Logische Complexiteit

VIII : Church–Turing Stelling

Jeroen Goudsmit
Universiteit Utrecht
dinsdag 6 maart 2012

Overzicht

Herhaling

Variaties van Turing machines

Church–Turing These

Ambigu?

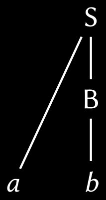
$$\mathrm{S} \rightarrow a\,\mathrm{A} \mid a\,\mathrm{B} \qquad \mathrm{A} \rightarrow b \qquad \mathrm{B} \rightarrow b$$

Ambigu?

$$S \rightarrow a\,A \mid a\,B \qquad A \rightarrow b \qquad B \rightarrow b$$

Ambigu?

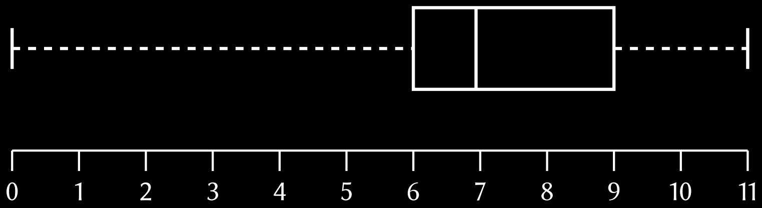
$$S \to a\,A \mid a\,B \qquad A \to b \qquad B \to b$$



Inleveropgave 3.1.d

$$
\begin{array}{rcl}
\mathsf{S} & \rightarrow & 0\ \mathsf{A} \\
\mathsf{S} & \rightarrow & 1\ \mathsf{A} \\
\mathsf{S} & \rightarrow & (\ \mathsf{S}\ ) \\
\mathsf{A} & \rightarrow & +\ \mathsf{S} \\
\mathsf{A} & \rightarrow & \cdot\ \mathsf{S} \\
\mathsf{A} & \rightarrow & 0\ \mathsf{A} \\
\mathsf{A} & \rightarrow & 1\ \mathsf{A} \\
\mathsf{A} & \rightarrow & \varepsilon
\end{array}
$$

Inleveropgave 3



Deeltoets 1.4: Invullen

Lemma
Als $\mathcal{L}$ context-vrij over Σ_1 en $\mathcal{L}_x$ context-vrij over Σ_2 voor $x \in \Sigma_1$ dan is het volgende context-vrij

$$\left\{ v_1 \dots v_n \;\middle|\; \begin{array}{l} n \in \mathbb{N} \text{ en } w_1 \dots w_n \in \mathcal{L} \\ \text{zodat } v_1 \in \mathcal{L}_{w_1}, \dots, v_n \in \mathcal{L}_{w_n} \end{array} \right\}.$$

Deeltoets 1.4: Invullen

$\langle\, V, \Sigma_1, R, S \,\rangle$ in Chomsky normaalvorm

$\langle\, V_x, \Sigma_2, R_x, S_x \,\rangle$

Lemma
Als $\mathcal{L}$ context-vrij over Σ_1 en $\mathcal{L}_x$ context-vrij over Σ_2 voor $x \in \Sigma_1$ dan is het volgende context-vrij

$$\left\{\; v_1 \ldots v_n \;\middle|\; \begin{array}{l} n \in \mathbb{N} \text{ en } w_1 \ldots w_n \in \mathcal{L} \\ \text{zodat } v_1 \in \mathcal{L}_{w_1}, \ldots, v_n \in \mathcal{L}_{w_n} \end{array} \;\right\}.$$

Bewijs.

∎

Deeltoets 1.4: Invullen

$\langle\, V, \Sigma_1, R, S \,\rangle$ in Chomsky normaalvorm

$\langle\, V_x, \Sigma_2, R_x, S_x \,\rangle$

Lemma
Als $\mathcal{L}$ context-vrij over Σ_1 en $\mathcal{L}_x$ context-vrij over Σ_2 voor $x \in \Sigma_1$ dan is het volgende context-vrij

$$\left\{\, v_1 \ldots v_n \;\middle|\; \begin{array}{l} n \in \mathbb{N} \text{ en } w_1 \ldots w_n \in \mathcal{L} \\ \text{zodat } v_1 \in \mathcal{L}_{w_1}, \ldots, v_n \in \mathcal{L}_{w_n} \end{array} \,\right\}.$$

Bewijs.

Maak CFG $\langle\, V, \Sigma_2, R, S \,\rangle$.

∎

Deeltoets 1.4: Invullen

$\langle\, V, \Sigma_1, R, S \,\rangle$ in Chomsky normaalvorm

$\langle\, V_x, \Sigma_2, R_x, S_x \,\rangle$

Lemma

Als $\mathcal{L}$ context-vrij over Σ_1 en $\mathcal{L}_x$ context-vrij over Σ_2 voor $x \in \Sigma_1$ dan is het volgende context-vrij

$$\left\{\, v_1 \ldots v_n \;\middle|\; \begin{array}{l} n \in \mathbb{N} \text{ en } w_1 \ldots w_n \in \mathcal{L} \\ \text{zodat } v_1 \in \mathcal{L}_{w_1}, \ldots, v_n \in \mathcal{L}_{w_n} \end{array} \,\right\}.$$

Bewijs.

Maak CFG $\langle\, V, \Sigma_2, R, S \,\rangle$. Zet $V := V \cup \bigcup_{x \in \Sigma_2} V_x$ en $S = S$.

∎

Deeltoets 1.4: Invullen

$\langle\, V, \Sigma_1, R, S \,\rangle$ in Chomsky normaalvorm

$\langle\, V_x, \Sigma_2, R_x, S_x \,\rangle$

Lemma

Als $\mathcal{L}$ context-vrij over Σ_1 en $\mathcal{L}_x$ context-vrij over Σ_2 voor $x \in \Sigma_1$ dan is het volgende context-vrij

$$\left\{ v_1 \ldots v_n \;\middle|\; \begin{array}{l} n \in \mathbb{N} \text{ en } w_1 \ldots w_n \in \mathcal{L} \\ \text{zodat } v_1 \in \mathcal{L}_{w_1}, \ldots, v_n \in \mathcal{L}_{w_n} \end{array} \right\}.$$

Bewijs.

Maak CFG $\langle\, V, \Sigma_2, R, S \,\rangle$. Zet $V := V \cup \bigcup_{x \in \Sigma_2} V_x$ en $S = S$.
Per regel $A \to x$, maak regel $A \to S_x$ en neem de andere regels over.

$\blacksquare$

Deeltoets 1.4: Invullen

$\langle\, V, \Sigma_1, R, S \,\rangle$ in Chomsky normaalvorm

$\langle\, V_x, \Sigma_2, R_x, S_x \,\rangle$

Lemma

Als $\mathcal{L}$ context-vrij over Σ_1 en $\mathcal{L}_x$ context-vrij over Σ_2 voor $x \in \Sigma_1$ dan is het volgende context-vrij

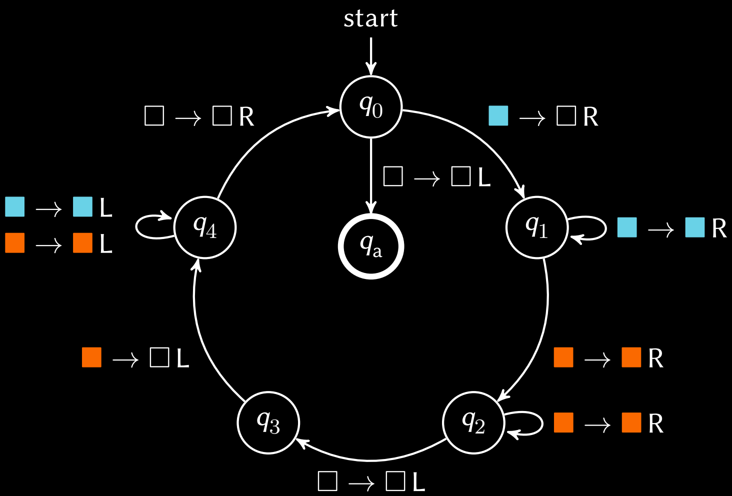
$$\left\{\; v_1 \ldots v_n \;\middle|\; \begin{array}{l} n \in \mathbb{N} \text{ en } w_1 \ldots w_n \in \mathcal{L} \\ \text{zodat } v_1 \in \mathcal{L}_{w_1}, \ldots, v_n \in \mathcal{L}_{w_n} \end{array} \;\right\}.$$

