Finite automata

Dynamic Logic – Part 3

Igor Sedlár

Institute of Computer Science of the Czech Academy of Sciences

Faculty of Arts, Charles University
Fall Semester 2023-24

Overview

- We introduce (deterministic) finite automata, an "operational" counterpart of regular expressions and Kleene algebras; we show that every automaton is equivalent (in the sense of recognizing the same language) to a deterministic one
- We discuss bisimulation, a decidable relation between automata that implies (and, in the case of deterministic automata, is implied by) equivalence
- We prove (one half of) Kleene's theorem, stating that a language is regular iff it is recognized by a finite automaton; together with the other results, this implies that equivalence of regular expressions over arbitrary Kleene algebras is decidable
- We modify the notion of a finite automaton to match guarded languages and Kleene algebra with tests; we generalize the results on automata leading to decidability of equivalence to the guarded setting

Finite automata – 1

Definition 1

Let Σ be a finite alphabet. A finite automaton (for Σ) is $A = \langle Q, \delta, I, F \rangle$ where

- Q *is a finite set* ...states
- $\delta : Q \times \Sigma \to 2^Q$ ...transition relation
- $I, F \subseteq Q$ ...initial and final states

An automaton is deterministic if (i) $\delta(q, \mathtt{a})$ is a singleton for all $q \in Q$, $\mathtt{a} \in \Sigma$, and (ii) I is a singleton.

We will often write $q \xrightarrow{\mathtt{a}} q'$ for $q' \in \delta(q, \mathtt{a})$.

Definition 2

The language of a $q \in Q$ of A, or $L_A(q)$, is the smallest subset of Σ^* such that

- *if $q \in F$, then $\epsilon \in L_A(q)$*
- *if $w \in L_A(q')$ and $q \xrightarrow{\mathtt{a}} q'$, then $\mathtt{a}w \in L_A(q)$*

The language of A (language recognized by A) is $L(A) := \bigcup_{q \in I} L_A(q)$.

Finite automata – 2

A path in an automaton is a sequence of the form

$$q_1 \mathtt{a}_1 q_2 \dots \mathtt{a}_{n-1} q_n$$

where (i) $n \geq 1$, $q_i \in Q$ and $\mathtt{a}_j \in \Sigma$, and (ii) $q_i \xrightarrow{\mathtt{a}_i} q_{i+1}$ for all $i < n$. We will often denote paths as $q_1 \xrightarrow{\mathtt{a}_1} q_2 \dots \xrightarrow{\mathtt{a}_{n-1}} q_n$.

We define the accessibility relation $\delta^* : Q \times \Sigma^* \to 2^Q$ by induction on the length of $w \in \Sigma^*$ ($q \xrightarrow{w} q'$ means $q' \in \delta^*(q, w)$):

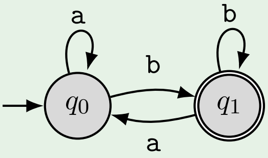
- $q \xrightarrow{\epsilon} q'$ iff $q = q'$
- $q \xrightarrow{\mathtt{a}w} q'$ iff there is $p \in Q$ such that $q \xrightarrow{\mathtt{a}} p$ and $p \xrightarrow{w} q'$.

Exercise 1

Prove that $w \in L_A(q)$ iff there is $p \in F$ of A such that $q \xrightarrow{w} p$.

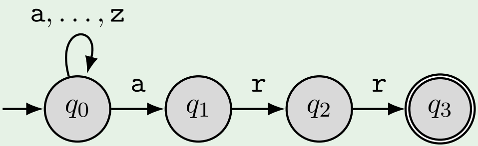
Finite automata – 3

Example (Automaton A_1)



- in $L(A_1)$: b, bab, baab, baabb, ...
- not in $L(A_1)$: a, aba, ...

Example (Automaton A_2)



- in $L(A_2)$: arr, goldarr, ...

Bisimulation – 1

Let $A_i = \langle Q_i, \delta_i, I_i, F_i \rangle$ be a finite automaton for the same Σ and $i \in \{1, 2\}$.

