Theoretische Grundlagen der Informatik
Übung

4. Übungstermin · 30. November 2017

Guido Brückner

INSTITUT FÜR THEORETISCHE INFORMATIK · LEHRSTUHL ALGORITHMIK

Gliederung

Turingmaschinen und Berechenbarkeit

- Universelle Turingmaschinen
- Entscheidbarkeit und Semi-Entscheidbarkeit
- Satz von Rice
- Post'sches Korrespondenzproblem

Komplexitätsklassen

- Sprachen, Problem und Zeitkomplexität
- Klasse NP
- Über die Klasse P und NP hinaus

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Wiederholung

Entscheidbarkeit

Satz: Eine Sprache $L \subseteq \Sigma^\star$ heißt **rekursiv** oder **entscheidbar**, wenn es eine Turing-Maschine gibt, die auf allen Eingaben stoppt und eine Eingabe w genau dann akzeptiert, wenn $w \in L$ gilt.

$\Rightarrow \mathcal{M}$ entscheidet L.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Semi-Entscheidbarkeit

Satz: Eine Sprache $L \subseteq \Sigma^\star$ heißt **rekursiv-aufzählbar** oder **semi-entscheidbar**, wenn es eine Turing-Maschine gibt, die genau die Eingaben w akzeptiert für die $w \in L$. Das Verhalten der Turing-Maschine für Eingaben $w \notin L$ ist damit nicht definiert. D.h., die Turing-Maschine stoppt entweder nicht in einem Endzustand oder aber stoppt gar nicht.

$\Rightarrow \mathcal{M}$ akzeptiert L.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Beziehung zwischen Entscheidbarkeit und Semi-Entscheidbarkeit

Satz: Eine Sprache L ist genau dann entscheidbar, wenn L und deren Komplement L^c semi-entscheidbar sind.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Bisher:

- Bislang beschriebenen DTMs sind für spezielle Aufgaben

Intuitiver Wunsch:

- Eine Art programmierbarer Rechner, der als Eingabe ein Programm und die Eingabe für dieses Programm bekommt

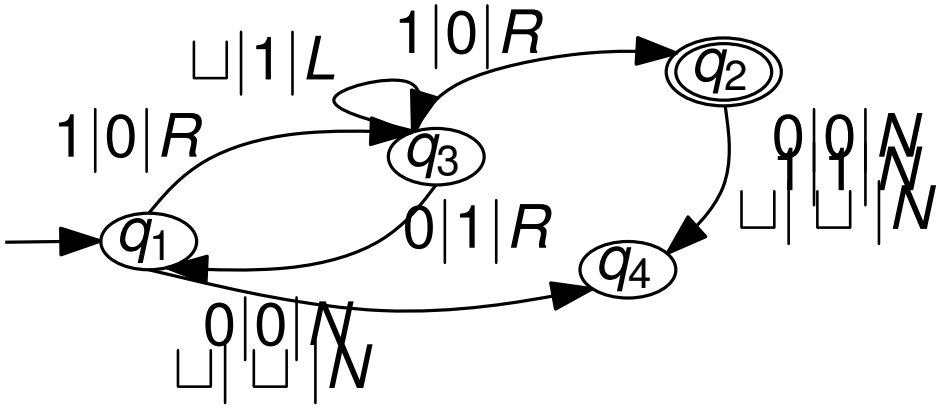
Beschreibung einer TM

- $\mathcal{M} := (Q, \Sigma, \Gamma, \delta, s, F)$
- Gödelnummer $\langle \mathcal{M} \rangle$ von $\mathcal{M}$, ist definiert durch folgende Kodierungsvorschrift:
  1. Kodiere $\delta(q_i, a_j) = (q_r, a_s, d_t)$ durch $0^i 1 0^j 1 0^r 1 0^s 1 0^t$,
     mit $d_t \in \{d_1, d_2, d_3\}$, d_1 für L, d_2 für R und d_3 für N
  2. Turing-Maschine wird kodiert durch:
     $111\text{code}_1 11\text{code}_2 11 \ldots 11\text{code}_z 111$,
     mit code_i für $i = 1, \ldots, z$ entspricht allen Funktionswerten von δ

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

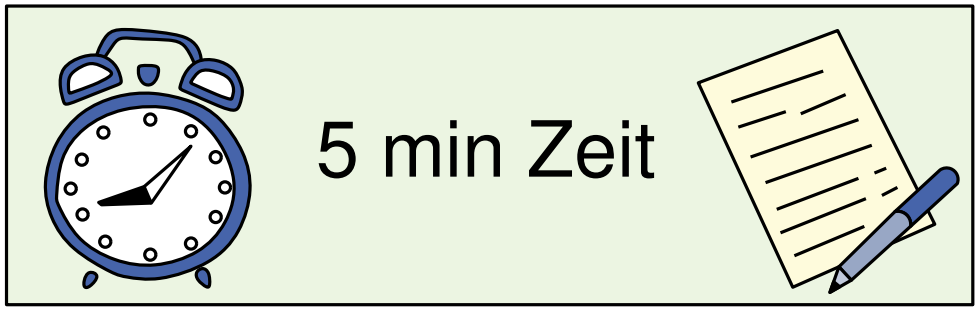
Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ:

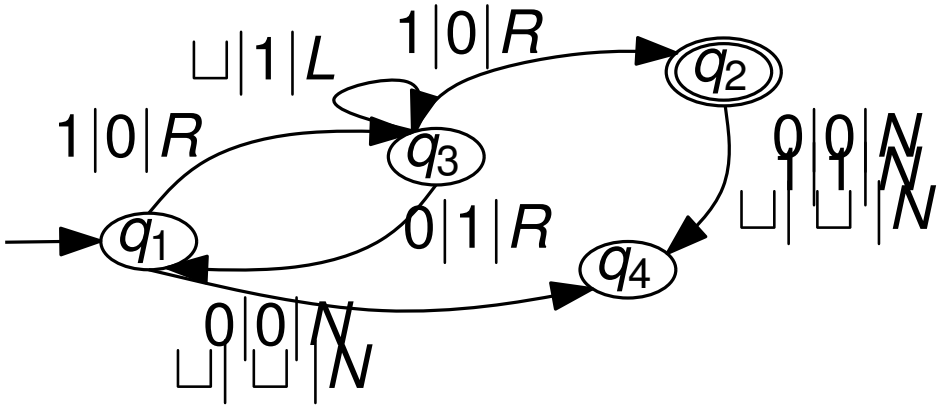


Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ:





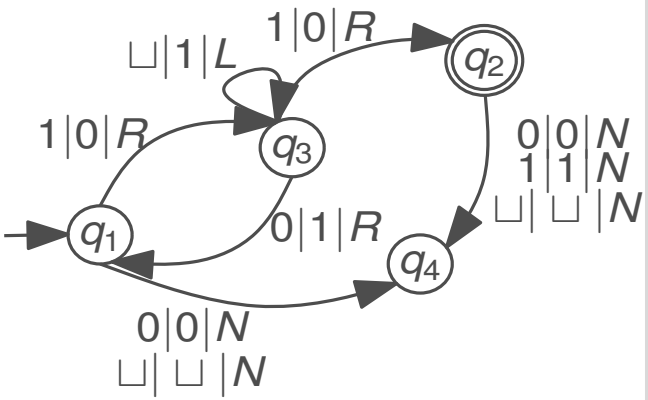
- Gödelnummer $\langle \mathcal{M} \rangle$ von $\mathcal{M}$, ist definiert durch folgende Kodierungsvorschrift:
  1. Kodiere $\delta(q_i, a_j) = (q_r, a_s, d_t)$ durch $0^i 1 0^j 1 0^r 1 0^s 1 0^t$, mit $d_t \in \{d_1, d_2, d_3\}$, d_1 für L, d_2 für R und d_3 für N
  2. Turing-Maschine wird kodiert durch: $111\text{code}_1 11\text{code}_2 11 \ldots 11\text{code}_z 111$, mit code_i für $i = 1, \ldots, z$ entspricht allen Funktionswerten von δ

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

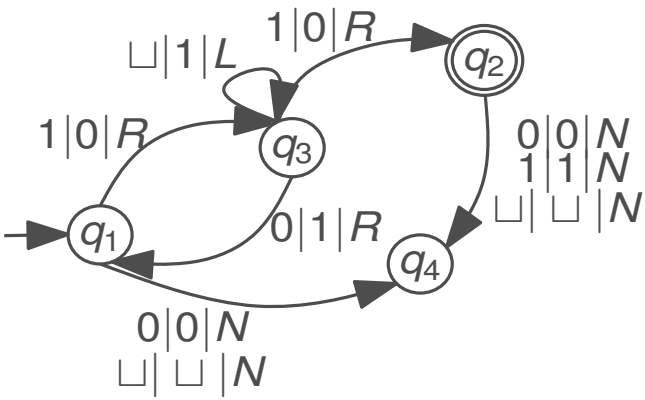
Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

$X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup$

$D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N$

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

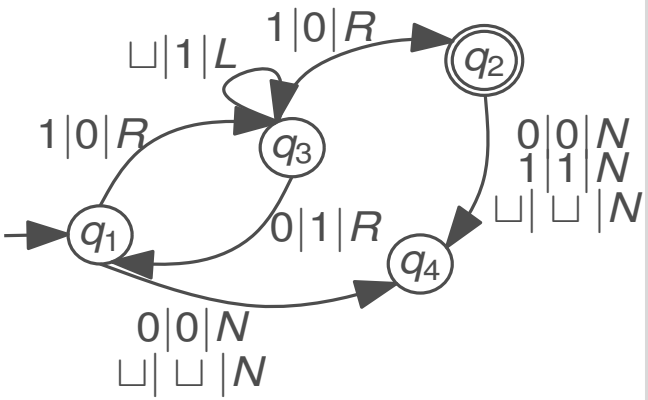
Universelle Turingmaschine

Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

$$\left.\begin{array}{l} X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup \\ D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N \end{array}\right\} \quad \delta(q_i, X_j) = (q_k, X_\ell, D_m)$$