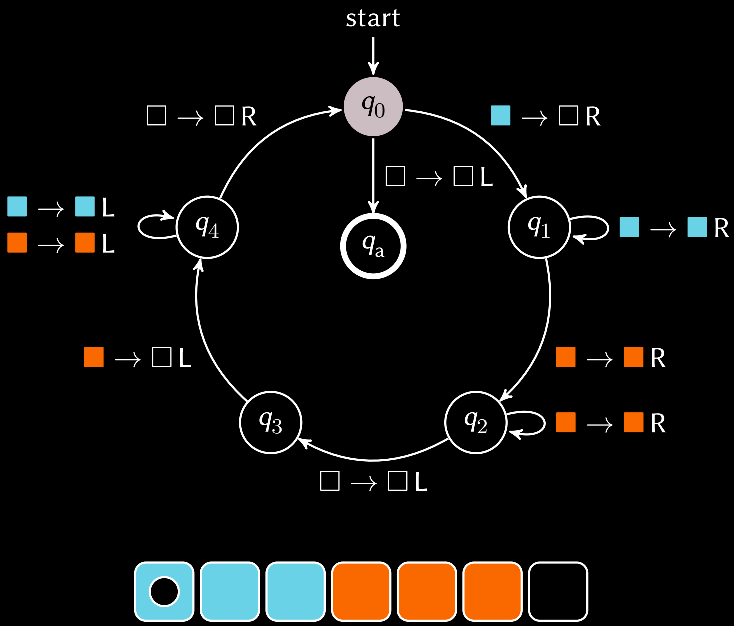
Bewijs.

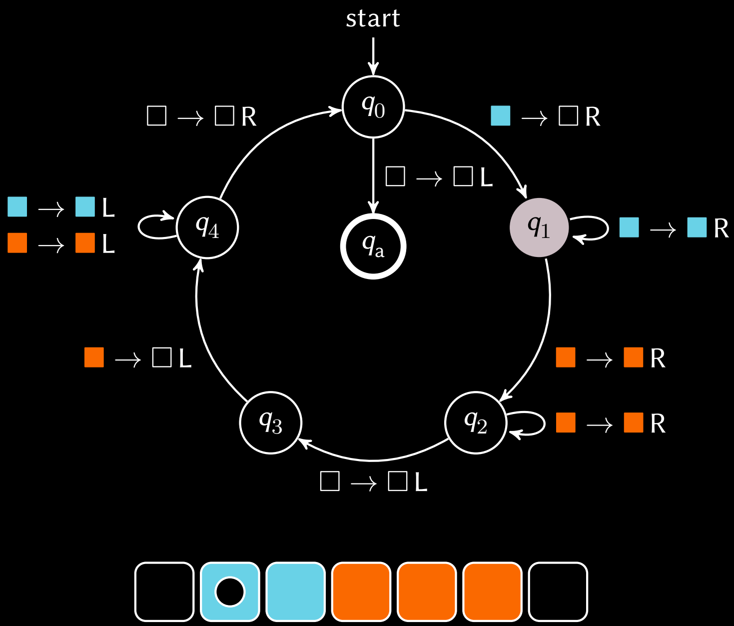
Maak CFG $\langle\, V, \Sigma_2, R, S \,\rangle$. Zet $V := V \cup \bigcup_{x \in \Sigma_2} V_x$ en $S = S$.
Per regel $A \to x$, maak regel $A \to S_x$ en neem de andere regels over.
Nu geldt $S \overset{*}{\Rightarrow} v_1 \ldots v_n$ d.e.s.d.a. $S \overset{*}{\Rightarrow} w_1 \ldots w_n$ en $S_{w_i} \overset{*}{\Rightarrow} v_i$. ∎

