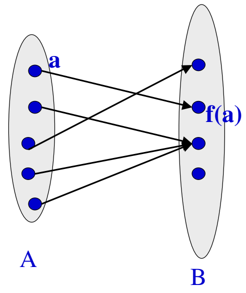
Funktion f:A→ B

f injektiv: ∀ a,b ∈ A, a≠b : f(a) ≠ f(b)

f surjektiv: ∀ b∈B ∃ a∈A: b=f(a)

f bijektiv: f ist injektiv und auch surjektiv



f Funktion: in jedem a∈A beginnt ein Pfeil

f Injektion: in jedem b∈B endet höchstens ein Pfeil

f Surjektion: in jedem b∈B endet mindestens ein Pfeil

f Bijektion: in jedem b∈B endet genau ein Pfeil

Funktion f : A → B

f injektiv ⇔ ∀ a,a' ∈ A, a ≠ a' : f(a) ≠ f(a')

f surjektiv ⇔ ∀ b∈ B ∃ a∈ A : f(a) = b

f bijektiv ⇔ f injektiv und f surjektiv

Mengen A und B sind gleichmächtig ⇔ ∃ Bijektion f : A→ B

Menge A ist endlich ⇔ ∃ k∈ ℕ : A gleichmächtig mit {1,…,k}

Menge A ist abzählbar unendlich ⇔ A gleichmächtig mit ℕ

Menge A ist abzählbar ⇔ A ist endlich oder A ist abzählbar unendlich

Lemma 1.0:

A abzählbar ⇔ ∃ Injektion A → ℕ ⇔ ∃ Surjektion ℕ → A

Lemma 1.1: B abzählbar und A ⊂ B ⇒ A abzählbar

Lemma 1.2: $A_1, \dots, A_k$ endlich ⇒ $\bigcup_{1\le i\le k} A_i$ ist endlich
$A_1 \times \cdots \times A_k$ ist endlich

Lemma 1.3: A_i endlich für $i \in \mathbb{N}$ ⇒ $\bigcup_{i\in\mathbb{N}} A_i$ ist abzählbar

Kor. für später: Σ endliches Alphabet ⇒ Σ^* ist abzählbar

$(\Sigma^* = \bigcup_{i\in\mathbb{N}} \Sigma^i)$

Lemma 1.4: A, B abzählbar ⇒ A ∪ B abzählbar
A × B abzählbar

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	...
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	...
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	...
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	...
(4,0)	(4,1)	(4,2)	(4,3)	(4,4)	...
⋮	⋮	⋮	⋮	⋮	⋱

Kor.: ℚ ist abzählbar

Lemma 1.5: A_i abzählbar für i∈ℕ ⇒ $\bigcup_{i\in\mathbb{N}} A_i$ ist abzählbar

Korollar 1: $2^{\mathbb{N}}$ ist nicht abzählbar

Korollar 2: Die Menge aller unbeschränkten Bitfolgen
$\{\, \alpha \mid \alpha : \mathbb{N} \to \{0,1\} \,\}$
is nicht abzählbar.

	0	1	2	3	4	5	6	7	...
α_0	1	1	0	1	0	1	1	0	...
α_1	0	1	1	1	0	0	1	0	...
α_2	0	0	0	1	0	1	1	1	...
α_3	1	0	0	0	0	0	0	0	...
α_4	1	1	0	1	0	1	0	0	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Cantorsche Diagonalisierung

Korollar 1: $2^{\mathbb{N}}$ ist nicht abzählbar

Korollar 2: Die Menge aller unbeschränkten Bitfolgen
$\{\, \alpha \mid \alpha : \mathbb{N} \to \{0,1\} \,\}$
is nicht abzählbar.

	0	1	2	3	4	5	6	7	...
α_0	1 0	1	0	1	0	1	1	0	...
α_1	0	1 0	1	1	0	0	1	0	...
α_2	0	0	0 1	1	0	1	1	1	...
α_3	1	0	0	0 1	0	0	0	0	...
α_4	1	1	0	1	0 1	1	0	0	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots\vdots$	$\vdots$	$\vdots$	$\ddots$

Cantorsche Diagonalisierung

$\delta(i) = 1-\alpha_i(i)$

Korollar 1: $2^{\mathbb{N}}$ ist nicht abzählbar

Korollar 2: Die Menge aller unbeschränkten Bitfolgen
$\{\, \alpha \mid \alpha : \mathbb{N} \to \{0,1\} \,\}$
is nicht abzählbar.