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

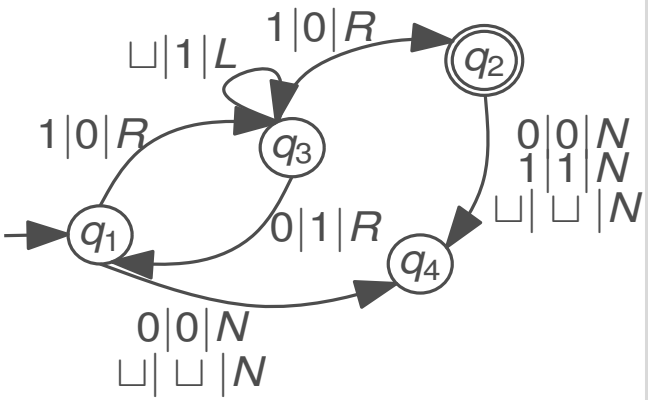
Universelle Turingmaschine

Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

$$\left.\begin{array}{l} X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup \\ D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N \end{array}\right\} \quad \delta(q_i, X_j) = (q_k, X_\ell, D_m) \quad \Big\} \quad 0^i\ 10^j\ 1\ 0^k\ 10^\ell 10^m$$

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

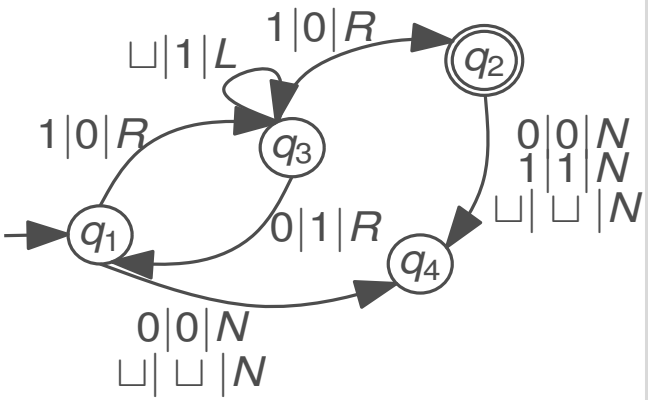
Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

$$\left.\begin{array}{l} X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup \\ D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N \end{array}\right\} \quad \delta(q_i, X_j) = (q_k, X_\ell, D_m) \quad \Big\} \quad 0^i\ 10^j\ 1\ 0^k\ 10^\ell 10^m$$

#	Eintrag	Kodierung

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

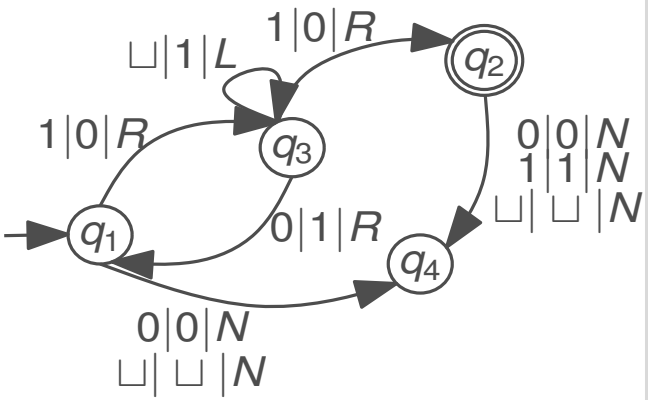
Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

$$\left.\begin{array}{l} X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup \\ D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N \end{array}\right\} \quad \delta(q_i, X_j) = (q_k, X_\ell, D_m) \quad \Big\} \quad 0^i\ 10^j\ 1\ 0^k\ 10^\ell 10^m$$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	
2	$\delta(q_1, 1) = (q_3, 0, R)$	
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	
4	$\delta(q_3, 0) = (q_1, 1, R)$	
5	$\delta(q_3, 1) = (q_2, 0, R)$	
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

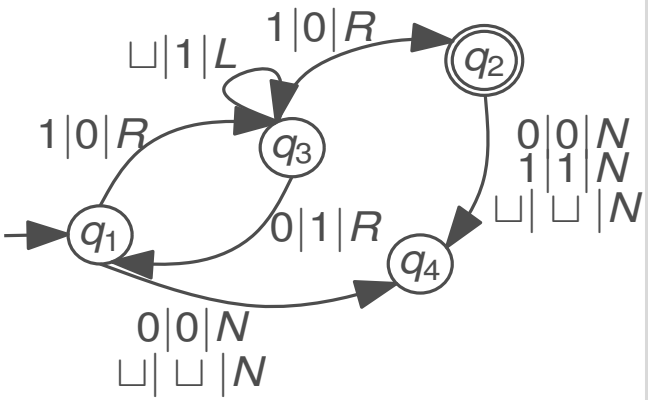
Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

$$\left.\begin{array}{l} X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup \\ D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N \end{array}\right\} \quad \delta(q_i, X_j) = (q_k, X_\ell, D_m) \quad \Big\} \quad 0^i\ 10^j\ 1\ 0^k\ 10^\ell 10^m$$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	01010000101000
2	$\delta(q_1, 1) = (q_3, 0, R)$	
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	
4	$\delta(q_3, 0) = (q_1, 1, R)$	
5	$\delta(q_3, 1) = (q_2, 0, R)$	
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

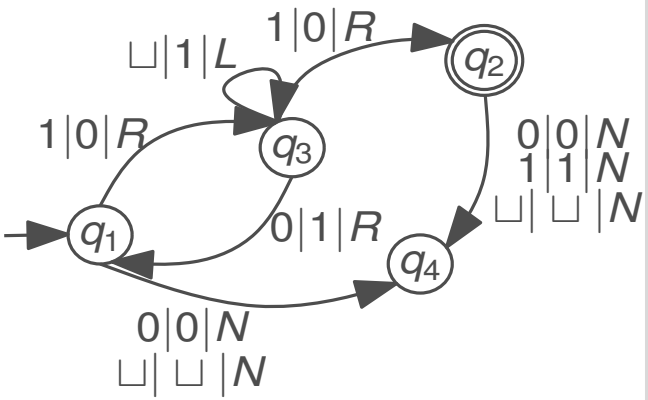
Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

$$\left.\begin{array}{l} X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup \\ D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N \end{array}\right\} \quad \delta(q_i, X_j) = (q_k, X_\ell, D_m) \quad 0^i 1 0^j 1 0^k 1 0^\ell 1 0^m$$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	01010000101000
2	$\delta(q_1, 1) = (q_3, 0, R)$	
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	
4	$\delta(q_3, 0) = (q_1, 1, R)$	
5	$\delta(q_3, 1) = (q_2, 0, R)$	
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

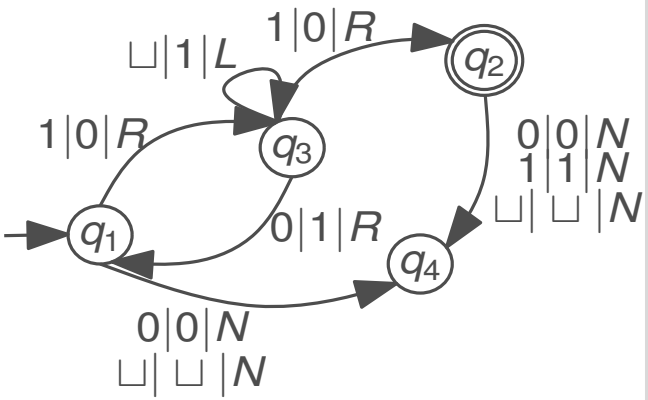
Kodierung:

$$X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup$$
$$D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N$$

$\Big\}$ $\delta(q_i, X_j) = (q_k, X_\ell, D_m)\Big\}$ $0^i\ 1\ 0^j\ 1\ 0^k\ 1\ 0^\ell\ 1\ 0^m$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	0101000010101000
2	$\delta(q_1, 1) = (q_3, 0, R)$	
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	
4	$\delta(q_3, 0) = (q_1, 1, R)$	
5	$\delta(q_3, 1) = (q_2, 0, R)$	
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

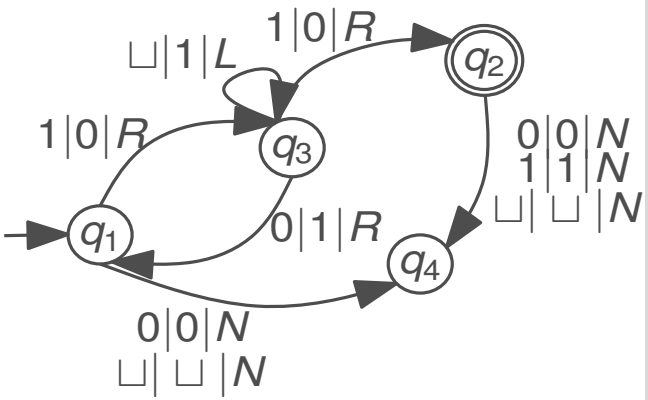
Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

$$\left.\begin{array}{l} X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup \\ D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N \end{array}\right\} \quad \delta(q_i, X_j) = (q_k, X_\ell, D_m) \quad 0^i\,1\,0^j\,1\,0^k\,1\,0^\ell\,1\,0^m$$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	0101000010100 0
2	$\delta(q_1, 1) = (q_3, 0, R)$	
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	
4	$\delta(q_3, 0) = (q_1, 1, R)$	
5	$\delta(q_3, 1) = (q_2, 0, R)$	
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

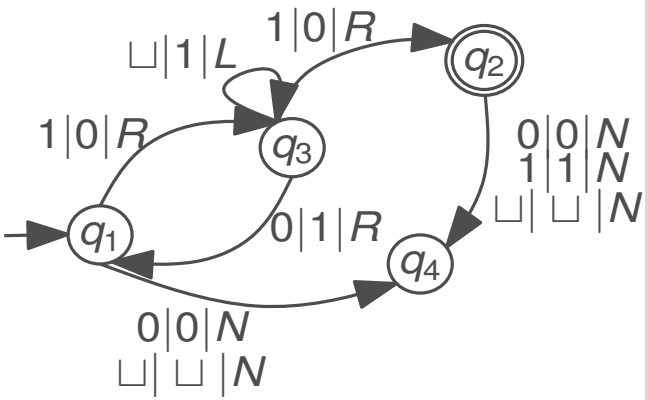
Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

$$\left.\begin{array}{l} X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup \\ D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N \end{array}\right\} \quad \delta(q_i, X_j) = (q_k, X_\ell, D_m) \quad 0^i 1 0^j 1 0^k 1 0^\ell 1 0^m$$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	0101000010101000
2	$\delta(q_1, 1) = (q_3, 0, R)$	
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	
4	$\delta(q_3, 0) = (q_1, 1, R)$	
5	$\delta(q_3, 1) = (q_2, 0, R)$	
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