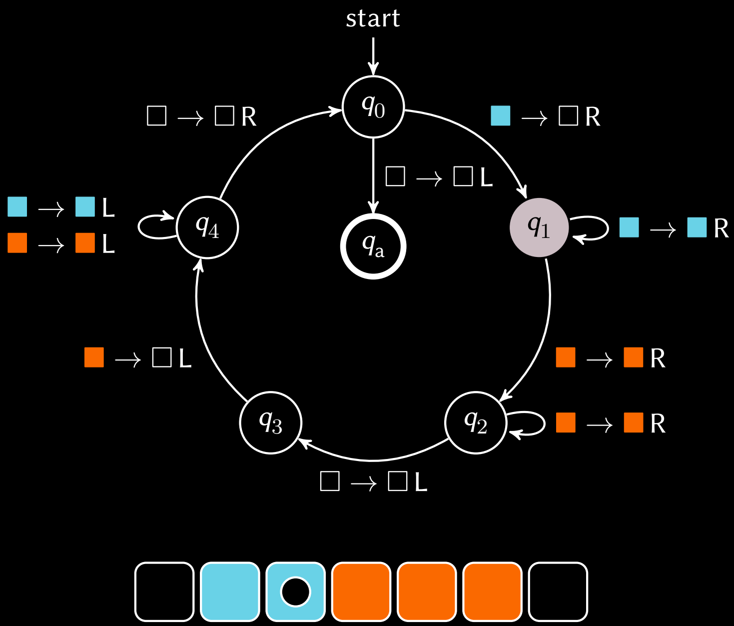
deel II

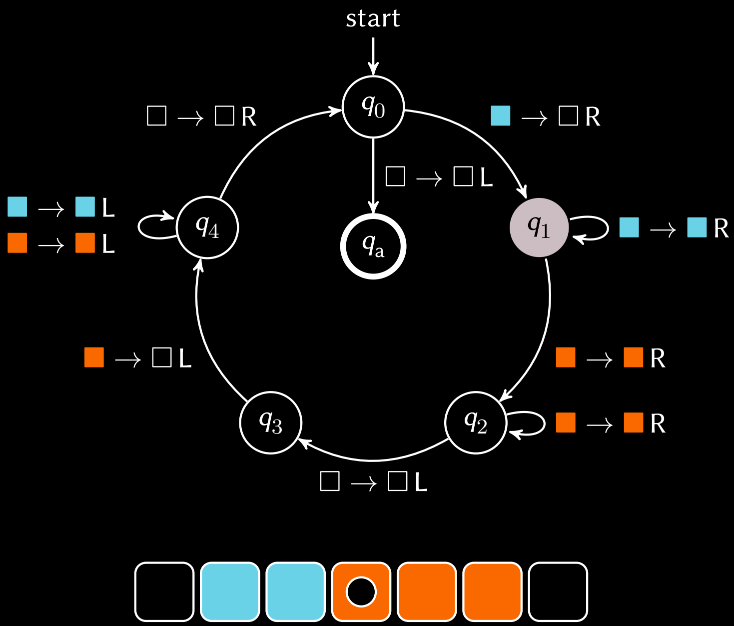
Berekenbaarheid

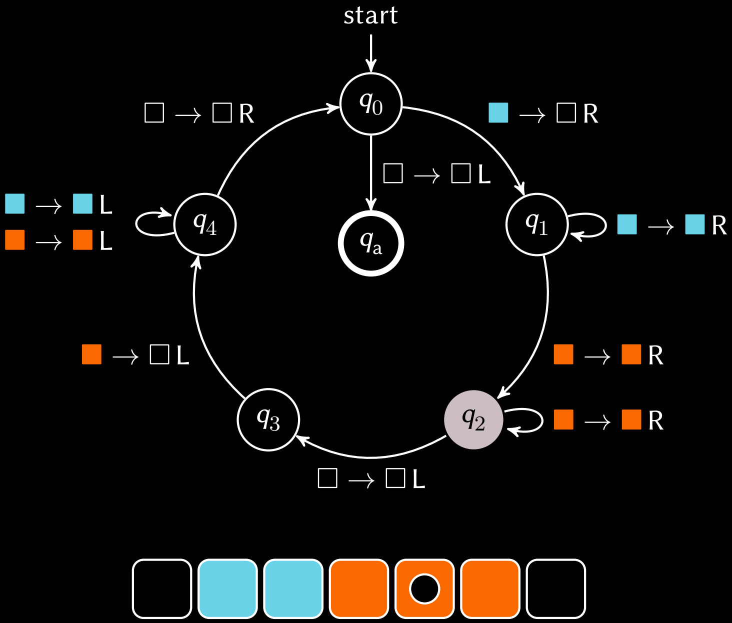

start
q_0
q_1
q_2
q_3
q_4
q_a

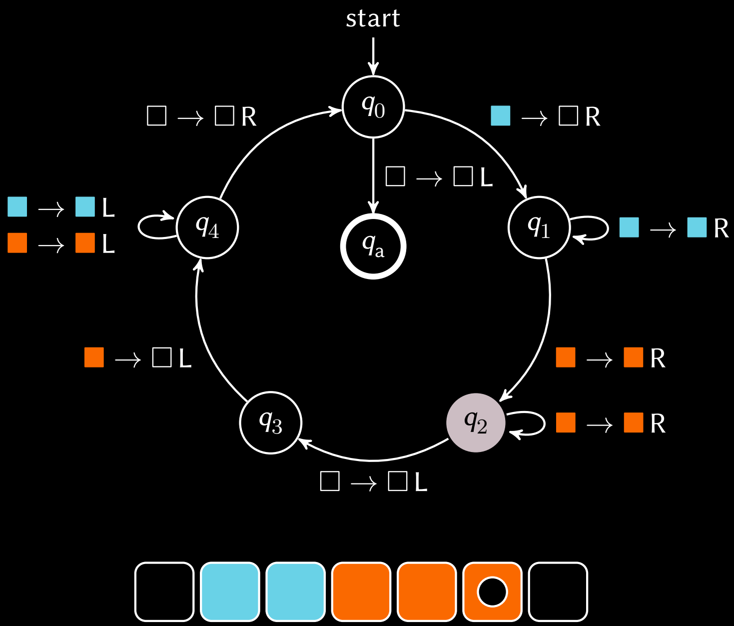

start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
□ → □ L
□ → □ L

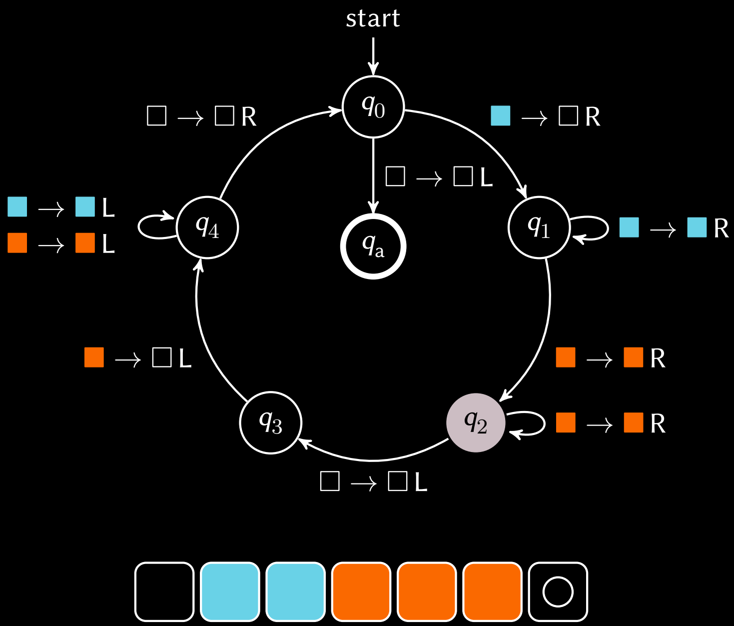

start
q_0
q_1
q_2
q_3
q_4
q_a

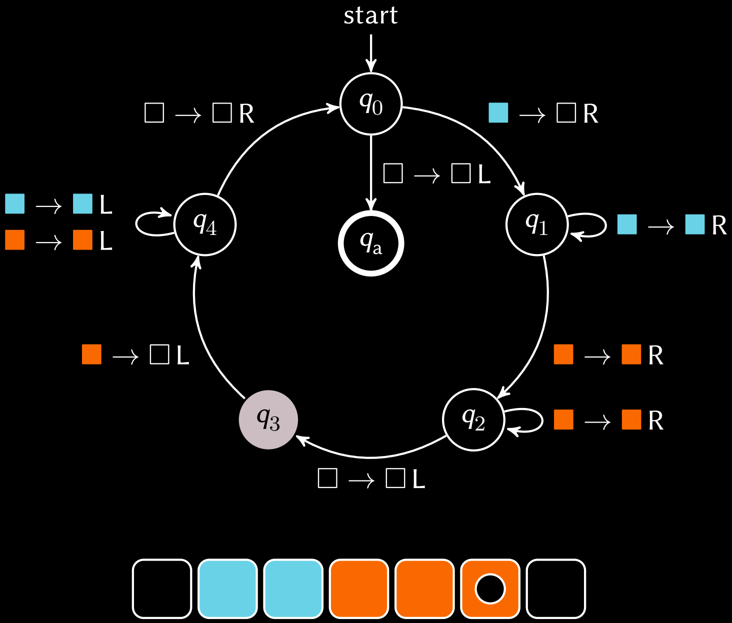

start
q0
q1
q2
q3
q4
qa

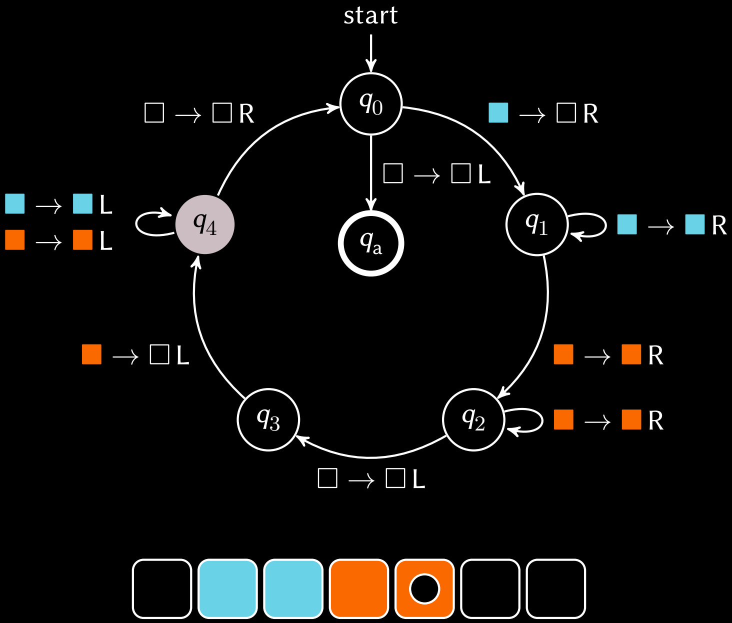

start
q_0
q_1
q_2
q_3
q_4
q_a

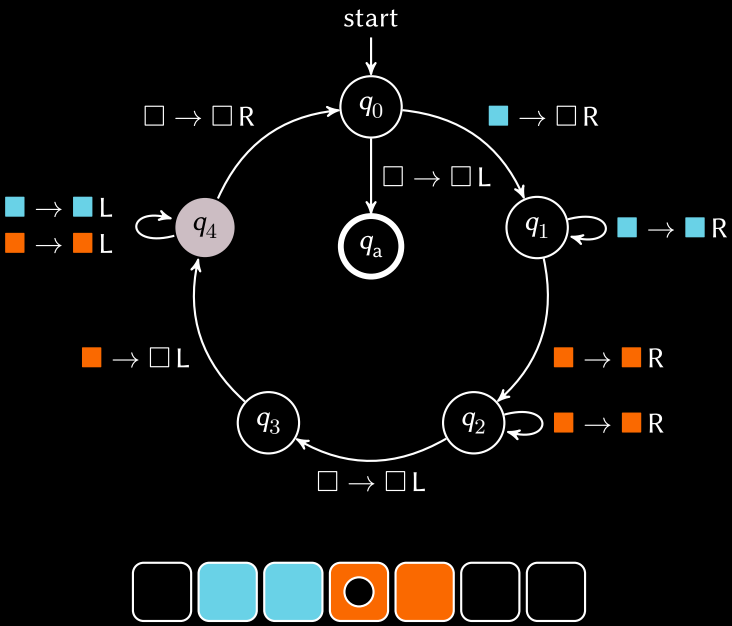

start
q_0
q_1
q_2
q_3
q_4
q_a

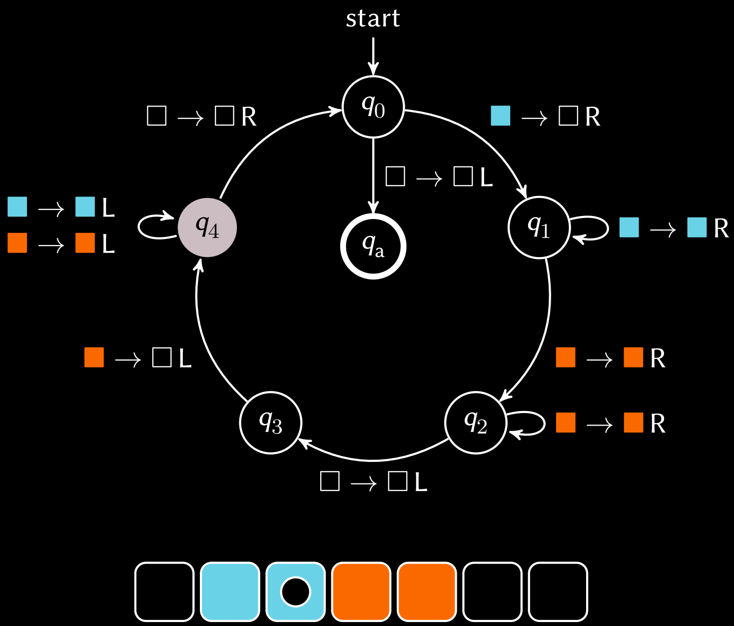

start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