Definition 3

A simulation of A_1 by A_2 is a relation $R \subseteq Q_1 \times Q_2$ such that, for all $q_1 \in Q_1$ and $q_2 \in Q_2$, $q_1 \; R \; q_2$ implies

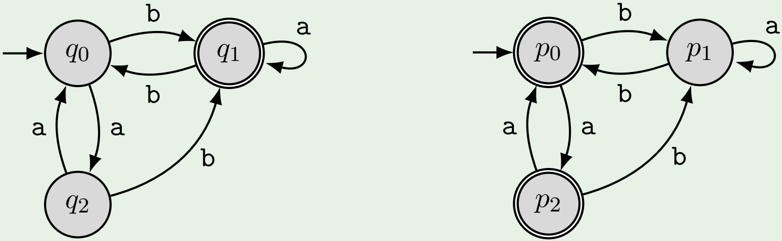
1. $q_1 \in F_1$ *only if* $q_2 \in F_2$
2. $q_1 \xrightarrow{\mathtt{a}}_1 q_1'$ *only if there is* q_2' *such that* $q_2 \xrightarrow{\mathtt{a}}_2 q_2'$ *and* $q_1' R q_2'$.

A bisimulation between A_1 and A_2 is a simulation of A_1 by A_2 such that its converse is a simulation of A_2 by A_1.

A state q_1 of A_1 is (bi)similar to a state q_2 of A_2 iff there is a (bi)simulation R such that $q_1 \; R \; q_2$ (notation: $q_1 \underline{\rightarrow} q_2$ and $q_1 \underline{\leftrightarrow} q_2$). A_1 is (bi)similar to A_2 (notation $A_1 \underline{\rightarrow} A_2$, resp. $A_1 \underline{\leftrightarrow} A_2$) iff there is a (bi)simulation R "defined on" each $q \in I_1$ (each $q_1 \in I_2$ and $Q_2 \in I_2$).

Bisimulation – 2

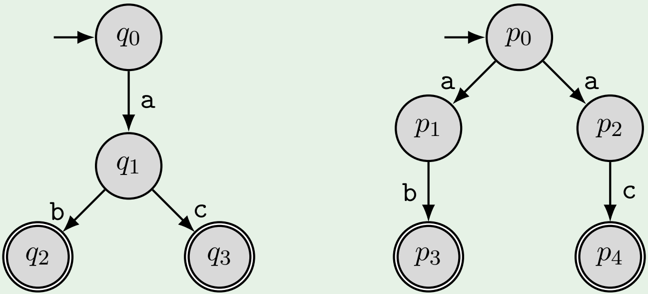
Example (Bisimilar automata)



$$R = \{\langle q_0, p_1\rangle, \langle q_2, p_1\rangle, \langle q_1, p_0\rangle, \langle q_1, p_2\rangle\}$$

Bisimulation – 3

Non-bisimilar automata



Bisimulation – 4

Proposition 1

Let A_1, A_2 be two finite automata and $q_1 \in Q_1$, $q_2 \in Q_2$. Then:

1. $q_1 \rightarrow q_2$ *implies* $L(q_1) \subseteq L(q_2)$
2. *if A_2 is deterministic, then* $L(q_1) \subseteq L(q_2)$ *implies* $q_1 \rightarrow q_2$.

Proof (sketch). (1.) Assume $q_1 \rightarrow q_2$ and prove that $w \in L(q_1) \implies w \in L(q_2)$ by induction on the length of $w \in \Sigma^*$. (2.) Define $R = \{\langle p_1, p_2 \rangle \in Q_1 \times Q_2 \mid L(p_1) \subseteq L(p_2)\}$ and show that R is a simulation (use Exercise 2). □

Corollary

If A_1, A_2 are deterministic, then $q_1 \leftrightarrow q_2$ iff $L(q_1) = L(q_2)$.

Fact: There is a polynomial-time algorithm for checking if $q_1 \leftrightarrow q_2$. (See (Kappé, 2023), lecture 3.)