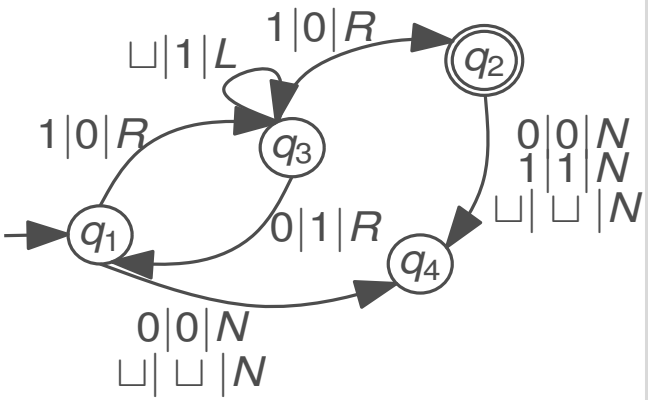
$X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup$

$D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N$

$\Big\}$ $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$ $\Big\}$ $0^i\,1\,0^j\,1\,0^k\,1\,0^\ell\,1\,0^m$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	0101000010101000
2	$\delta(q_1, 1) = (q_3, 0, R)$	
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	
4	$\delta(q_3, 0) = (q_1, 1, R)$	
5	$\delta(q_3, 1) = (q_2, 0, R)$	
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

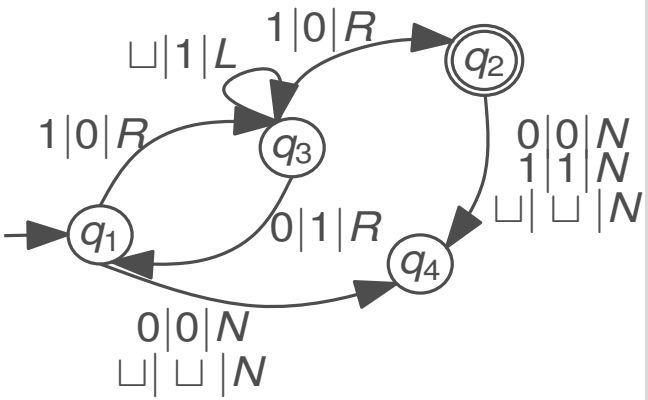
$X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup$

$D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N$

$\Big\}$ $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$ $\Big\}$ $0^i\ 1\ 0^j\ 1\ 0^k\ 1\ 0^\ell\ 1\ 0^m$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	0101000010101000
2	$\delta(q_1, 1) = (q_3, 0, R)$	0100100010100
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	
4	$\delta(q_3, 0) = (q_1, 1, R)$	
5	$\delta(q_3, 1) = (q_2, 0, R)$	
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

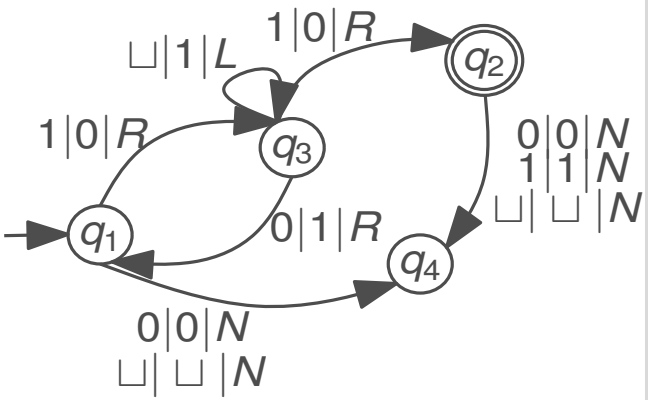
$X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup$

$D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N$

$\Big\}$ $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$ $\Big\}$ $0^i\,1\,0^j\,1\,0^k\,1\,0^\ell\,1\,0^m$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	0101000010101000
2	$\delta(q_1, 1) = (q_3, 0, R)$	0100100010100
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	0100010000100010000
4	$\delta(q_3, 0) = (q_1, 1, R)$	
5	$\delta(q_3, 1) = (q_2, 0, R)$	
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

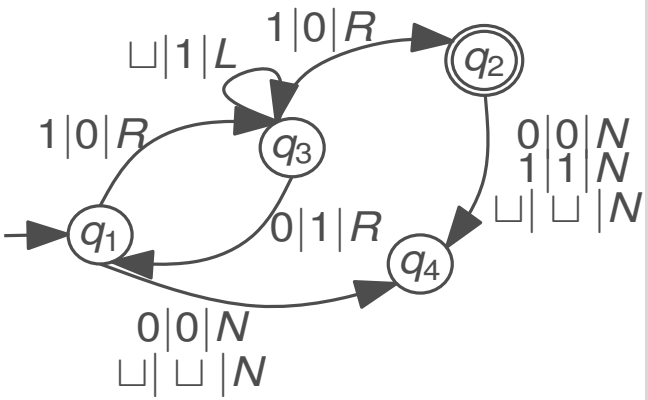
$X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup$

$D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N$

$\Big\}$ $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$ $\Big\}$ $0^i\,1\,0^j\,1\,0^k\,1\,0^\ell\,1\,0^m$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	0101000010101000
2	$\delta(q_1, 1) = (q_3, 0, R)$	0100100010100
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	0100010000100010000
4	$\delta(q_3, 0) = (q_1, 1, R)$	0001010000101000
5	$\delta(q_3, 1) = (q_2, 0, R)$	
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

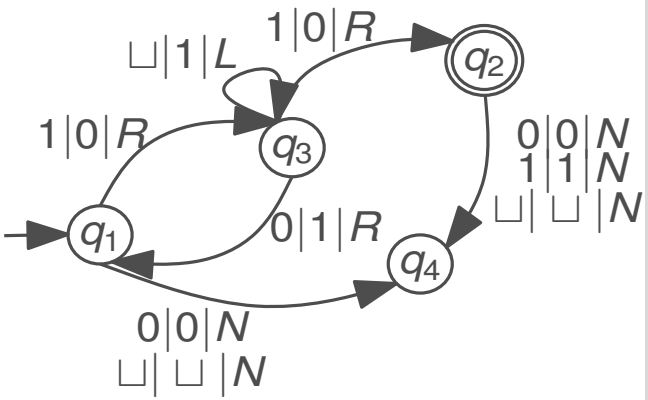
$X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup$

$D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N$

$\Big\}$ $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$ $\Big\}$ $0^i\,1\,0^j\,1\,0^k\,1\,0^\ell\,1\,0^m$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	0101000010101000
2	$\delta(q_1, 1) = (q_3, 0, R)$	0100100010100
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	0100010000100010000
4	$\delta(q_3, 0) = (q_1, 1, R)$	0001010000101000
5	$\delta(q_3, 1) = (q_2, 0, R)$	0001001001010100
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

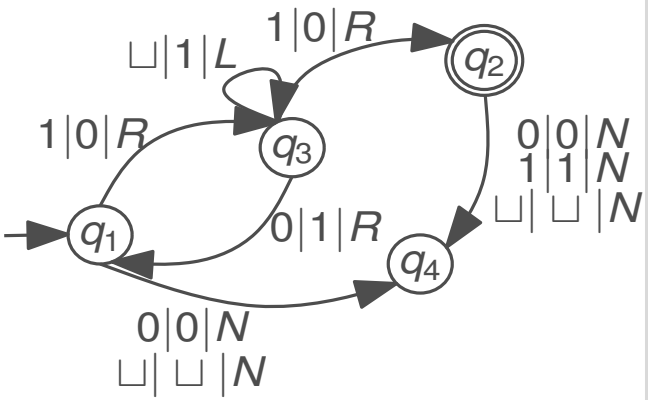
$X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup$

$D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N$

$\Big\}$ $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$ $\Big\}$ $0^i\,1\,0^j\,1\,0^k\,1\,0^\ell\,1\,0^m$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	0101000010101000
2	$\delta(q_1, 1) = (q_3, 0, R)$	0100100010100
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	0100010000100010000
4	$\delta(q_3, 0) = (q_1, 1, R)$	0001010000101000
5	$\delta(q_3, 1) = (q_2, 0, R)$	0001001001010100
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	0001000100010010

δ	0	1	$\sqcup$
q_1	$(q_4, 0, N)$	$(q_3, 0, R)$	$(q_4, \sqcup, N)$
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

$$\left.\begin{array}{l} X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup \\ D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N \end{array}\right\} \quad \left.\delta(q_i, X_j) = (q_k, X_\ell, D_m)\right\} \quad 0^i\ 10^j\ 1\ 0^k\ 10^\ell 10^m$$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	01010000101000
2	$\delta(q_1, 1) = (q_3, 0, R)$	0100100010100
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	0100010000100010000
4	$\delta(q_3, 0) = (q_1, 1, R)$	0001010000101000
5	$\delta(q_3, 1) = (q_2, 0, R)$	0001001001010 0
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	0001000100010010

$\langle \mathcal{M} \rangle$ = 111010100001010001101001000101001101000100001000100001
00010100001010001100010010010100110001000100010010111

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

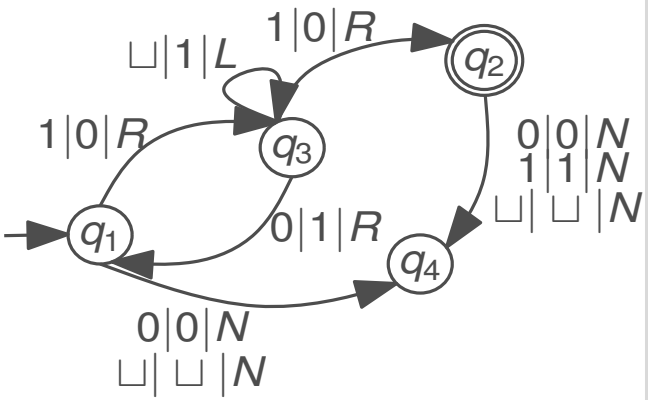
Gegeben ist folgende TM $\mathcal{M}$ mit $Q = \{q_1, q_2, q_3, q_f\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$, $s = q_1$, $F = \{q_2\}$ und δ. Was ist die Gödelnummer $\langle \mathcal{M} \rangle$ der TM?

