $\langle 2^{Q}, \rightarrow', \{I\}, F' \rangle$, where

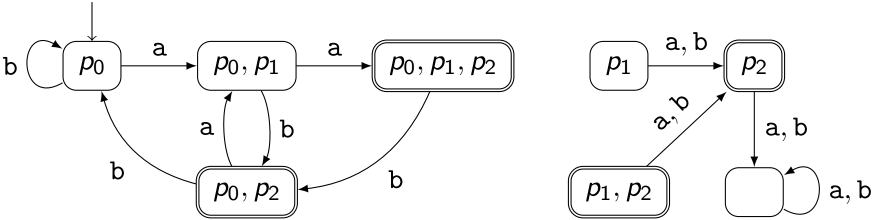
- $F' = \{S \subseteq Q : S \cap F \neq \emptyset\}$; and
- $\rightarrow'$ is the smallest relation where for all $S \subseteq Q$, we have

$$S \xrightarrow{\texttt{a}}{}' \{q' \in Q : \exists q \in S.\ q \xrightarrow{\texttt{a}} q'\}$$

Enforcing determinism



⇓



Enforcing determinism

Lemma

Let $A = \langle Q, \to, I, F \rangle$ be an automaton, and $A' = \langle 2^Q, \to', \{I\}, F' \rangle$ its powerset automaton. For all $S \subseteq Q$ we have $L_{A'}(S) = \bigcup_{q \in S} L_A(q)$. Thus, $L(A) = L(A')$.

Proof sketch.

Prove by induction on $w \in \Sigma^*$ that for all $S \subseteq Q$ we have $L_{A'}(S) = \bigcup_{q \in S} L_A(q)$.

Base: $\epsilon \in L_{A'}(S) \iff S \in F' \iff S \cap F \neq \emptyset \iff \epsilon \in \bigcup_{q \in S} L_A(q)$.

Inductive step: we derive as follows

$$\begin{aligned} \mathtt{a}w \in L_{A'}(S) &\iff w \in L_{A'}(\{q' \in Q : \exists q \in S . q \xrightarrow{\mathtt{a}} q'\}) \\ &\overset{IH}{\iff} \exists q' \in Q, q \in S.\ q \xrightarrow{\mathtt{a}} q' \wedge w \in L_A(q') \\ &\iff \exists q \in S.\ \mathtt{a}w \in L_A(q) \end{aligned}$$

□

The story so far

Language equivalence of q_0 and q_1 in automata A_0 and A_1 is decidable:

1. Make both automata deterministic using the powerset construction.
2. Decide positively precisely when $\{q_0\}$ is bisimilar to $\{q_1\}$.

But what about rational expressions?

Converting to automata

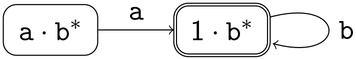
Theorem (Kleene '56)

One can construct a finite automaton A with a state q such that $L(q) = [\![e]\!]_{\mathbb{E}}$.

- *Many* different ways of proving this.
- Today's approach is due to Antimirov (1996) and Brzozowski (1964).

Antimirov's construction

- Basic idea: create an (infinite) automaton where states are expressions.
- Language of a state is intended to be the language of that expression.
- Some additional work necessary to tame this into an finite automaton.



Accepting expressions

If every state is an expression, which ones are accepting?

Definition (Accepting expressions)

We define $\mathbb{A}$ as the smallest subset of $\mathbb{E}$ satisfying the rules

$$\frac{}{1 \in \mathbb{A}} \qquad \frac{e \in \mathbb{A} \qquad f \in \mathbb{E}}{e + f, f + e \in \mathbb{A}} \qquad \frac{e, f \in \mathbb{A}}{e \cdot f \in \mathbb{A}} \qquad \frac{e \in \mathbb{E}}{e^* \in \mathbb{A}}$$

Idea: $\epsilon \in [\![e]\!]_{\mathbb{E}}$ if and only if $e \in \mathbb{A}$.