■ → □ R
□ → □ L
■ → ■ R
■ → ■ R
■ → ■ R
□ → □ L
■ → □ L
■ → ■ L
■ → ■ L

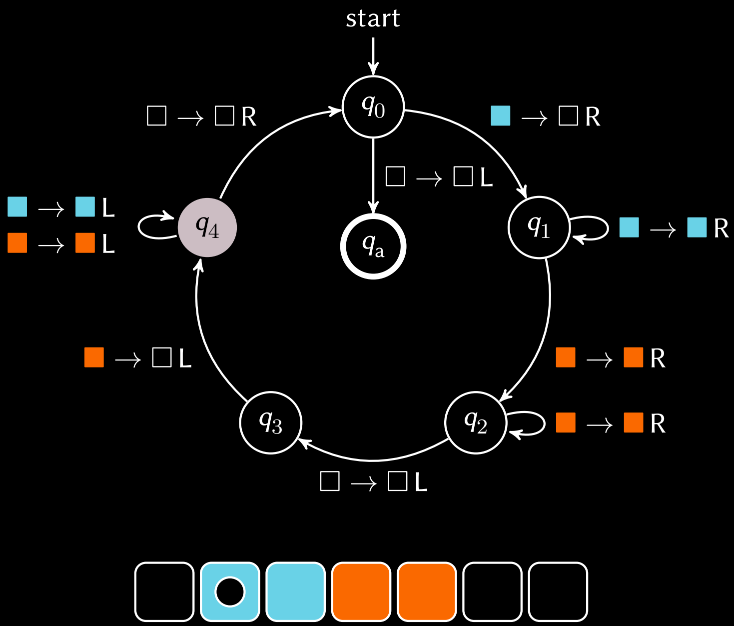

start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
□ → □ L
□ → □ L

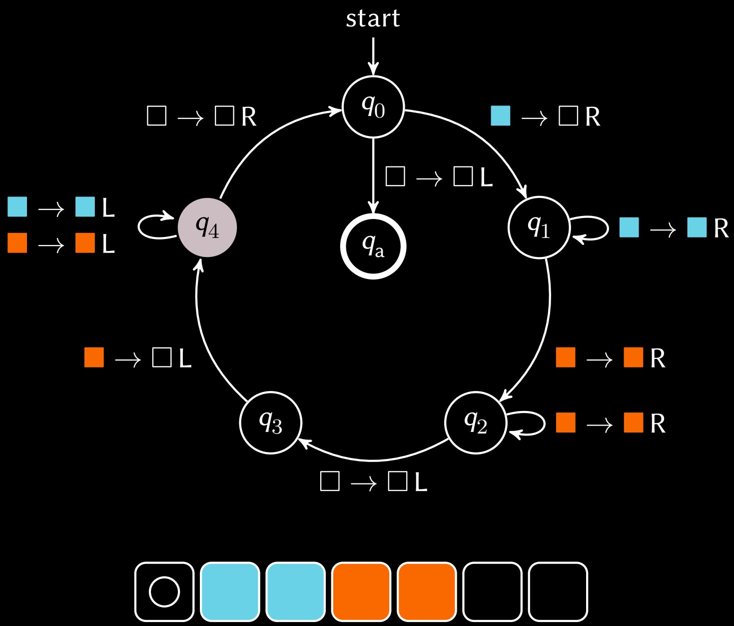

start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
□ → □ L
□ → □ L

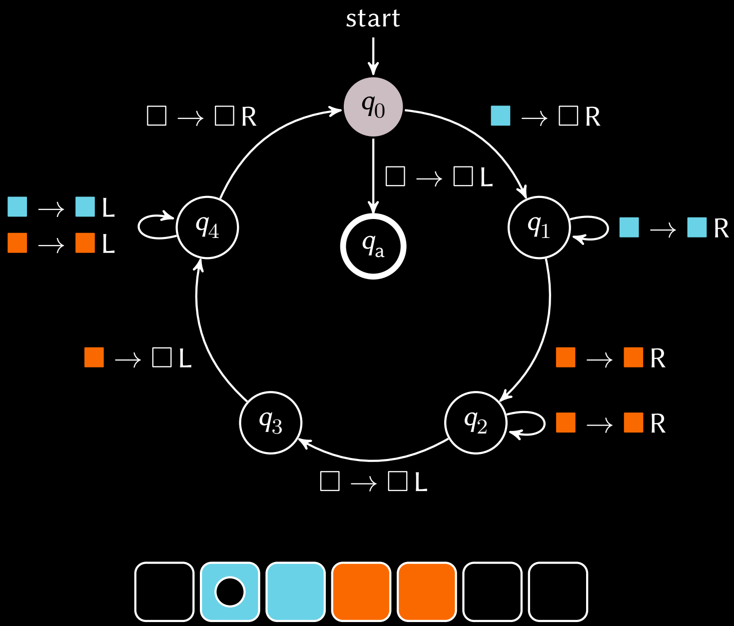

start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
□ → □ L
□ → □ L

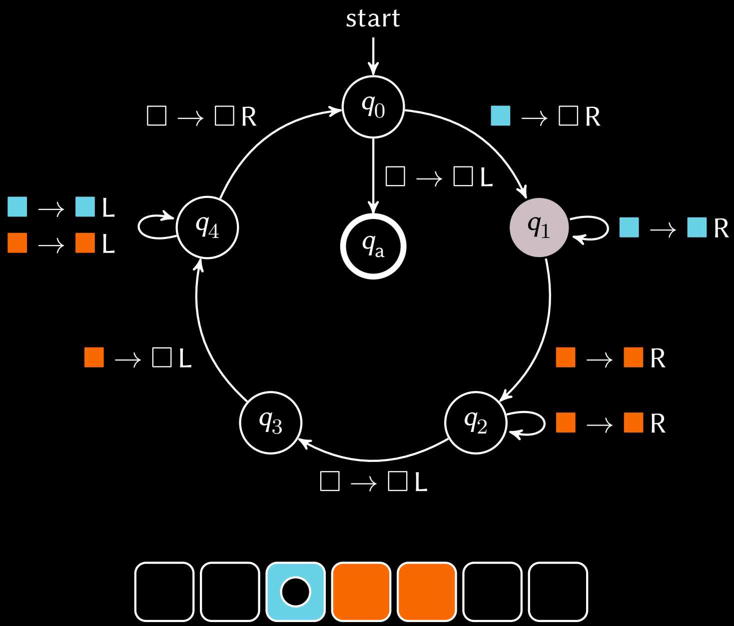

start
q_0
q_1
q_2
q_3
q_4
q_a
$\square \to \square$ R
$\square \to \square$ L
$\to \square$ R
$\to$ R
$\to$ R
$\to$ R
$\square \to \square$ L
$\to \square$ L
$\to$ L
$\to$ L