Kodierung:

$$\left.\begin{array}{l} X_1 \mathrel{\hat{=}} 0, X_2 \mathrel{\hat{=}} 1, X_3 \mathrel{\hat{=}} \sqcup \\ D_1 \mathrel{\hat{=}} L, D_2 \mathrel{\hat{=}} R, D_3 \mathrel{\hat{=}} N \end{array}\right\} \quad \delta(q_i, X_j) = (q_k, X_\ell, D_m) \quad \Big\} \quad 0^i\ 10^j\ 1\ 0^k\ 10^\ell 10^m$$

#	Eintrag	Kodierung
1	$\delta(q_1, 0) = (q_4, 0, N)$	01010000101000
2	$\delta(q_1, 1) = (q_3, 0, R)$	0100100010100
3	$\delta(q_1, \sqcup) = (q_4, \sqcup, N)$	010001000010001000
4	$\delta(q_3, 0) = (q_1, 1, R)$	0001010000101000
5	$\delta(q_3, 1) = (q_2, 0, R)$	0001001001010 0
6	$\delta(q_3, \sqcup) = (q_3, 1, L)$	0001000100010010

$\langle \mathcal{M} \rangle$ = 14836565411238925247228547960232 7

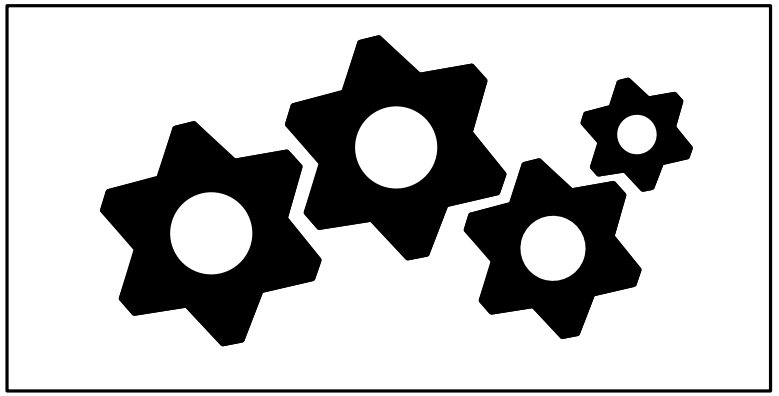


Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Definition Eine Turingmaschine $\mathcal{M}_0$ heißt universell, falls für jede 1-Band-DTM $\mathcal{M}$ und jedes $x \in \{0,1\}^\star$ gilt:

- $\mathcal{M}_0$ gestartet mit $\langle \mathcal{M} \rangle x$ hält genau dann, wenn $\mathcal{M}$ gestartet mit x hält.
- Falls $\mathcal{M}$ gestartet mit x hält, berechnet $\mathcal{M}_0$ gestartet mit $\langle \mathcal{M} \rangle x$ die gleiche Ausgabe wie $\mathcal{M}$ gestartet mit x. Insbesondere akzeptiert $\mathcal{M}_0$ die Eingabe $\langle \mathcal{M} \rangle x$ genau dann, wenn $\mathcal{M}$ die Eingabe x akzeptiert.



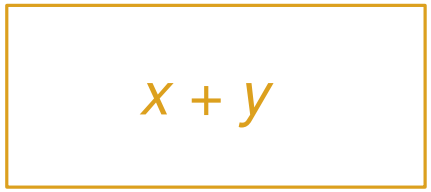
Universelle Turingmaschine simuliert $\mathcal{M}$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

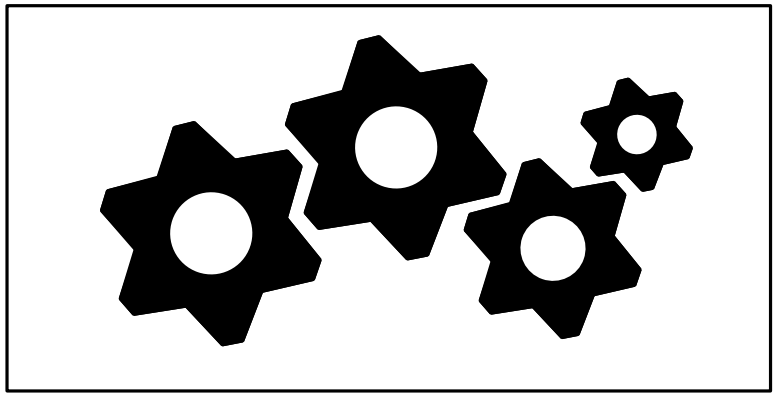
Universelle Turingmaschine

Definition Eine Turingmaschine $\mathcal{M}_0$ heißt universell, falls für jede 1-Band-DTM $\mathcal{M}$ und jedes $x \in \{0,1\}^\star$ gilt:

- $\mathcal{M}_0$ gestartet mit $\langle \mathcal{M} \rangle x$ hält genau dann, wenn $\mathcal{M}$ gestartet mit x hält.
- Falls $\mathcal{M}$ gestartet mit x hält, berechnet $\mathcal{M}_0$ gestartet mit $\langle \mathcal{M} \rangle x$ die gleiche Ausgabe wie $\mathcal{M}$ gestartet mit x. Insbesondere akzeptiert $\mathcal{M}_0$ die Eingabe $\langle \mathcal{M} \rangle x$ genau dann, wenn $\mathcal{M}$ die Eingabe x akzeptiert.



Spezielle Turingmaschine $\mathcal{M}$
z.B.: Addition zweier Zahlen



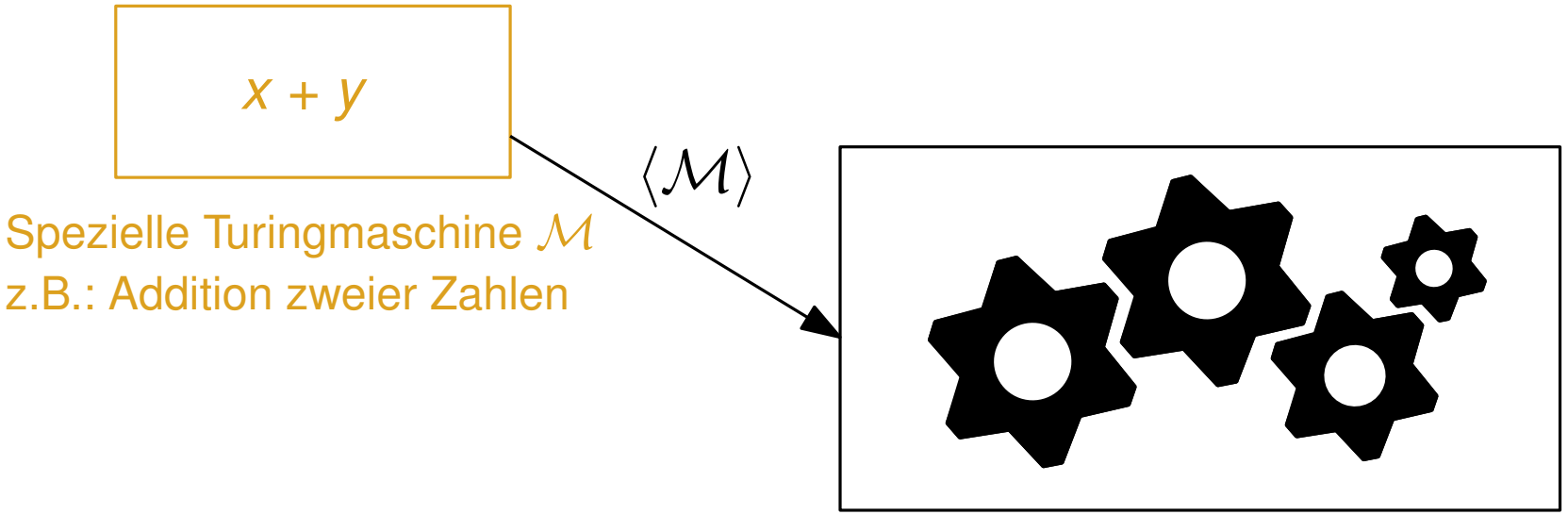
Universelle Turingmaschine simuliert $\mathcal{M}$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Definition Eine Turingmaschine $\mathcal{M}_0$ heißt universell, falls für jede 1-Band-DTM $\mathcal{M}$ und jedes $x \in \{0,1\}^\star$ gilt:

- $\mathcal{M}_0$ gestartet mit $\langle \mathcal{M} \rangle x$ hält genau dann, wenn $\mathcal{M}$ gestartet mit x hält.
- Falls $\mathcal{M}$ gestartet mit x hält, berechnet $\mathcal{M}_0$ gestartet mit $\langle \mathcal{M} \rangle x$ die gleiche Ausgabe wie $\mathcal{M}$ gestartet mit x. Insbesondere akzeptiert $\mathcal{M}_0$ die Eingabe $\langle \mathcal{M} \rangle x$ genau dann, wenn $\mathcal{M}$ die Eingabe x akzeptiert.