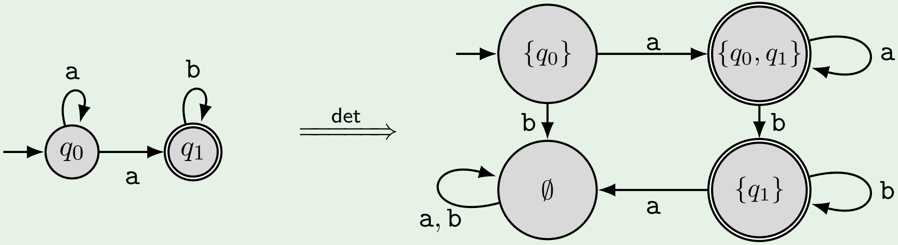
Determinization – 1

Definition 4

Let $A = \langle Q, \delta, I, F \rangle$ be a finite automaton. The determinization of A is the deterministic automaton A^{det} such that

- $Q^{\mathsf{det}} = 2^Q$
- $\delta^{\mathsf{det}}(X, \mathtt{a}) = \{q' \mid q \xrightarrow{\mathtt{a}} q'$ *for some* $q \in X\}$ *for all* $X \subseteq Q$
- $I^{\mathsf{det}} = \{I\}$
- $F^{\mathsf{det}} = \{X \subseteq Q \mid X \cap F \neq \emptyset\}$

Example



Determinization – 2

Proposition 2

For all A, $L(A) = L(A^{\text{det}})$.

Proof (sketch). $L(A) \subseteq L(A^{\text{det}})$ since $A \underline{\rightarrow} A^{\text{det}}$ for $R = \{\langle q, X\rangle \mid q \in X\}$. Converse inclusion: prove that $w \in L^{\text{det}}(X) \implies w \in L(X)$ by induction on the length of $w \in \Sigma^*$ (where $L^{\text{det}}(X)$ is $L_{A^{\text{det}}}(X)$ and $L(X) = \bigcup_{q \in X} L_A(q)$) □

Corollary

Hence, there is an algorithm for deciding $L(A) = L(B)$ for arbitrary automata A, B. (Its running time may be exponential in the size of A, B.)

Kleene's Theorem – 1

Theorem 1 (Kleene 1956)

A language $L \subseteq \Sigma^*$ is regular iff there is a deterministic finite automaton A such that $L = L(A)$.

Proof (sketch). (i) Regular expression $e \longrightarrow$ "Antimirov automaton" A_e such that $L(A_e) = [\![e]\!]$ (see below). (ii) E.g. solving systems of equations (Kappé, 2023) or state elimination (Hopcroft et al., 2007; Sipser, 2013). □

Kleene's Theorem – 2

Definition 5

The set of accepting expressions $\mathbb{A}$ *is the smallest subset of* $\mathbb{E}$ *such that*

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

Note that $e \in \mathbb{A}$ iff $\epsilon \in [\![e]\!]$. (Exercise 3.)

Definition 6

Expression accessibility: We define $\rightarrow_{\mathbb{E}} \subseteq \mathbb{E} \times \Sigma \times \mathbb{E}$ *as the smallest relation satisfying*

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

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

Kleene's Theorem – 3

Definition 7

<u>Reachable expressions:</u> for each $e \in \mathbb{E}$, the set $\rho(e) \subseteq \mathbb{E}$ is defined as follows:

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

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

Note: $\rho(e)$ is finite for all $e \in \mathbb{E}$.

Lemma 1

The following hold for all $e \in \mathbb{E}$:

1. *If $e \xrightarrow{\mathtt{a}}_{\mathbb{E}} e'$, then $e' \in \rho(e)$.*
2. *If $e' \in \rho(e)$ and $e' \xrightarrow{\mathtt{a}}_{\mathbb{E}} e''$, then $e'' \in \rho(e)$.*

Proof (sketch). Induction on the complexity of e. See (Kappé, 2023), lecture 3. □

Kleene's Theorem – 4

Definition 8

The Antimirov automaton for e is

$$A_e = \langle \hat{\rho}(e), \rightarrow_{\mathbb{E}}, \{e\}, \mathbb{A} \cap \hat{\rho}(e) \rangle \, ,$$

where $\hat{\rho}(e) = \rho(e) \cup \{e\}$.

Kleene's Theorem – 5

The Iverson bracket: $[\Phi(e)] = 1$ if e satisfies the predicate Φ and $= 0$ otherwise.