Transition structure

Definition (Transitions between expressions)

We define $\to_{\mathbb{E}} \subseteq \mathbb{E} \times \Sigma \times \mathbb{E}$ as the smallest relation satisfying

$$\frac{}{\mathtt{a} \xrightarrow{\mathtt{a}}_{\mathbb{E}} 1} \qquad \frac{e \xrightarrow{\mathtt{a}}_{\mathbb{E}} e'}{e + f \xrightarrow{\mathtt{a}}_{\mathbb{E}} e'} \qquad \frac{f \xrightarrow{\mathtt{a}}_{\mathbb{E}} f'}{e + f \xrightarrow{\mathtt{a}}_{\mathbb{E}} f'}$$

$$\frac{e \xrightarrow{\mathtt{a}}_{\mathbb{E}} e'}{e \cdot f \xrightarrow{\mathtt{a}}_{\mathbb{E}} e' \cdot f} \qquad \frac{e \in \mathbb{A} \qquad f \xrightarrow{\mathtt{a}}_{\mathbb{E}} f'}{e \cdot f \xrightarrow{\mathtt{a}}_{\mathbb{E}} f'} \qquad \frac{e \xrightarrow{\mathtt{a}}_{\mathbb{E}} e'}{e^* \xrightarrow{\mathtt{a}}_{\mathbb{E}} e' \cdot e^*}$$

Correctness

Theorem (Fundamental Theorem of Kleene Algebra)

Let $e \in \mathbb{E}$. The following holds:

$$e \equiv [e \in \mathbb{A}] + \sum_{e \xrightarrow{\mathtt{a}} e'} \mathtt{a} \cdot e'$$

Here $[e \in \mathbb{A}]$ is shorthand for 1 when $e \in \mathbb{A}$ and 0 otherwise.

Corollary

Let $A_e^\infty = \langle \mathbb{E}, \to_\mathbb{E}, \{e\}, \mathbb{A} \rangle$ be the (infinite) Antimirov automaton.

For $e \in \mathbb{E}$, it holds that $[\![e]\!]_\mathbb{E} = L(A_e^\infty)$.

Finiteness

The Antimirov automaton is infinite! Let's *restrict* it to a finite (relevant) set.

Definition

We define $\rho : \mathbb{E} \to 2^{\mathbb{E}}$ by induction, as follows.

$$\rho(0) = \rho(1) = \emptyset \qquad \rho(\mathtt{a}) = \{1\} \qquad \rho(e + f) = \rho(e) \cup \rho(f)$$

$$\rho(e \cdot f) = \{e' \cdot f : e' \in \rho(e)\} \cup \rho(f) \qquad \rho(e^*) = \{e' \cdot e^* : e' \in \rho(e)\}$$

We write $\hat{\rho}(e)$ for $\rho(e) \cup \{e\}$.

Lemma

If $e' \in \hat{\rho}(e)$ and $e' \xrightarrow{\mathtt{a}} e''$, then $e'' \in \hat{\rho}(e)$.

Corollary

If $A_e = \langle \hat{\rho}(e), \to_{\mathbb{E}} \cap\, \hat{\rho}(e)^2, \{e\}, \mathbb{A} \cap \hat{\rho}(e) \rangle$, then $L(A_e) = L(A_e^\infty)$.

The upshot

Language equivalence of rational expressions e and f is decidable.

1. Convert both expressions to their (finite) Antimirov automata.
2. Decide whether e (in A_e) is language equivalent to f (in A_f).

Other thoughts

- Converting an expression (program) to a machine is a kind of *compilation*.
- Automata in general are a great tool for decidability results.
- There exist methods to make bisimulation checking more efficient.
- Brzozowski's approach has echoes in *structural operational semantics*.

Next lecture

- Converse construction: from automata to expressions.
- Matrices of rational expressions as a powerful tool.