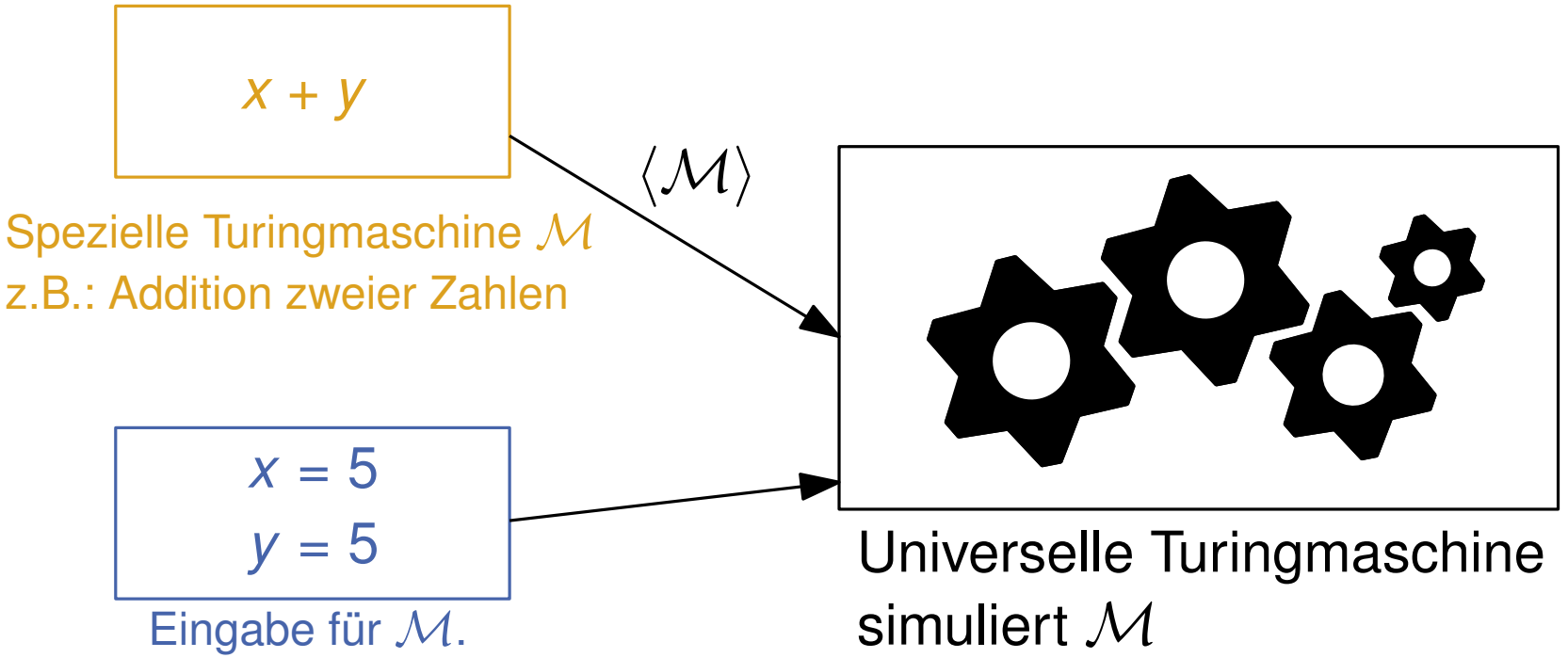
Spezielle Turingmaschine $\mathcal{M}$
z.B.: Addition zweier Zahlen

Universelle Turingmaschine simuliert $\mathcal{M}$

Universelle Turingmaschine

Definition Eine Turingmaschine $\mathcal{M}_0$ heißt universell, falls für jede 1-Band-DTM $\mathcal{M}$ und jedes $x \in \{0,1\}^\star$ gilt:

- $\mathcal{M}_0$ gestartet mit $\langle \mathcal{M} \rangle x$ hält genau dann, wenn $\mathcal{M}$ gestartet mit x hält.
- Falls $\mathcal{M}$ gestartet mit x hält, berechnet $\mathcal{M}_0$ gestartet mit $\langle \mathcal{M} \rangle x$ die gleiche Ausgabe wie $\mathcal{M}$ gestartet mit x. Insbesondere akzeptiert $\mathcal{M}_0$ die Eingabe $\langle \mathcal{M} \rangle x$ genau dann, wenn $\mathcal{M}$ die Eingabe x akzeptiert.

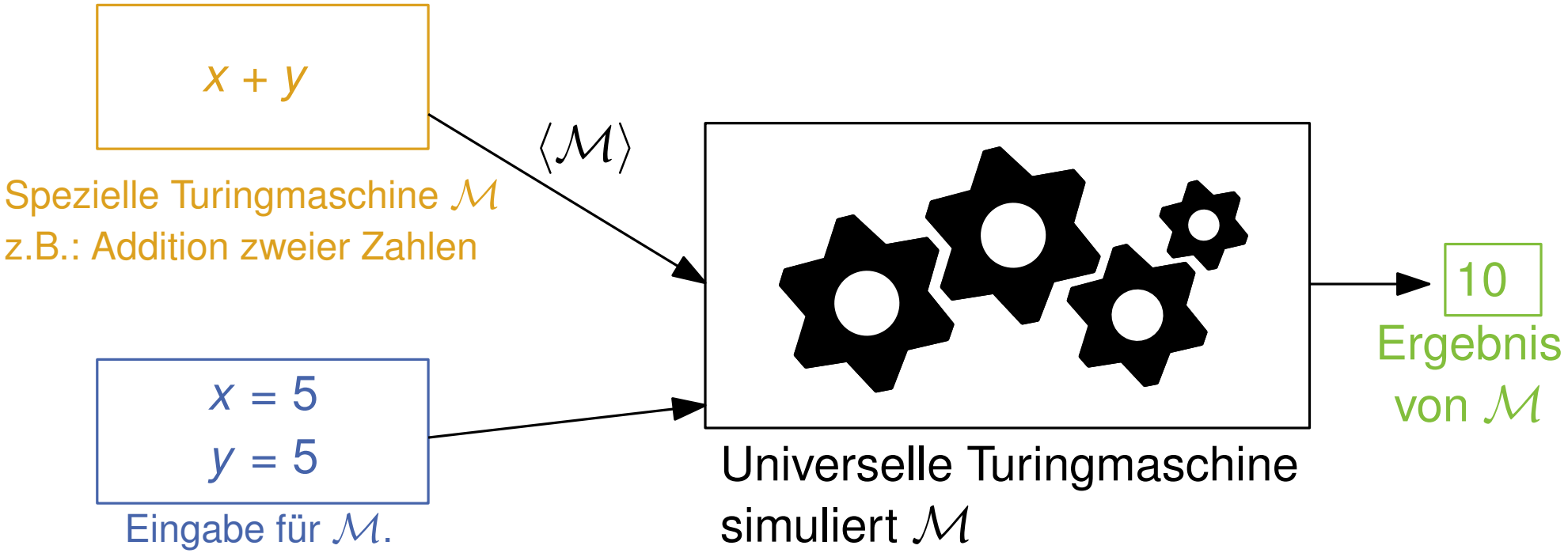


Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle Turingmaschine

Definition Eine Turingmaschine $\mathcal{M}_0$ heißt universell, falls für jede 1-Band-DTM $\mathcal{M}$ und jedes $x \in \{0,1\}^\star$ gilt:

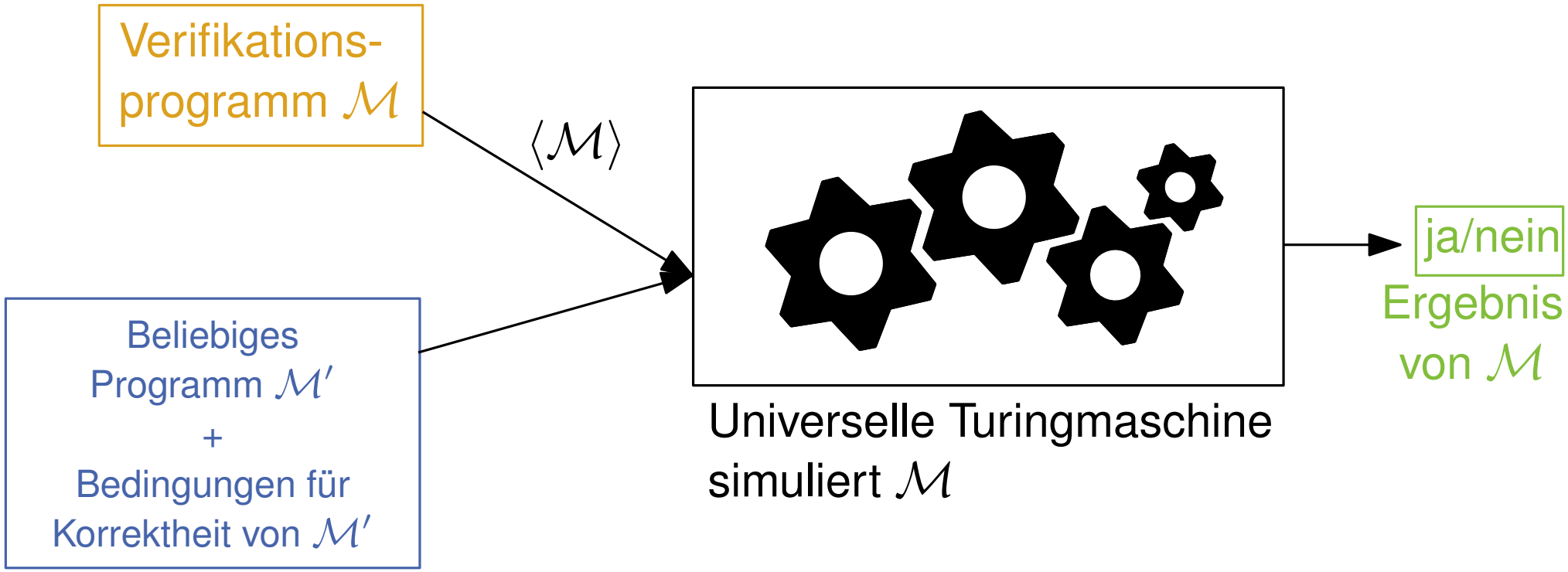
- $\mathcal{M}_0$ gestartet mit $\langle \mathcal{M} \rangle x$ hält genau dann, wenn $\mathcal{M}$ gestartet mit x hält.
- Falls $\mathcal{M}$ gestartet mit x hält, berechnet $\mathcal{M}_0$ gestartet mit $\langle \mathcal{M} \rangle x$ die gleiche Ausgabe wie $\mathcal{M}$ gestartet mit x. Insbesondere akzeptiert $\mathcal{M}_0$ die Eingabe $\langle \mathcal{M} \rangle x$ genau dann, wenn $\mathcal{M}$ die Eingabe x akzeptiert.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle TM – Beispiel