	0	1	2	3	4	5	6	7	...
α_0	1 0	1	0	1	0	1	1	0	...
α_1	0	1 0	1	1	0	0	1	0	...
α_2	0	0	0 1	1	0	1	1	1	...
α_3	1	0	0	0 1	0	0	0	0	...
α_4	1	1	0	1	0 1	1	0	0	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots\vdots$	$\vdots$	$\vdots$	$\ddots$

Cantorsche Diagonalisierung

$\delta(i) = 1-\alpha_i(i)$

Korollar 3: $\mathbb{R}$ ist nicht abzählbar

Wörter, Strings, Sprachen

Σ . . . endliche Menge "Alphabet"

Σ^k . . . geordnete k-Tupel $\quad w = (a_1, a_2, \ldots, a_k)$
Wörter (Strings) der Länge k $\quad w = a_1 a_2 \cdots a_k \quad |w| = k$

ε leeres Wort, Wort der Länge 0 $\quad |\varepsilon| = 0$

$\Sigma^0 = \{\varepsilon\}$

$\Sigma^* = \bigcup_{k\in\mathbb{N}} \Sigma^k$ alle endlichen Stings über Σ

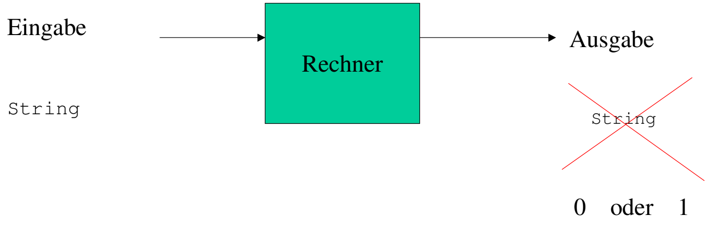
$\Sigma^+ = \bigcup_{k>0} \Sigma^k$ alle endlichen, nichtleeren Stings über Σ

Konkatenation:
$a = a_1 \cdots a_m$, $b = b_1 \cdots b_n$ $\quad a \cdot b = a_1 \cdots a_m b_1 \cdots b_n$

$w \in \Sigma^*$, $a \in \Sigma$ $\quad \#_a(w)$ = wie oft kommt Symbol a in Wort w vor?

$L \subset \Sigma^*$ . . . Sprache über Σ

Einfache Rechensituation



Erkennen einer „Sprache“

Konfiguration (Momentaufnahme der Maschine)

- Inhalt des Eingabebandes
- Position des Eingabekopfes
- Zustand
- Speicherinhalt

Startkonfiguration für Eingabe x: $start_x$
Endkonfigurationen

Automat akzeptiert Eingabe *x*, wenn er durch eine endliche Anzahl von Schritten aus $start_x$ eine Endkonfiguration erreicht.

Determinismus

Konfiguration (Momentaufnahme der Maschine)

- Inhalt des Eingabebandes
- Position des Eingabekopfes
- Zustand
- Speicherinhalt

Startkonfiguration für Eingabe x: $start_x$
Endkonfigurationen

Automat akzeptiert Eingabe *x*, wenn er durch eine endliche Anzahl von Schritten aus $start_x$ eine Endkonfiguration erreichen kann

Nicht - **Determinismus**

Konfigurationsgraph einer Maschine R

- Knoten sind die möglichen Konfigurationen von R (u.U. unendlich viele)
- gerichtete Kante von Konfiguration K nach Konfiguration K' , genau dann wenn man durch Anwendung von einer der endlich vielen Rechenschrittregeln der Maschine R von K nach K' kommt

Startkonfiguration für Eingabe x: $start_x$ Endkonfigurationen

Maschine R akzeptiert Eingabe *x*, wenn sie durch eine endliche Anzahl von Schritten aus $start_x$ eine Endkonfiguration erreichen kann,

anders gesagt, wenn es im Konfigurationsgraphen von R einen Pfad von $start_x$ zu einer Endkonfiguration gibt.