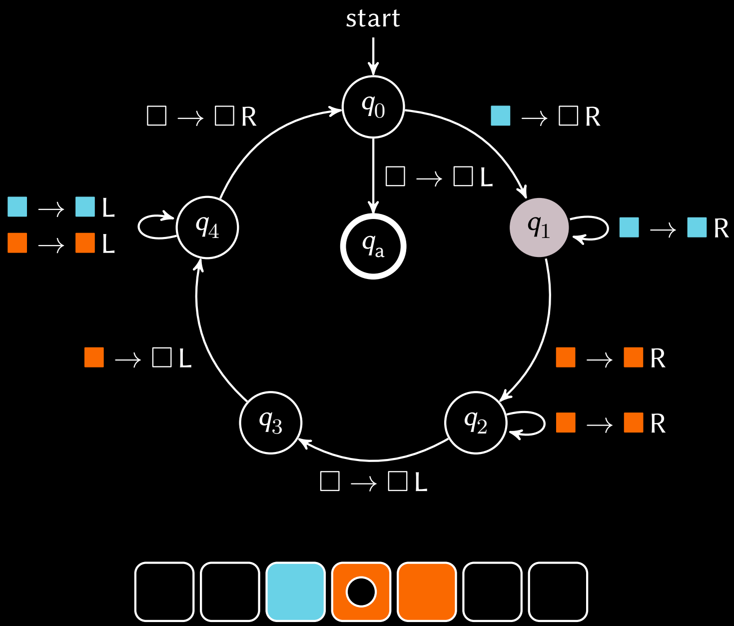

start
q0
q1
q2
q3
q4
qa

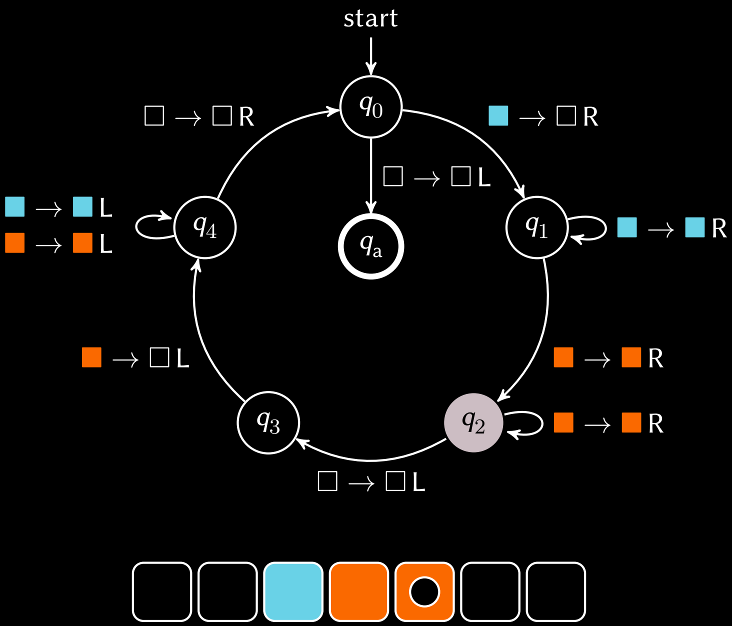

start
q_0
q_1
q_2
q_3
q_4
q_a

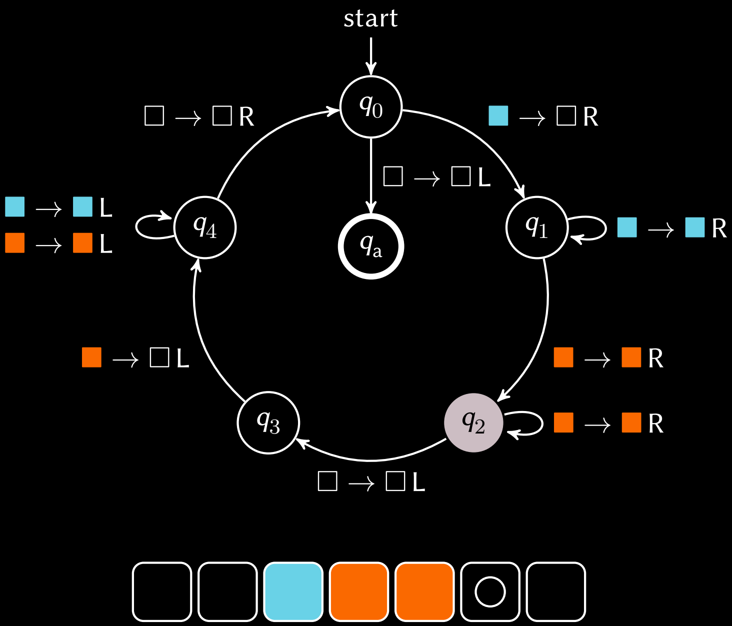

start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
□ → □ L
□ → □ L

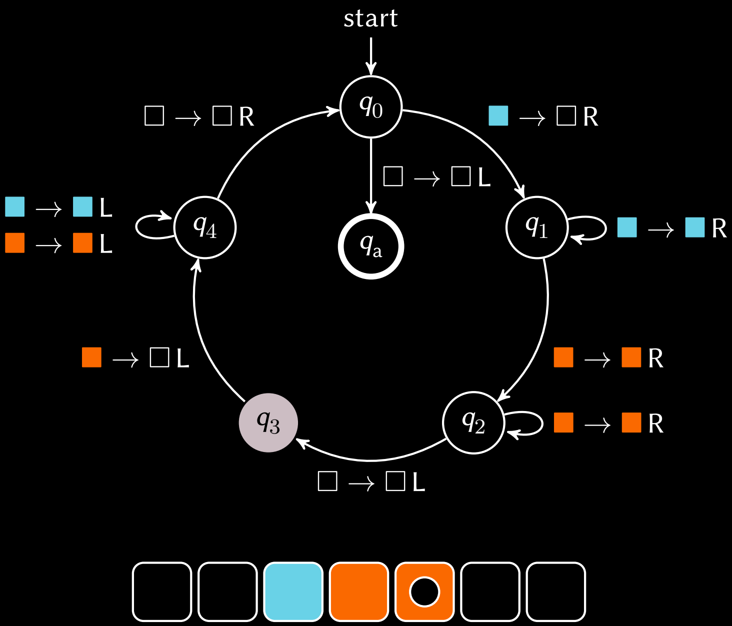

start
q_0
q_1
q_2
q_3
q_4
q_a

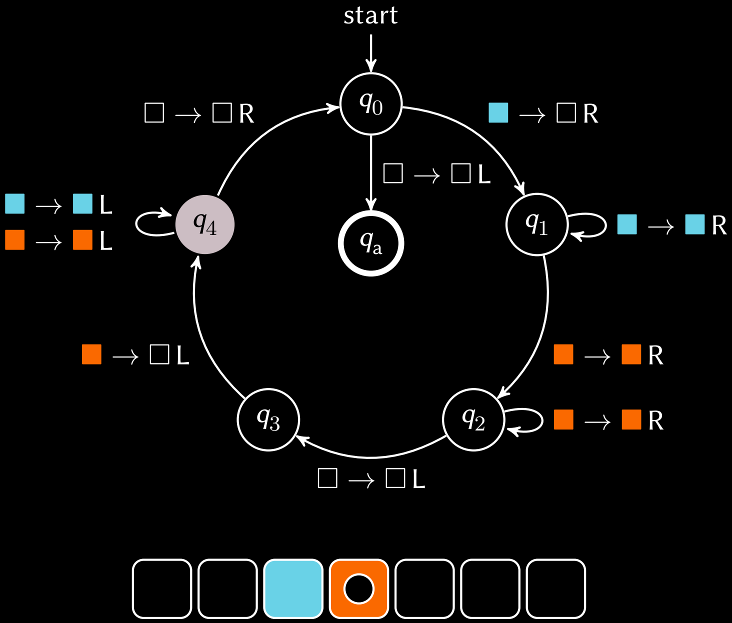

start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
→ □ R
□ → □ L
→ R
→ R
→ R
□ → □ L
→ □ L
→ L
→ L


start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
→ □ R
□ → □ L
→ R
→ L
→ L
→ R
→ R
→ □ L
□ → □ L