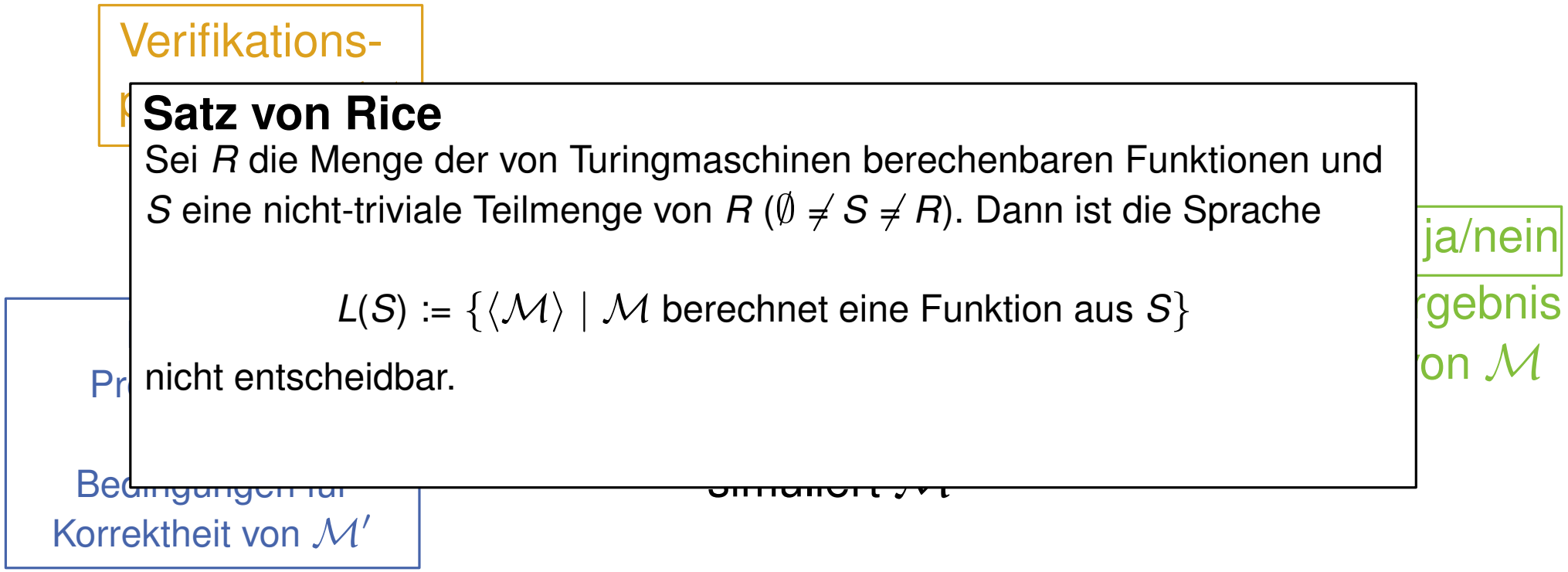
Gibt es eine Programm $\mathcal{M}$, dass für jedes beliebige Programm $\mathcal{M}'$ dessen Korrektheit beweist?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Universelle TM – Beispiel

Gibt es eine Programm $\mathcal{M}$, dass für jedes beliebige Programm $\mathcal{M}'$ dessen Korrektheit beweist?



Satz von Rice

Sei R die Menge der von Turingmaschinen berechenbaren Funktionen und S eine nicht-triviale Teilmenge von R ($\emptyset \neq S \neq R$). Dann ist die Sprache

$$L(S) := \{\langle \mathcal{M} \rangle \mid \mathcal{M} \text{ berechnet eine Funktion aus } S\}$$

nicht entscheidbar.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgaben zu Entscheidbarkeit

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

a) Die Menge der entscheidbaren Sprachen ist unter dem Kleene'schen Abschluss abgeschlossen, d.h. für jede entscheidbare Sprache L gilt, dass $L^\star$ auch entscheidbar ist.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

a) Die Menge der entscheidbaren Sprachen ist unter dem Kleene'schen Abschluss abgeschlossen, d.h. für jede entscheidbare Sprache L gilt, dass $L^\star$ auch entscheidbar ist.

Idee:

- Sei L eine entscheidbare Sprache.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

a) Die Menge der entscheidbaren Sprachen ist unter dem Kleene'schen Abschluss abgeschlossen, d.h. für jede entscheidbare Sprache L gilt, dass $L^\star$ auch entscheidbar ist.

Idee:

- Sei L eine entscheidbare Sprache.
- Es gibt also eine Turing-Maschine $\mathcal{M}$, die L entscheidet: $L(\mathcal{M}) = L$.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

a) Die Menge der entscheidbaren Sprachen ist unter dem Kleene'schen Abschluss abgeschlossen, d.h. für jede entscheidbare Sprache L gilt, dass $L^\star$ auch entscheidbar ist.

Idee:

- Sei L eine entscheidbare Sprache.
- Es gibt also eine Turing-Maschine $\mathcal{M}$, die L entscheidet: $L(\mathcal{M}) = L$.
- Konstruiere eine NTM $\mathcal{M}'$.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

a) Die Menge der entscheidbaren Sprachen ist unter dem Kleene'schen Abschluss abgeschlossen, d.h. für jede entscheidbare Sprache L gilt, dass $L^\star$ auch entscheidbar ist.

Idee:

- Sei L eine entscheidbare Sprache.
- Es gibt also eine Turing-Maschine $\mathcal{M}$, die L entscheidet: $L(\mathcal{M}) = L$.
- Konstruiere eine NTM $\mathcal{M}'$.

Verfahren: Sei x die Eingabe.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

a) Die Menge der entscheidbaren Sprachen ist unter dem Kleene'schen Abschluss abgeschlossen, d.h. für jede entscheidbare Sprache L gilt, dass $L^\star$ auch entscheidbar ist.

Idee:

- Sei L eine entscheidbare Sprache.
- Es gibt also eine Turing-Maschine $\mathcal{M}$, die L entscheidet: $L(\mathcal{M}) = L$.
- Konstruiere eine NTM $\mathcal{M}'$.

Verfahren: Sei x die Eingabe.

- Wähle nicht-deterministisch ein nicht-leeres Präfix π von x.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

a) Die Menge der entscheidbaren Sprachen ist unter dem Kleene'schen Abschluss abgeschlossen, d.h. für jede entscheidbare Sprache L gilt, dass $L^\star$ auch entscheidbar ist.

Idee:

- Sei L eine entscheidbare Sprache.
- Es gibt also eine Turing-Maschine $\mathcal{M}$, die L entscheidet: $L(\mathcal{M}) = L$.
- Konstruiere eine NTM $\mathcal{M}'$.

Verfahren: Sei x die Eingabe.

- Wähle nicht-deterministisch ein nicht-leeres Präfix π von x.
- Überprüfe mithilfe von $\mathcal{M}$, ob π in L liegt.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

a) Die Menge der entscheidbaren Sprachen ist unter dem Kleene'schen Abschluss abgeschlossen, d.h. für jede entscheidbare Sprache L gilt, dass $L^\star$ auch entscheidbar ist.

Idee:

- Sei L eine entscheidbare Sprache.
- Es gibt also eine Turing-Maschine $\mathcal{M}$, die L entscheidet: $L(\mathcal{M}) = L$.
- Konstruiere eine NTM $\mathcal{M}'$.

Verfahren: Sei x die Eingabe.

- Wähle nicht-deterministisch ein nicht-leeres Präfix π von x.
- Überprüfe mithilfe von $\mathcal{M}$, ob π in L liegt.

  Fall: π liegt nicht in $L \rightarrow M$ akzeptiert x nicht.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

a) Die Menge der entscheidbaren Sprachen ist unter dem Kleene'schen Abschluss abgeschlossen, d.h. für jede entscheidbare Sprache L gilt, dass $L^\star$ auch entscheidbar ist.

Idee:

- Sei L eine entscheidbare Sprache.
- Es gibt also eine Turing-Maschine $\mathcal{M}$, die L entscheidet: $L(\mathcal{M}) = L$.
- Konstruiere eine NTM $\mathcal{M}'$.

Verfahren: Sei x die Eingabe.

- Wähle nicht-deterministisch ein nicht-leeres Präfix π von x.
- Überprüfe mithilfe von $\mathcal{M}$, ob π in L liegt.

  Fall: π liegt nicht in $L \rightarrow M$ akzeptiert x nicht.

  Fall: π liegt in L: M löscht π vom Band. Falls das Band nun leer ist, *akzeptiert* M die Eingabe x. Sonst wiederhole Verfahren.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

a) Die Menge der entscheidbaren Sprachen ist unter dem Kleene'schen Abschluss abgeschlossen, d.h. für jede entscheidbare Sprache L gilt, dass $L^\star$ auch entscheidbar ist.

Idee:

- Sei L eine entscheidbare Sprache.
- Es gibt also eine Turing-Maschine $\mathcal{M}$, die L entscheidet: $L(\mathcal{M}) = L$.
- Konstruiere eine NTM $\mathcal{M}'$.

Verfahren: Sei x die Eingabe.

- Wähle nicht-deterministisch ein nicht-leeres Präfix π von x.
- Überprüfe mithilfe von $\mathcal{M}$, ob π in L liegt.

  Fall: π liegt nicht in $L \rightarrow M$ akzeptiert x nicht.

  Fall: π liegt in L: M löscht π vom Band. Falls das Band nun leer ist, *akzeptiert* M die Eingabe x. Sonst wiederhole Verfahren.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

b) Die Menge der entscheidbaren Sprachen ist bzgl. der Operation min abgeschlossen, d.h. für jede entscheidbare Sprache L gilt, dass $\min(L)$ auch entscheidbar ist. Die Operation min ist für eine entscheidbare Sprache L definiert als

$$\min(L) := \{x \in L \mid \text{kein echtes Präfix von } x \text{ ist in L}\}$$

Hinweis: Ein Präfix von x heißt *echt*, wenn es nicht x ist.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

b) Die Menge der entscheidbaren Sprachen ist bzgl. der Operation min abgeschlossen mit

$$\min(L) := \{x \in L \mid \text{kein echtes Präfix von } x \text{ ist in L}\}$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

b) Die Menge der entscheidbaren Sprachen ist bzgl. der Operation min abgeschlossen mit

$$\min(L) := \{x \in L \mid \text{kein echtes Präfix von } x \text{ ist in L}\}$$

Idee:

- Die TM T_L entscheidet die Sprache $L \subseteq \Sigma^\star$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

b) Die Menge der entscheidbaren Sprachen ist bzgl. der Operation min abgeschlossen mit

$$\min(L) := \{x \in L \mid \text{kein echtes Präfix von } x \text{ ist in L}\}$$

Idee:

- Die TM T_L entscheidet die Sprache $L \subseteq \Sigma^\star$
- T' generiert alle echten Präfixe ihrer Eingabe ohne Wiederholung.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

b) Die Menge der entscheidbaren Sprachen ist bzgl. der Operation min abgeschlossen mit

$$\min(L) := \{x \in L \mid \text{kein echtes Präfix von } x \text{ ist in L}\}$$

Idee:

- Die TM T_L entscheidet die Sprache $L \subseteq \Sigma^\star$
- T' generiert alle echten Präfixe ihrer Eingabe ohne Wiederholung.