Theorem 2 (The fundamental theorem)

For all $e \in \mathbb{E}$*:*

$$e \equiv [e \in \mathbb{A}] + \sum\{\mathtt{a} \cdot e' \mid e \xrightarrow{\mathtt{a}}_{\mathbb{E}} e'\}$$

Proof (sketch). Induction on e. The base case: $\mathtt{a} \equiv 0 + \mathtt{a} \cdot 1$. Induction step for $e \cdot f$:

$$\begin{aligned} e \cdot f &\equiv [e \in \mathbb{A}] \cdot [f \in \mathbb{A}] + [e \in \mathbb{A}] \cdot \sum_{f \xrightarrow{\mathtt{a}} f'} \mathtt{a} \cdot f' + \sum_{e \xrightarrow{\mathtt{a}} e'} \mathtt{a} \cdot e' \cdot f \\ &\equiv [e \cdot f \in \mathbb{A}] + \sum_{e \cdot f \xrightarrow{\mathtt{a}} g} \mathtt{a} \cdot g \end{aligned}$$

(Note that $\sum \delta(e \cdot f, \mathtt{a}) \equiv \sum\{e' \cdot f \mid e \xrightarrow{\mathtt{a}} e'\} + [e \in \mathbb{A}] \cdot \sum\{f' \mid f \xrightarrow{\mathtt{a}} f'\}$.) (Exercise 4.)

Corollary

For all $e \in \mathbb{E}$*:* $L(A_e) = [\![e]\!]$.

Proof (sketch). $w \in L(A_e)$ iff $w \in [\![e]\!]$ by induction on the length of w. (Exercise 5.)

Kleene's Theorem – 6

Compiling regular expressions:

$$\begin{aligned} e \equiv f &\iff [\![e]\!] = [\![f]\!] \\ &\iff L(A_e) = L(A_f) \\ &\iff L(A_e^{\mathsf{det}}) = L(A_f^{\mathsf{det}}) \\ &\iff (A_e^{\mathsf{det}}, e) \underline{\leftrightarrow} (A_f^{\mathsf{det}}, f) \end{aligned}$$

To decide if $e \equiv f$:

1. construct A_e and A_f,
2. determinize to A_e^{det} and A_f^{det},
3. check if $(A_e^{\mathsf{det}}, e) \underline{\leftrightarrow} (A_f^{\mathsf{det}}, f)$.

Guarded automata – 1

Recall: At is the set of atoms over Π; guarded strings over Σ, Π are words in $(At \cdot \Sigma)^* \cdot At$.

Definition 9

An guarded automaton (over Σ, Π) is $A = \langle Q, \delta, I, F \rangle$ where

- $\delta : Q \times At \times \Sigma \to 2^Q$ *...guarded transition relation*
- $I \subseteq Q$ *...initial states*
- $F : Q \to 2^{At}$ *...guards of finality*

A is deterministic iff I and the range of δ are singletons.

We often write $q \xrightarrow{S|\mathsf{a}} q'$ for $q' \in \delta(q, S, \mathsf{a})$ and $S \in F(q)$ for $F(q, S) = 1$.

Note that "ordinary" automata are a special case for $\Pi = \emptyset$. (In that case, $At = \{\epsilon\}$.)

Guarded automata – 2

Definition 10

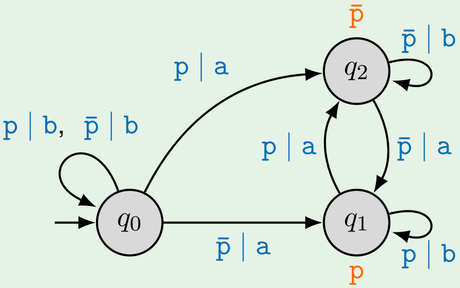
The language of $q \in Q$ of A, or $L_A(q)$, is the smallest subset of GS such that

- *if $S \in F(q)$, then $S \in L_A(q)$,*
- *if $w \in L_A(q')$ and $q \xrightarrow{S|\mathtt{a}} q'$, then $S\mathtt{a}w \in L_A(q)$.*

The language of A (language recognized by A) is $L(A) := \bigcup_{q \in I} L_A(q)$.

Guarded automata – 3

Example



Accepted (in $L(q_0)$):

- $\bar{\mathtt{p}}\mathtt{ap}$, $\mathtt{pa}\bar{\mathtt{p}}$
- $\bar{\mathtt{p}}\mathtt{apa}\bar{\mathtt{p}}$, $\bar{\mathtt{p}}\mathtt{apbpa}\bar{\mathtt{p}}$, ...