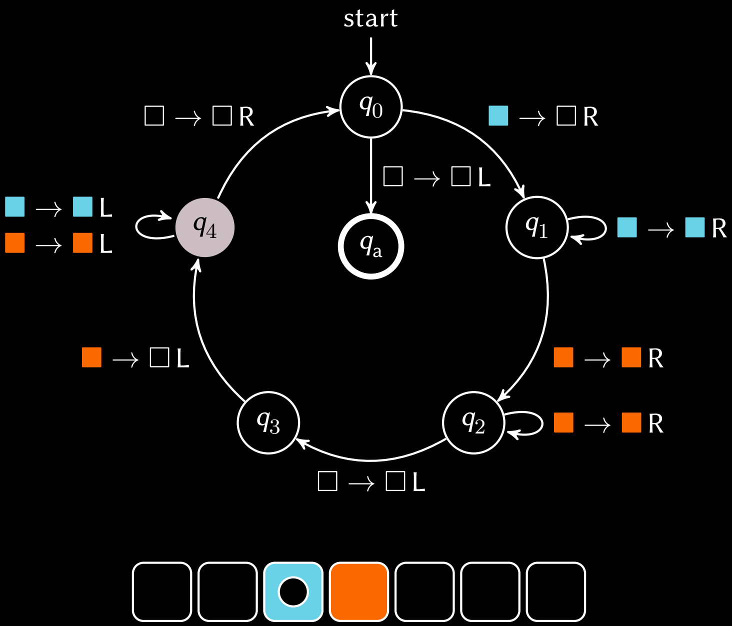

start
q_0
q_1
q_2
q_3
q_4
q_a

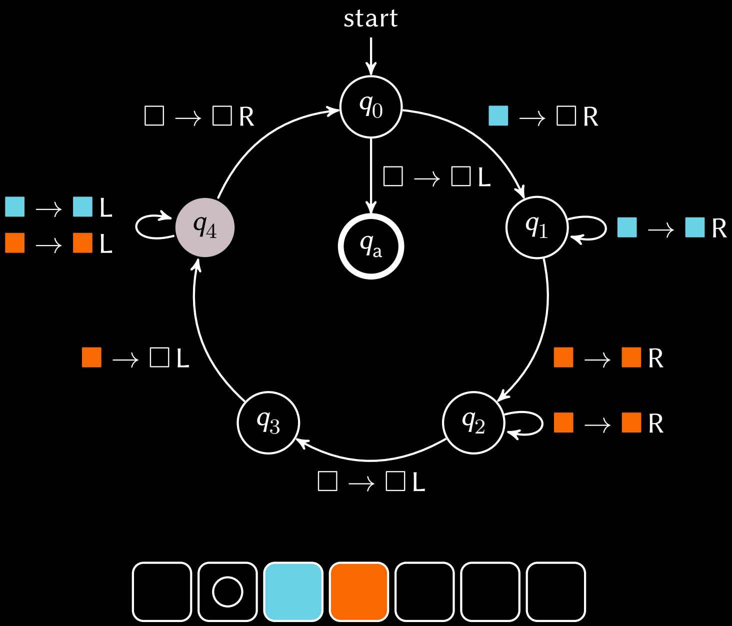

start
q_0
q_1
q_2
q_3
q_4
q_a

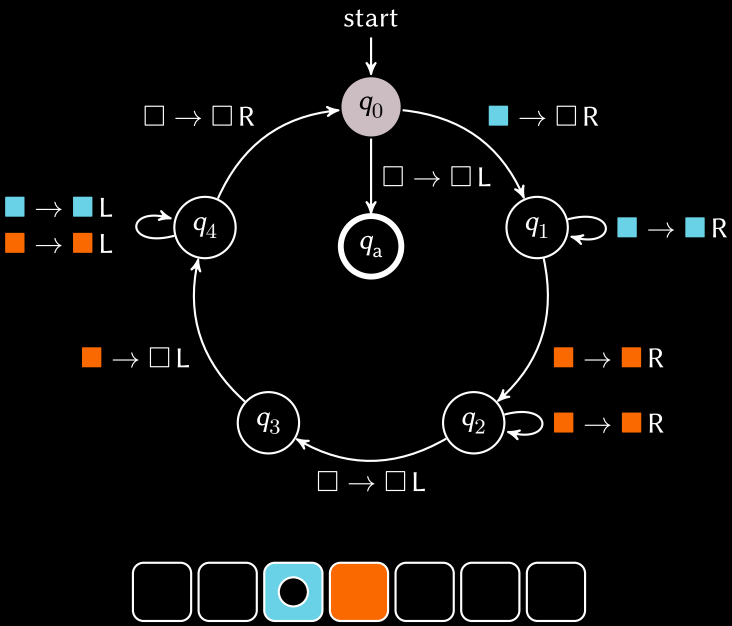

start
q_0
q_1
q_2
q_3
q_4
q_a
$\rightarrow$ R
$\rightarrow$ L



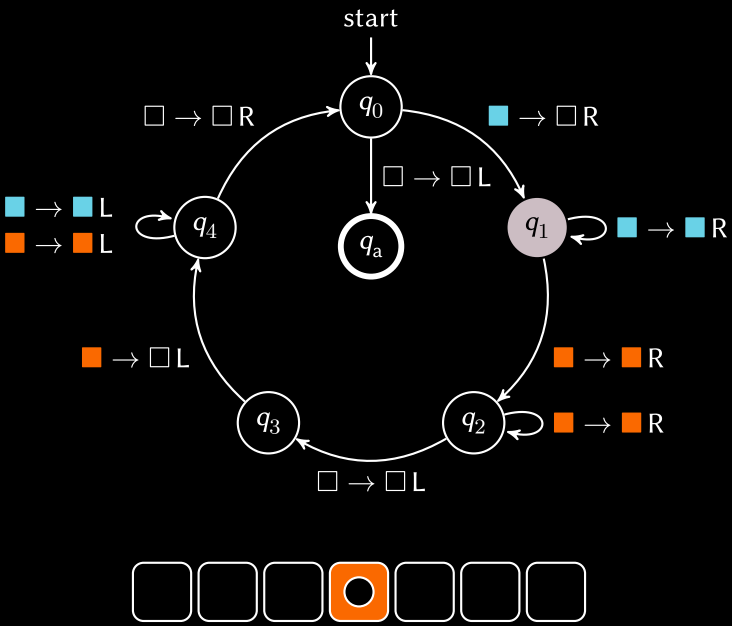
start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
■ → □ R
□ → □ L
■ → ■ R
■ → ■ R
■ → ■ R
□ → □ L
■ → □ L
■ → ■ L
■ → ■ L

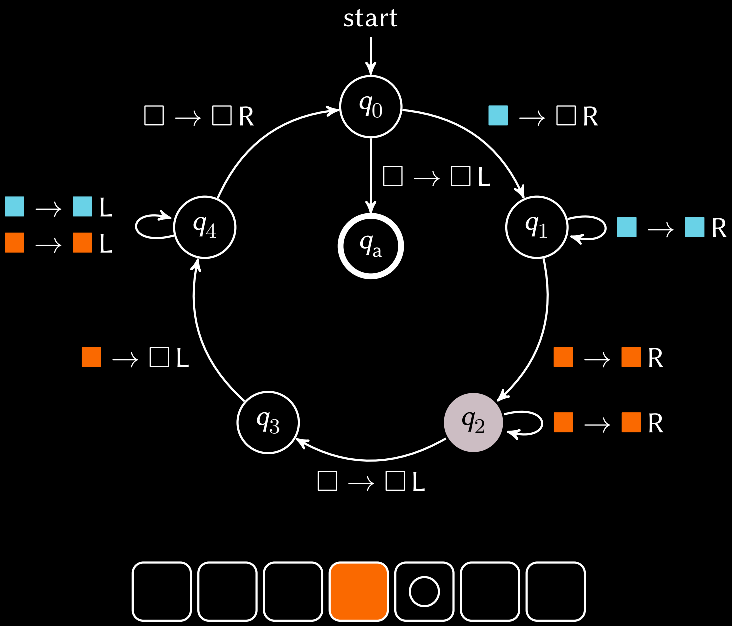

start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
□ → □ L
□ → □ R
□ → □ L
R
R
R
R
L
L
L