Arbeitsweise der TM T, die $\min(L)$ entscheidet mit der Eingabe x.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

b) Die Menge der entscheidbaren Sprachen ist bzgl. der Operation min abgeschlossen mit

$$\min(L) := \{x \in L \mid \text{kein echtes Präfix von } x \text{ ist in L}\}$$

Idee:

- Die TM T_L entscheidet die Sprache $L \subseteq \Sigma^\star$
- T' generiert alle echten Präfixe ihrer Eingabe ohne Wiederholung.

Arbeitsweise der TM T, die $\min(L)$ entscheidet mit der Eingabe x.

1. T_L entscheidet x.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

b) Die Menge der entscheidbaren Sprachen ist bzgl. der Operation min abgeschlossen mit

$$\min(L) := \{x \in L \mid \text{kein echtes Präfix von } x \text{ ist in L}\}$$

Idee:

- Die TM T_L entscheidet die Sprache $L \subseteq \Sigma^\star$
- T' generiert alle echten Präfixe ihrer Eingabe ohne Wiederholung.

Arbeitsweise der TM T, die $\min(L)$ entscheidet mit der Eingabe x.

1. T_L entscheidet x.
2. Wenn T_L nicht akzeptiert, hält T und akzeptiert nicht.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

b) Die Menge der entscheidbaren Sprachen ist bzgl. der Operation min abgeschlossen mit

$$\min(L) := \{x \in L \mid \text{kein echtes Präfix von } x \text{ ist in L}\}$$

Idee:

- Die TM T_L entscheidet die Sprache $L \subseteq \Sigma^\star$
- T' generiert alle echten Präfixe ihrer Eingabe ohne Wiederholung.

Arbeitsweise der TM T, die $\min(L)$ entscheidet mit der Eingabe x.

1. T_L entscheidet x.
2. Wenn T_L nicht akzeptiert, hält T und akzeptiert nicht.
3. T' generiert das nächste echte Präfix p von x.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

b) Die Menge der entscheidbaren Sprachen ist bzgl. der Operation min abgeschlossen mit

$$\min(L) := \{x \in L \mid \text{kein echtes Präfix von } x \text{ ist in L}\}$$

Idee:

- Die TM T_L entscheidet die Sprache $L \subseteq \Sigma^\star$
- T' generiert alle echten Präfixe ihrer Eingabe ohne Wiederholung.

Arbeitsweise der TM T, die $\min(L)$ entscheidet mit der Eingabe x.

1. T_L entscheidet x.
2. Wenn T_L nicht akzeptiert, hält T und akzeptiert nicht.
3. T' generiert das nächste echte Präfix p von x.
4. Es gibt kein weiteres Präfix mehr: T akzeptiert.

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

b) Die Menge der entscheidbaren Sprachen ist bzgl. der Operation min abgeschlossen mit

$$\min(L) := \{x \in L \mid \text{kein echtes Präfix von } x \text{ ist in L}\}$$

Idee:

- Die TM T_L entscheidet die Sprache $L \subseteq \Sigma^\star$
- T' generiert alle echten Präfixe ihrer Eingabe ohne Wiederholung.

Arbeitsweise der TM T, die $\min(L)$ entscheidet mit der Eingabe x.

1. T_L entscheidet x.
2. Wenn T_L nicht akzeptiert, hält T und akzeptiert nicht.
3. T' generiert das nächste echte Präfix p von x.
4. Es gibt kein weiteres Präfix mehr: T akzeptiert.
5. T_L entscheidet p.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

b) Die Menge der entscheidbaren Sprachen ist bzgl. der Operation min abgeschlossen mit

$$\min(L) := \{x \in L \mid \text{kein echtes Präfix von } x \text{ ist in L}\}$$

Idee:

- Die TM T_L entscheidet die Sprache $L \subseteq \Sigma^\star$
- T' generiert alle echten Präfixe ihrer Eingabe ohne Wiederholung.

Arbeitsweise der TM T, die $\min(L)$ entscheidet mit der Eingabe x.

1. T_L entscheidet x.
2. Wenn T_L nicht akzeptiert, hält T und akzeptiert nicht.
3. T' generiert das nächste echte Präfix p von x.
4. Es gibt kein weiteres Präfix mehr: T akzeptiert.
5. T_L entscheidet p.
6. Wenn $p \in L$, dann hält T und akzeptiert nicht.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

b) Die Menge der entscheidbaren Sprachen ist bzgl. der Operation min abgeschlossen mit

$$\min(L) := \{x \in L \mid \text{kein echtes Präfix von } x \text{ ist in L}\}$$

Idee:

- Die TM T_L entscheidet die Sprache $L \subseteq \Sigma^\star$
- T' generiert alle echten Präfixe ihrer Eingabe ohne Wiederholung.

Arbeitsweise der TM T, die $\min(L)$ entscheidet mit der Eingabe x.

1. T_L entscheidet x.
2. Wenn T_L nicht akzeptiert, hält T und akzeptiert nicht.
3. T' generiert das nächste echte Präfix p von x.
4. Es gibt kein weiteres Präfix mehr: T akzeptiert.
5. T_L entscheidet p.
6. Wenn $p \in L$, dann hält T und akzeptiert nicht.

sonst

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

b) Die Menge der entscheidbaren Sprachen ist bzgl. der Operation min abgeschlossen mit

$$\min(L) := \{x \in L \mid \text{kein echtes Präfix von } x \text{ ist in L}\}$$

Idee:

- Die TM T_L entscheidet die Sprache $L \subseteq \Sigma^\star$
- T' generiert alle echten Präfixe ihrer Eingabe ohne Wiederholung.

Arbeitsweise der TM T, die $\min(L)$ entscheidet mit der Eingabe x.

1. T_L entscheidet x.
2. Wenn T_L nicht akzeptiert, hält T und akzeptiert nicht.
3. T' generiert das nächste echte Präfix p von x.
4. Es gibt kein weiteres Präfix mehr: T akzeptiert.
5. T_L entscheidet p.
6. Wenn $p \in L$, dann hält T und akzeptiert nicht.

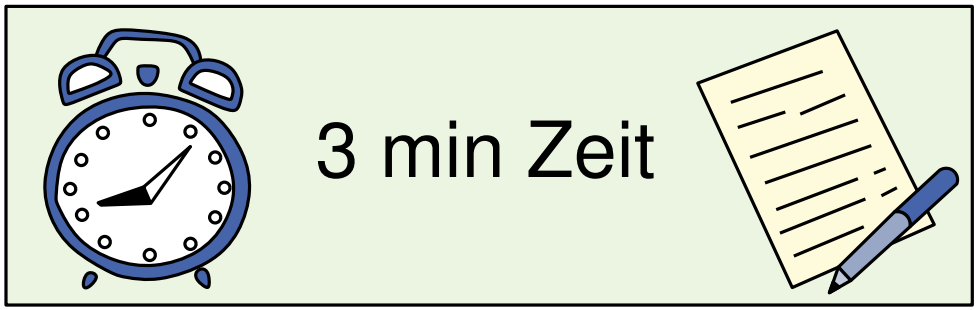
sonst

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

(c) Die Menge der semi-entscheidbaren Sprachen ist unter Komplementbildung nicht abgeschlossen.



Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

(c) Die Menge der semi-entscheidbaren Sprachen ist unter Komplementbildung nicht abgeschlossen.

Idee:

- Verwende universelle Sprache $L_u := \{wv \mid v \in L(T_w)\}$.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

(c) Die Menge der semi-entscheidbaren Sprachen ist unter Komplementbildung nicht abgeschlossen.

Idee:

- Verwende universelle Sprache $L_u := \{wv \mid v \in L(T_w)\}$.

Aus der Vorlesung:

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

(c) Die Menge der semi-entscheidbaren Sprachen ist unter Komplementbildung nicht abgeschlossen.

Idee:

- Verwende universelle Sprache $L_u := \{wv \mid v \in L(T_w)\}$.

Aus der Vorlesung:

(a) L_u ist semi-entscheidbar (Satz 3.15 im Skript, Seite 46).

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

(c) Die Menge der semi-entscheidbaren Sprachen ist unter Komplementbildung nicht abgeschlossen.

Idee:

- Verwende universelle Sprache $L_u := \{wv \mid v \in L(T_w)\}$.

Aus der Vorlesung:

a) L_u ist semi-entscheidbar (Satz 3.15 im Skript, Seite 46).

b) L_u ist nicht entscheidbar (Satz 3.14 im Sktipt, Seite 45).

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

(c) Die Menge der semi-entscheidbaren Sprachen ist unter Komplementbildung nicht abgeschlossen.

Idee:

- Verwende universelle Sprache $L_u := \{wv \mid v \in L(T_w)\}$.

Aus der Vorlesung:

(a) L_u ist semi-entscheidbar (Satz 3.15 im Skript, Seite 46).

(b) L_u ist nicht entscheidbar (Satz 3.14 im Sktipt, Seite 45).

(c) Für eine Sprache L gilt L und L^c sind semi-entscheidbar gdw. L ist entscheidbar (Vorlesung).

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

(c) Die Menge der semi-entscheidbaren Sprachen ist unter Komplementbildung nicht abgeschlossen.

Idee:

- Verwende universelle Sprache $L_u := \{wv \mid v \in L(T_w)\}$.

Aus der Vorlesung:

(a) L_u ist semi-entscheidbar (Satz 3.15 im Skript, Seite 46).

(b) L_u ist nicht entscheidbar (Satz 3.14 im Sktipt, Seite 45).