Not accepted:

- $\mathtt{pbp}$, $\bar{\mathtt{p}}\mathtt{b}\bar{\mathtt{p}}$, $\mathtt{pb}\bar{\mathtt{p}}$, $\bar{\mathtt{p}}\mathtt{bp}$
- $\mathtt{pa}\bar{\mathtt{p}}\mathtt{b}\bar{\mathtt{p}}\mathtt{a}\bar{\mathtt{p}}$, $\mathtt{pa}\bar{\mathtt{p}}\mathtt{bp}$, ...

Guarded bisimulation – 1

Definition 11

A guarded simulation of A_1 *by* A_2 *is a relation* $R \subseteq Q_1 \times Q_2$ *such that* $q_1 \; R \; q_2$ *entails*

1. $S \in F_1(q_1)$ *only if* $S \in F_2(q_2)$
2. $q_1 \xrightarrow{S|\mathtt{a}}_1 q_1'$ *only if there is* q_2' *such that* $q_2 \xrightarrow{S|\mathtt{a}}_2 q_2'$ *and* $q_1' \; R \; q_2'$.

A guarded bisimulation between A_1 *and* A_2 *is a simulation of* A_1 *by* A_2 *such that its converse is a guarded simulation of* A_2 *by* A_1.

Guarded (bi) similarity of states (automata) is defined (and denoted) similarly as before.

Guarded bisimulation – 2

Proposition 3

Let A_1, A_2 *be two guarded automata and* $q_1 \in Q_1$, $q_2 \in Q_2$. *Then:*

1. $q_1 \rightarrowtail q_2$ *implies* $L(q_1) \subseteq L(q_2)$
2. *if* A_2 *is deterministic, then* $L(q_1) \subseteq L(q_2)$ *implies* $q_1 \rightarrowtail q_2$.

Proof (sketch). Similar as the proof of Prop. 1; see Exercise 7. □

Corollary

If A_1, A_2 *are deterministic, then* $q_1 \leftrightarroweq q_2$ *iff* $L(q_1) = L(q_2)$.

Fact: As before, there is a polynomial-time algorithm for checking if $q_1 \leftrightarroweq q_2$.

Guarded determinization – 1

Definition 12

Let $A = \langle Q, \delta, I, F \rangle$ *be a guarded automaton. The* *determinization* *of* A *is the deterministic guarded automaton* A^{det} *such that*

- $Q^{\mathsf{det}} = 2^Q$
- $\delta^{\mathsf{det}}(X, S, \mathtt{a}) = \{q' \mid q \xrightarrow{S|\mathtt{a}} q'$ *for some* $q \in X\}$ *for all* $X \subseteq Q$
- $I^{\mathsf{det}} = \{I\}$
- $F^{\mathsf{det}}(X) = \bigcup_{q \in X} F(q)$ *for all* $X \subseteq Q$

Guarded determinization – 2

Proposition 4

For all guarded A, $L(A) = L(A^{\mathsf{det}})$.

Proof (sketch). Similar to the proof of Prop. 2. See Exercise 8. □

Corollary

Hence, there is an algorithm for deciding $L(A) = L(B)$ for arbitrary guarded automata A, B. (Its running time may be exponential in the size of A, B.)

Kleene's Theorem for guarded automata – 1

Theorem 3

A guarded language $L \subseteq GS$ is regular iff there is a deterministic guarded automaton A such that $L = L(A)$.

Kleene's Theorem for guarded automata – 2

In this section, let $\mathbb{E}$ be $\mathbb{E}(\Sigma, \Pi)$ for some fixed Σ and Π, and let $At = At(\Pi)$. For atom S and Boolean formula b, we write $S \vDash b$ if S satisfies b (in the obvious sense).

Definition 13

Let the accepting atoms function $\mathbb{A} : \mathbb{E} \to 2^{At}$ be defined as follows:

$$\mathbb{A}(\mathtt{a}) = \emptyset \qquad \mathbb{A}(b) = \{S \mid S \vDash b\} \qquad \mathbb{A}(e + f) = \mathbb{A}(e) \cup \mathbb{A}(f)$$

$$\mathbb{A}(e \cdot f) = \mathbb{A}(e) \cap \mathbb{A}(f) \qquad \mathbb{A}(e^*) = At$$

Note that $\mathbb{A}(e) = [\![e]\!] \cap At$.