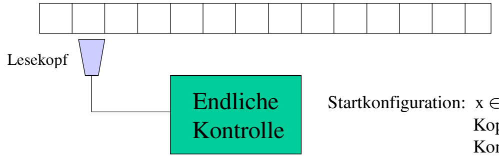
Maschine R **deterministisch**: in jeder Konfiguration höchstens eine Rechenschrittregel anwendbar ist (also für alle Knoten im Konfigurationsgraph ist der Ausgrad ≤ 1)

Sonst heißt R **nicht-deterministisch**

3 Arten von Rechenmaschinen

- **kein Speicher** endlicher Automat
- **Speicher ist Band mit Zellen und Schreib/Lesekopf** Turing Maschine
- **Speicher ist Band mit Zellen und Schreib/Lesekopf, der nur an einem Ende des Bandes agiert** Keller Automat

Endlicher Automat



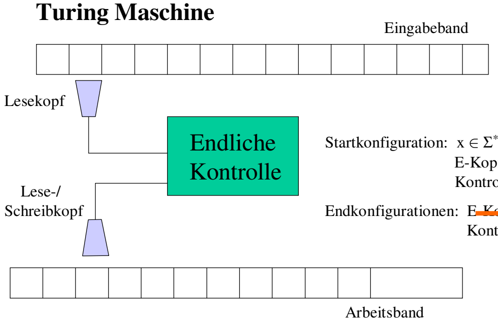
Startkonfiguration: $x \in \Sigma^*$ auf Band,
Kopf ganz links,
Kontrolle in Startzustand

Endkonfigurationen: Kopf ganz rechts
Kontrolle in einem Endzustand

Rechenschrittregeln:

derzeitiger Zustand gelesenes Symbol → neuer Zustand Kopf nach rechts

Turing Maschine



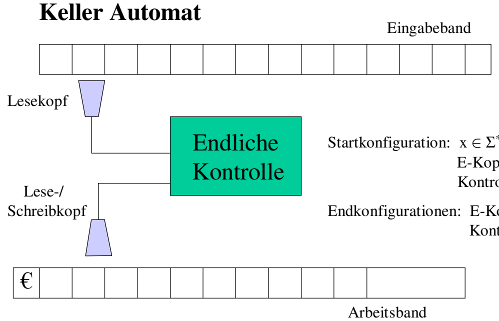
Startkonfiguration: $x \in \Sigma^*$ auf E-Band, nichts auf A-Band
E-Kopf ganz links,
Kontrolle in Startzustand

Endkonfigurationen: ~~E-Kopf ganz rechts~~
Kontrolle in einem Endzustand

Rechenschrittregeln:

derzeitiger Zustand gelesenes E-Symbol gelesenes A-Symbol → neuer Zustand E-Kopf nach rechts schreibe A-Symbol bewege A-Kopf

Keller Automat



Startkonfiguration: $x \in \Sigma^*$ auf E-Band, € links auf A-Band
E-Kopf ganz links, A-Kopf ganz links
Kontrolle in Startzustand

Endkonfigurationen: E-Kopf ganz rechts
Kontrolle in einem Endzustand

Rechenschrittregeln:

derzeitiger Zustand gelesenes E-Symbol gelesenes A-Symbol → neuer Zustand E-Kopf nach rechts schreibe A-Symbol bewege A-Kopf

Wenn A-Kopf nach links bewegt wird, hinterlässt er leere Zelle.

L DEA-Sprache : Es gibt einen deterministischen endlichen Automaten, der L akzeptiert.

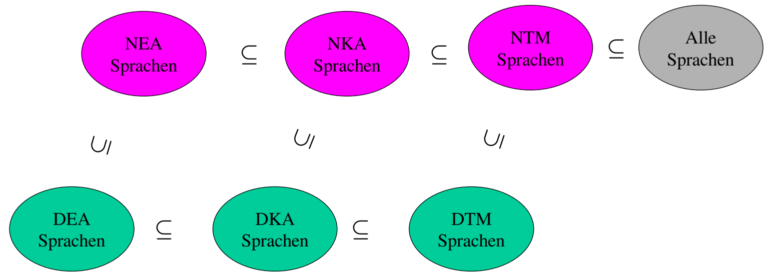
L NEA-Sprache : Es gibt einen nicht-deterministischen endlichen Automaten, der L akzeptiert.

L DKA-Sprache : Es gibt einen deterministischen Keller-Automaten, der L akzeptiert.

L NKA-Sprache : Es gibt einen nicht-deterministischen Keller-Automaten, der L akzeptiert.

L DTM-Sprache : Es gibt eine deterministischen Turing Maschine, die L akzeptiert.

L NTM-Sprache : Es gibt eine nicht-deterministische Turing Maschine, die L akzeptiert.



Lemma: Es gibt Sprachen, die keine NTM-Sprachen sind.

Beweis: es gibt nur abzählbare viele Turing Maschinen (denn jede TM kann durch einen endlichen ASCII String beschrieben werden),

also

gibt es nur abzählbar viele NTM-Sprachen.

Aber die Menge aller Sprachen ist **nicht** abzählbar.