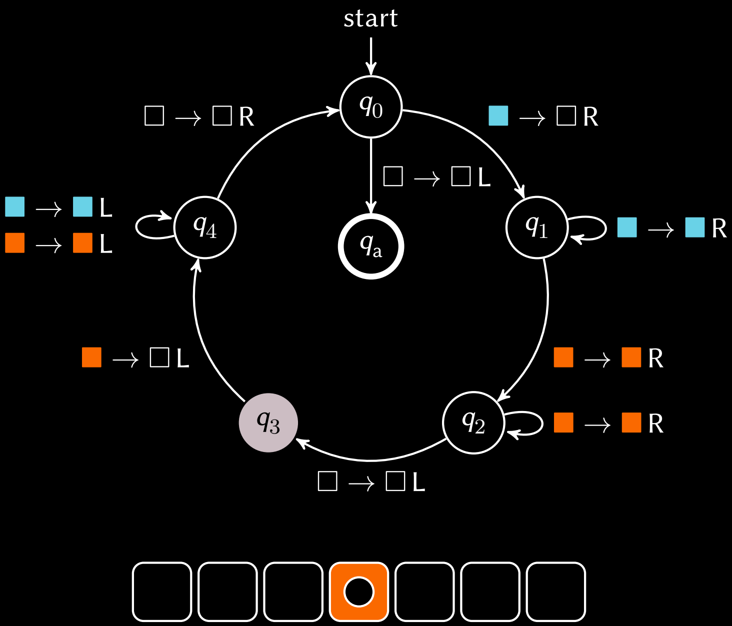

start
q_0
q_1
q_2
q_3
q_4
q_a



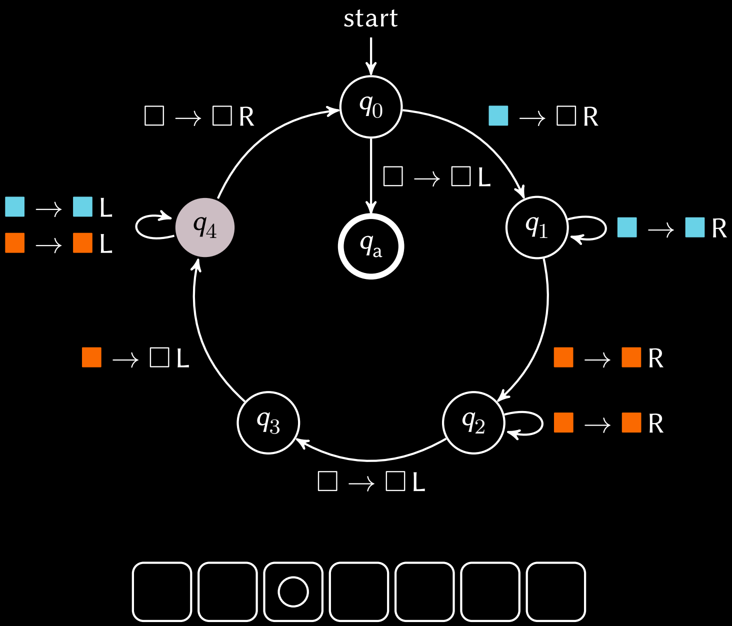
start
q0
q1
q2
q3
q4
qa
□ → □ R
□ → □ L
□ → □ L

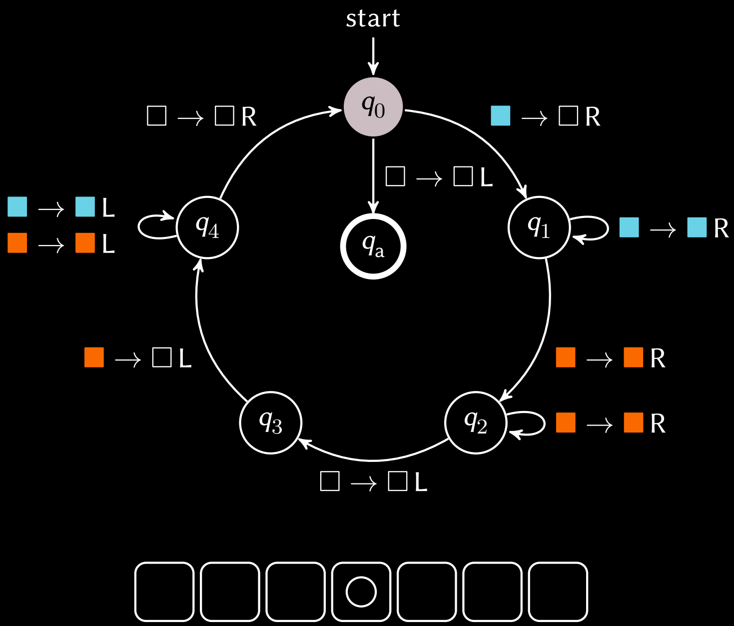

start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
→ □ R
□ → □ L
→ R
→ R
→ R
□ → □ L
→ □ L
→ L
→ L


start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
□ → □ L
□ → □ L

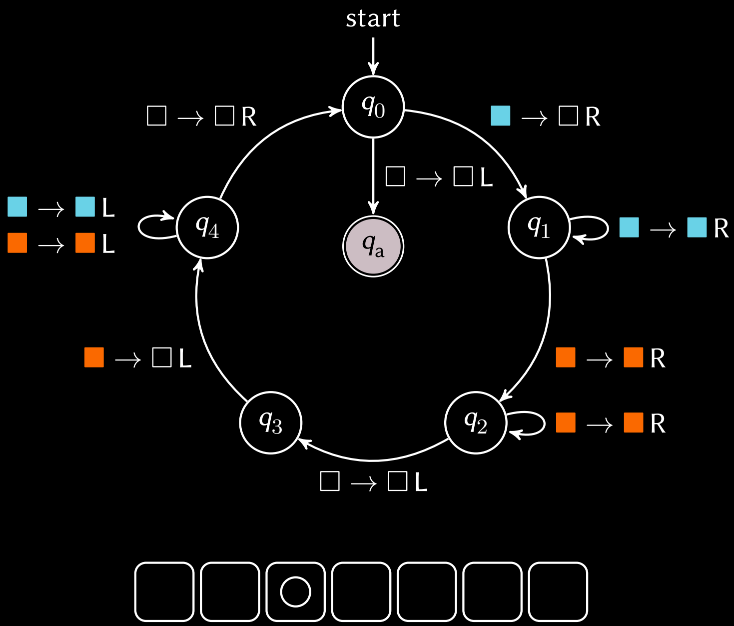


start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
□ → □ L
□ → □ L


start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
□ → □ L
□ → □ L



start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
→ □ R
□ → □ L
→ R
→ R
→ R
□ → □ L
→ □ L
→ L
→ L


start
q_0
q_1
q_2
q_3
q_4
q_a
□ → □ R
□ → □ L
□ → □ L

Hoog-niveau omschrijving

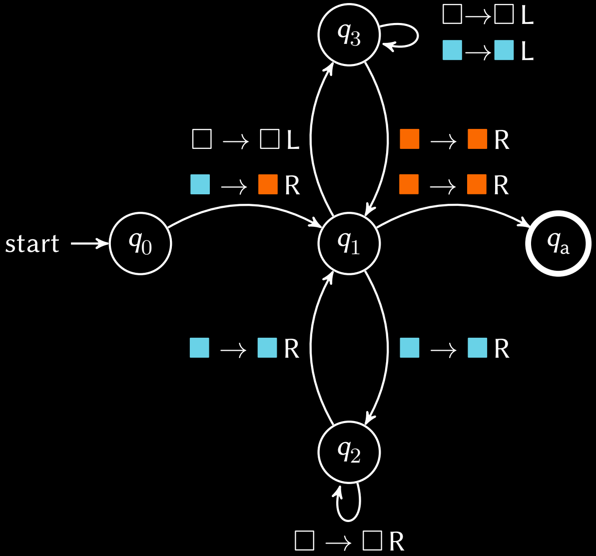
M = “Op invoer w:

1. Is de tape leeg, accepteer.

2. Wis de buitenste twee symbolen, als ze ■ en ■ zijn.

3. Ga naar de linkerkant van het woord, en ga verder vanaf 1.”

Voorbeeld



Belisser

Een TM heet een beslisser als ze altijd afwijst of accepteert.