Kleene's Theorem for guarded automata – 3

Definition 14

Expression accessibility: We define $\rightarrow_{\mathbb{E}} \subseteq \mathbb{E} \times At \times \Sigma \times \mathbb{E}$ *as the smallest relation satisfying*

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

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

Kleene's Theorem for guarded automata – 4

Definition 15

Reachable expressions: for each $e \in \mathbb{E}$, the set $\rho(e) \subseteq \mathbb{E}$ is defined as follows:

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

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

Note: $\rho(e)$ is finite for all $e \in \mathbb{E}$.

Lemma 2

The following claims hold for all $e \in \mathbb{E}$:

1. *If $e \xrightarrow{S|\mathtt{a}}_{\mathbb{E}} e'$, then $e' \in \rho(e)$.*
2. *If $e' \in \rho(e)$ and $e' \xrightarrow{S|\mathtt{a}}_{\mathbb{E}} e''$, then $e'' \in \rho(e)$.*

Proof (sketch). Induction on the complexity of e, similar to the proof of Lemma 1. (Exercise 9.)

Kleene's Theorem for guarded automata – 5

Definition 16

The Antimirov automaton for e *is*

$$A_e = \langle \hat{\rho}(e), \rightarrow_{\mathbb{E}}, \{e\}, \mathbb{A}|_{\hat{\rho}(e)} \rangle ,$$

where $\hat{\rho}(e) = \rho(e) \cup \{e\}$ *and* $\mathbb{A}|_{\hat{\rho}(e)}$ *is the restriction of* $\mathbb{A}$ *to* $\hat{\rho}(e)$*.*

Theorem 4 (The guarded fundamental theorem)

For all $e \in \mathbb{E}$*:*

$$e \equiv \sum \mathbb{A}(e) + \sum \{\mathtt{a} \cdot e' \mid e \xrightarrow{\mathtt{a}}_{\mathbb{E}} e'\}$$

Corollary

For all $e \in \mathbb{E}$*:* $L(A_e) = [\![e]\!]$*.*

Exercises

2. Prove that (i) $\epsilon \in L_A(q)$ iff $q \in F$ of A, and (ii) if A is deterministic, then $\mathtt{a}w \in L_A(q)$ iff $q \in L_A(\delta(q, \mathtt{a}))$.

3. Prove that $e \in \mathbb{A}$ iff $\epsilon \in [\![e]\!]$. (Hint: Prove that $e \leqq f$ and $e \in \mathbb{A}$ only if there is $f' \equiv f$ such that $f \in \mathbb{A}$.)

4. ★ Finish the proof of Theorem 2. (Hint for the case *: Use $e^* \equiv 1 + e \cdot e^*$ and reason by cases according to whether $e \in \mathbb{A}$ or not.)

5. Prove the corollary to Theorem 2.

6. Define a suitable notion of accessibility relation for guarded automata and prove that $w \in L_A(q)$ iff there is p such that $\mathsf{last}(w) \in F(p)$ and $q \xrightarrow{w} p$. (See Exercise 1.)

7. Prove Proposition 3.

8. Prove Proposition 4.

9. Prove Lemma 2.

10. ★ Prove Theorem 4 and its corollary.

Notes

- Finite automata go back to Kleene (1956) and Rabin and Scott (1959).
- Excellent introductions to automata theory are (Hopcroft et al., 2007), (Hopcroft and Ullman, 1979) and (Sakarovitch, 2009).
- Guarded automata were introduced (in a different form) by Kozen (2003).
- These notes are largely based on Lecture 3 in (Kappé, 2023).

References

- J. Hopcroft, R. Motwani, and J. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Pearson Education, 3rd edition, 2007.
- J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley Publishing Company, 1979.
- T. Kappé. Elements of Kleene algebra. Course notes, ESSLLI 2023, 2023. URL: `https://tobias.kap.pe/esslli/`.
- S. C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 3 – 41. Princeton University Press, 1956. `doi:10.1515/9781400882618-002`.
- D. Kozen. Automata on guarded strings and applications. *Matématica Contemporânea*, 24:117–139, 2003. `doi:10.21711/231766362003/rmc246`.
- M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959. `doi:10.1147/rd.32.0114`.
- J. Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, Cambridge, 2009.
- M. Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 3rd edition edition, 2013.