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.

![Diagram of an automaton]

**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 \( \langle q, a, q' \rangle \in \rightarrow \), we write \( q \xrightarrow{a} q' \).
A *deterministic* automaton has no “ambiguity” in the transitions.

![Deterministic Automaton Diagram](image)

**Definition (Determinism)**

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

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

\[
\begin{align*}
q \in F \\
\epsilon \in L_A(q) \\
w \in L_A(q') \quad q \xrightarrow{a} q' \\
aw \in L_A(q)
\end{align*}
\]

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

$$
\frac{q_0 R q_1 \quad q_0 \in F_0}{q_1 \in F_1}
$$

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 $q_0$. 

![Diagram](attachment:image.png)
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 $q_0 \sim_R q_1$. 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 $aw \in L_{A_0}(q_0)$, then $q_0 \xrightarrow{a} q'_0$ and $w \in L_{A_0}(q'_0)$. There exists $q'_1 \in Q_1$ such that $q_1 \xrightarrow{a} q'_1$ and $q'_0 \sim_R q'_1$. By induction, $w \in L_{A_1}(q'_1)$, so $aw \in L_{A_1}(q_1)$. \qed
Bisimilarity 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 $q'_0 \xrightarrow{a} q''_0$. Because $A_0$ is deterministic, $q''_0 = (q'_0)_a$. We should find $q''_1$ such that $q'_1 \xrightarrow{a} q''_1$ and $(q'_0)_a R q''_1$. We choose $q''_1 = (q'_1)_a$. A quick proof shows that $L_{A_0}((q'_0)_a) = L_{A_1}((q'_1)_a)$, and so $(q'_0)_a R (q'_1)_a$.

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. \qed
Deciding bisimilarity
Deciding bisimilarity

Data: det. automata $\langle Q_i, F_i, \delta_i \rangle$ with state $q_i \in Q_i$, for $i \in \{1, 2\}$.
Result: **true** if $q_1$ is similar to $q_2$, **false** otherwise.

$R \leftarrow \emptyset$; $T \leftarrow \{\langle q_1, q_2 \rangle\}$;
while $T \neq \emptyset$ do
    pop $\langle q'_1, q'_2 \rangle$ from $T$;
    if $\langle q'_1, q'_2 \rangle \notin R$ then
        if $q'_1 \in F_1 \implies q'_2 \in F_2$ then
            add $\langle q'_1, q'_2 \rangle$ to $R$;
            add $\langle (q'_1)_a, (q'_2)_a \rangle$ to $T$ for all $a \in \Sigma$;
        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
  $$S \xrightarrow{\text{a}}' \{ q' \in Q : \exists q \in S. \ q \xrightarrow{\text{a}} q' \}$$
Enforcing determinism
Enforcing determinism

Lemma
Let $A = \langle Q, \rightarrow, I, F \rangle$ be an automaton, and $A' = \langle 2^Q, \rightarrow', \{ I \}, F' \rangle$ 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
\[
aw \in L_{A'}(S) \iff w \in L_{A'}(\{ q' \in Q : \exists q \in S. q \xrightarrow{a} q' \}) \\
\text{IH} \iff \exists q' \in Q, q \in S. q \xrightarrow{a} q' \land w \in L_A(q') \\
\iff \exists q \in S. aw \in L_A(q)
\]
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 $L(q) = [e]_E$.

- 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 a 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

\[
\begin{align*}
1 & \in \mathbb{A} \\
e + f, f + e & \in \mathbb{A} \\
e, f & \in \mathbb{A} \\
e & \in \mathbb{E} \\
e^* & \in \mathbb{A}
\end{align*}
\]

Idea: $\epsilon \in \llbracket e \rrbracket_\mathbb{E}$ if and only if $e \in \mathbb{A}$.
**Definition (Transitions between expressions)**

We define $\rightarrow_E \subseteq E \times \Sigma \times E$ as the smallest relation satisfying

- $a \stackrel{a}{\rightarrow}_E 1$
- $e \rightarrow_E e'$
- $f \rightarrow_E f'$
- $e + f \rightarrow_E e'$
- $e \cdot f \rightarrow_E e' \cdot f$
- $e \in A$
- $f \rightarrow_E f'$
- $e \cdot f \rightarrow_E f'$
- $e^* \rightarrow_E e' \cdot e^*$
Correctness

**Theorem (Fundamental Theorem of Kleene Algebra)**

Let $e \in E$. The following holds:

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

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

**Corollary**

Let $A^\infty_e = \langle E, \rightarrow_E, \{e\}, A \rangle$ be the (infinite) Antimirov automaton.

For $e \in E$, it holds that $\llbracket e \rrbracket_E = L(A^\infty_e)$. 
Finiteness

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

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

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

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

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

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

Corollary
If $A_e = \langle \hat{\rho}(e), \rightarrow_{\mathbb{E}} \cap \hat{\rho}(e)^2, \{e\}, A \cap \hat{\rho}(e) \rangle$, 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.