Variaties

Variaties

minder symbolen

Variaties

minder symbolen

meer tapes

Variaties

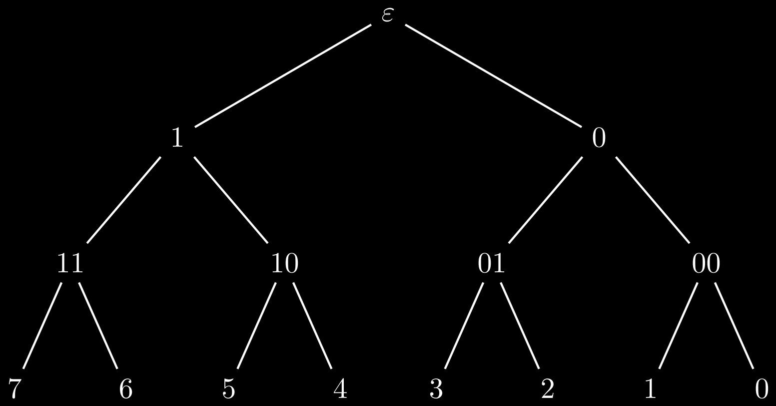
minder symbolen

meer tapes

non-determinisme

Getallen Lezen

Getallen Lezen



Minder symbolen

Elke TM M over $\{0, 1\}$ is equivalent aan een TM met tape-alfabet $\{0, 1, \text{␣}\}$.

Minder symbolen

$\langle\, Q, \Sigma, \Gamma, \delta, q_0, q_a, q_w \,\rangle$

Elke TM M over $\{0, 1\}$ is equivalent aan een TM met tape-alfabet $\{0, 1, \text{␣}\}$.

Minder symbolen

$\langle\, Q, \Sigma, \Gamma, \delta, q_0, q_{\mathrm{a}}, q_{\mathrm{w}} \,\rangle$

Elke TM M over $\{0, 1\}$ is equivalent aan een TM met tape-alfabet $\{0, 1, \textvisiblespace\}$.

Bewijsidee. Doe net alsof $\Gamma \subseteq \{0, 1, \ldots, 2^n - 1\}$. Codeer nu $x \in \Gamma$ als binair getal in n bits. Lezen en schrijven kost nu n stappen.

Minder symbolen

$\langle\, Q, \Sigma, \Gamma, \delta, q_0, q_a, q_w \,\rangle$

Elke TM M over $\{0,1\}$ is equivalent aan een TM met tape-alfabet $\{0,1,\sqcup\}$.

Bewijsidee. Doe net alsof $\Gamma \subseteq \{0,1,\ldots,2^n-1\}$. Codeer nu $x \in \Gamma$ als binair getal in n bits. Lezen en schrijven kost nu n stappen. Beginsituatie moet aangepast worden door gepast op te schuiven.

Minder symbolen

$\langle Q, \Sigma, \Gamma, \delta, q_0, q_a, q_w \rangle$

Elke TM M over $\{0, 1\}$ is equivalent aan een TM met tape-alfabet $\{0, 1, \text{␣}\}$.

Bewijsidee. Doe net alsof $\Gamma \subseteq \{0, 1, \ldots, 2^n - 1\}$. Codeer nu $x \in \Gamma$ als binair getal in n bits. Lezen en schrijven kost nu n stappen. Beginsituatie moet aangepast worden door gepast op te schuiven.

Coderen

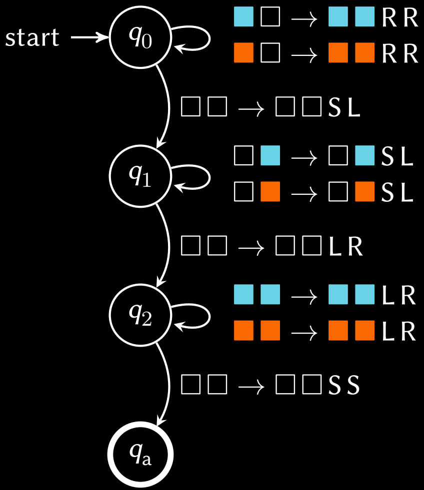
Eindige wiskundige structuren kunnen we coderen.

Multitape Turing Machine

Definitie
Een multitape TM is als een gewone TM, maar met:

$$\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{\mathsf{L}, \mathsf{S}, \mathsf{R}\}^k$$

Voorbeeld



Multitape Turing Machine

Elke TM met minstens twee tapes is equivalent aan een TM met een tape minder.

Non-Determinisme

Definitie
Een non-deterministische TM is als een gewone TM maar met:

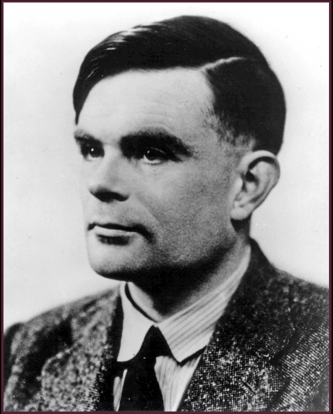
$$\delta : Q \times \Gamma \to \mathbf{P}\left(Q \times \Gamma \times \{\mathsf{L}, \mathsf{R}\}\right).$$

Non-Determinisme

Elke non-deterministische TM is equivalent aan een gewone TM.

Alonzo Church

University of St. Andrews



Alan Mathison Turing

computerhistory.org

Church & Turing

> *"It is my contention that these operations include all those which are used in the computation of a number."*
>
> — Alan M. Turing (1937)

Church & Turing

"It is my contention that these operations include all those which are used in the computation of a number."

— Alan M. Turing (1937)

Of these, the first has the advantage of making the identification with effectiveness in the ordinary (not explicitly defined) sense evident immediately [...]

— Alonzo Church (1937)

Church & Turing

"It is my contention that these operations include all those which are used in the computation of a number."

— Alan M. Turing (1937)

As a matter of fact, there is involved here the equivalence of three different notions: computability by a Turing machine, general recursiveness in the sense of Herbrand-Gödel-Kleene, and λ-definability in the sense of Kleene [1935] and the present reviewer. Of these, the first has the advantage of making the identification with effectiveness in the ordinary (not explicitly defined) sense evident immediately [...]

— Alonzo Church (1937)

Church & Turing

It is found in practice that [Turing machines] can do anything that could be described as "rule of thumb" or "purely mechanical"

— Alan M. Turing (1948)

Church–Turing These

Elk effectief algoritme kan berekend worden door een Turing machine.

Church–Turing These

Elk effectief algoritme kan berekend worden door een Turing machine.

Gezien

Turing machines

Gezien

Turing machines:
hoog niveau & coderen

Gezien

Turing machines:
hoog niveau & coderen

minder symbolen

meer tapes

non-deterministisch

Gezien

Turing machines:
hoog niveau & coderen

minder symbolen

meer tapes

non-deterministisch

effectief algoritme berekend door TM

Referenties

Church, Alonzo (1937). „Review of “On Computable Numbers, with an Application to the Entscheidungsproblem” by A. M. Turing”. In: *The Journal of Symbolic Logic* 2.1, p. 42–43. ISSN: 00224812. JSTOR: 2268810.

Kleene, Stephen Cole (1935). „A Theory of Positive Integers in Formal Logic. Part I”. In: *American Journal of Mathematics* 57.1, p. 153–173. ISSN: 00029327. JSTOR: 2372027.

Turing, Alan Mathison (1937). „On Computable Numbers, with an Application to the Entscheidungsproblem”. In: *Proceedings of the London Mathematical Society* 2.1, p. 2–42. DOI: 10.1112/plms/s2-42.1.230.