(c) Für eine Sprache L gilt L und L^c sind semi-entscheidbar gdw. L ist entscheidbar (Vorlesung).

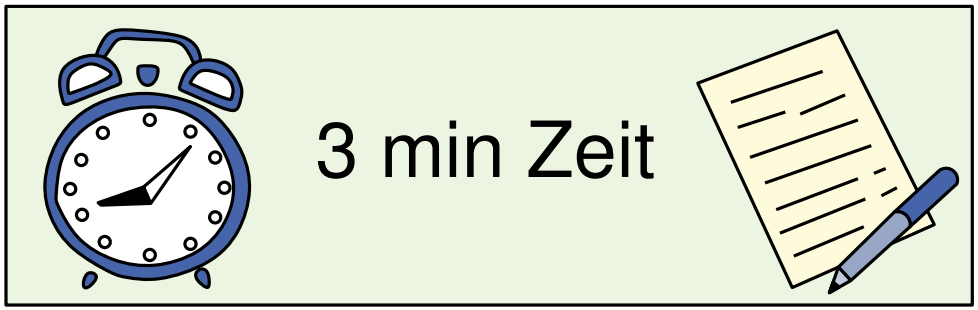
Annahme: L_u^c ist semi-entscheidbar. $\Rightarrow$ Da L_u semi-entscheidbar ist, wäre damit L_u entscheidbar, im Widerspruch zu L_u ist nicht entscheidbar.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

d) Die Menge der entscheidbaren Sprachen ist unter Vereinigung und Schnitt abgeschlossen.
Hinweis: Die Menge der entscheidbaren Sprachen ist unter Komplementbildung abgeschlossen



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

e) Die Menge der entscheidbaren Sprachen ist unter Vereinigung und Schnitt abgeschlossen.

1. Vereinigung

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

e) Die Menge der entscheidbaren Sprachen ist unter Vereinigung und Schnitt abgeschlossen.

1. Vereinigung

Sei $\mathcal{M}_1$ TM, die L_1 entscheidet, also $L_1 = L(\mathcal{M}_1)$ und sei $\mathcal{M}_2$ TM, die L_2 entscheidet, also $L_2 = L(\mathcal{M}_2)$.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

e) Die Menge der entscheidbaren Sprachen ist unter Vereinigung und Schnitt abgeschlossen.

1. Vereinigung

Sei $\mathcal{M}_1$ TM, die L_1 entscheidet, also $L_1 = L(\mathcal{M}_1)$ und sei $\mathcal{M}_2$ TM, die L_2 entscheidet, also $L_2 = L(\mathcal{M}_2)$.

Benutze 2-Band-TM $\mathcal{M}'$ mit einem Kopf (Skript, Seite 41/42):

- Simuliere $\mathcal{M}_1$ auf Band 1
- Simuliere $\mathcal{M}_2$ auf Band 2

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

e) Die Menge der entscheidbaren Sprachen ist unter Vereinigung und Schnitt abgeschlossen.

1. Vereinigung

Sei $\mathcal{M}_1$ TM, die L_1 entscheidet, also $L_1 = L(\mathcal{M}_1)$ und sei $\mathcal{M}_2$ TM, die L_2 entscheidet, also $L_2 = L(\mathcal{M}_2)$.

Benutze 2-Band-TM $\mathcal{M}'$ mit einem Kopf (Skript, Seite 41/42):

- Simuliere $\mathcal{M}_1$ auf Band 1
- Simuliere $\mathcal{M}_2$ auf Band 2

Falls $\mathcal{M}_1$ auf Band 1 akzeptiert $\Rightarrow$ akzeptiere Eingabe

Sonst Wechsel auf Band 2. Falls $\mathcal{M}_2$ auf Band 2 akzeptiert $\Rightarrow$ akzeptiere Eingabe

Sonst stopp in nicht-akzeptierendem Zustand.

Für genau die Eingaben in $L_1 \cup L_2$ tritt ein akzeptierender Fall ein!

$\Rightarrow$ $\mathcal{M}'$ akzeptiert $L_1 \cup L_2$ und stoppt immer, Vereinigung ist entscheidbar

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

e) Die Menge der entscheidbaren Sprachen ist unter Vereinigung und Schnitt abgeschlossen.

1. Vereinigung

Sei $\mathcal{M}_1$ TM, die L_1 entscheidet, also $L_1 = L(\mathcal{M}_1)$ und sei $\mathcal{M}_2$ TM, die L_2 entscheidet, also $L_2 = L(\mathcal{M}_2)$.

Benutze 2-Band-TM $\mathcal{M}'$ mit einem Kopf (Skript, Seite 41/42):

- Simuliere $\mathcal{M}_1$ auf Band 1
- Simuliere $\mathcal{M}_2$ auf Band 2

Falls $\mathcal{M}_1$ auf Band 1 akzeptiert $\Rightarrow$ akzeptiere Eingabe

Sonst Wechsel auf Band 2. Falls $\mathcal{M}_2$ auf Band 2 akzeptiert $\Rightarrow$ akzeptiere Eingabe

Sonst stopp in nicht-akzeptierendem Zustand.

Für genau die Eingaben in $L_1 \cup L_2$ tritt ein akzeptierender Fall ein!

$\Rightarrow$ $\mathcal{M}'$ akzeptiert $L_1 \cup L_2$ und stoppt immer, Vereinigung ist entscheidbar

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

e) Die Menge der entscheidbaren Sprachen ist unter Vereinigung und Schnitt abgeschlossen.

2. Schnitt

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

e) Die Menge der entscheidbaren Sprachen ist unter Vereinigung und Schnitt abgeschlossen.

2. Schnitt

Verwende De Morgan Gesetz für Mengen $(\overline{A} \cup \overline{B}) = (\overline{A \cap B})$. Betrachte entscheidbare Sprachen L_1, L_2 als Mengen $A := L_1, B := L_2$.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

e) Die Menge der entscheidbaren Sprachen ist unter Vereinigung und Schnitt abgeschlossen.

2. Schnitt

Verwende De Morgan Gesetz für Mengen $(\overline{A} \cup \overline{B}) = (\overline{A \cap B})$. Betrachte entscheidbare Sprachen L_1, L_2 als Mengen $A := L_1, B := L_2$.

Dann gilt $\overline{A} = L_1^c$ und $\overline{B} = L_2^c$ und $(\overline{A} \cup \overline{B}) = (L_1^c \cup L_2^c) = (L_1 \cap L_2)^c = (\overline{A \cap B})$.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

e) Die Menge der entscheidbaren Sprachen ist unter Vereinigung und Schnitt abgeschlossen.

2. Schnitt

Verwende De Morgan Gesetz für Mengen $(\overline{A} \cup \overline{B}) = (\overline{A \cap B})$. Betrachte entscheidbare Sprachen L_1, L_2 als Mengen $A := L_1, B := L_2$.

Dann gilt $\overline{A} = L_1^c$ und $\overline{B} = L_2^c$ und $(\overline{A} \cup \overline{B}) = (L_1^c \cup L_2^c) = (L_1 \cap L_2)^c = (\overline{A \cap B})$.

$\Rightarrow (L_1 \cap L_2)^c$ entscheidbar
$\Rightarrow ((L_1 \cap L_2)^c)^c = L_1 \cap L_2$ entscheidbar

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Abgeschlossenheit von entscheidbaren Sprachen

Zeigen Sie:

e) Die Menge der entscheidbaren Sprachen ist unter Vereinigung und Schnitt abgeschlossen.

2. Schnitt

Verwende De Morgan Gesetz für Mengen $(\overline{A} \cup \overline{B}) = (\overline{A \cap B})$. Betrachte entscheidbare Sprachen L_1, L_2 als Mengen $A := L_1, B := L_2$.

Dann gilt $\overline{A} = L_1^c$ und $\overline{B} = L_2^c$ und $(\overline{A} \cup \overline{B}) = (L_1^c \cup L_2^c) = (L_1 \cap L_2)^c = (\overline{A \cap B})$.

$\Rightarrow (L_1 \cap L_2)^c$ entscheidbar
$\Rightarrow ((L_1 \cap L_2)^c)^c = L_1 \cap L_2$ entscheidbar

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Entscheidbarkeit

Sei

$$L = \{1^n \mid 1^n \text{ ist Teilwort der Dezimaldarstellung von } \pi\}.$$

Zeigen Sie, dass L entscheidbar ist.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

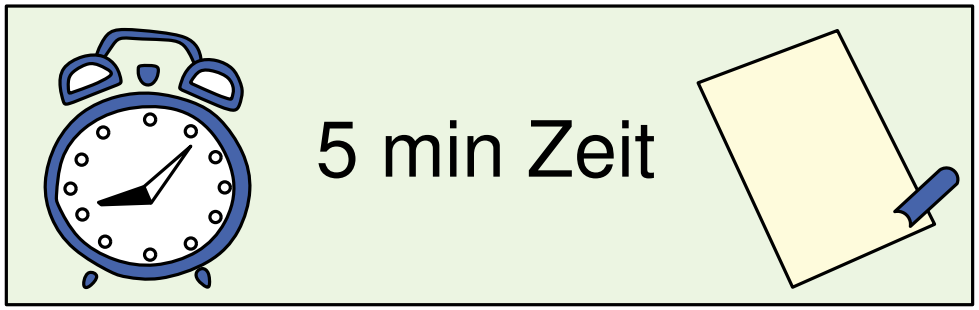
Aufgabe – Entscheidbarkeit

Sei

$$L = \{1^n \mid 1^n \text{ ist Teilwort der Dezimaldarstellung von } \pi\}.$$

Zeigen Sie, dass L entscheidbar ist.

Hinweis: Es ist nicht bekannt, für welche n die Dezimaldarstellung von π das Teilwort 1^n enthält, aber das ist hier auch nicht wichtig!



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Entscheidbarkeit

Sei

$$L = \{1^n \mid 1^n \text{ ist Teilwort der Dezimaldarstellung von } \pi\}.$$

Zeigen Sie, dass L entscheidbar ist.

Beweis: unterscheide zwei Fälle:

- die Dezimaldarstellung von π enthält 1^n für jedes natürliche n

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Entscheidbarkeit

Sei

$$L = \{1^n \mid 1^n \text{ ist Teilwort der Dezimaldarstellung von } \pi\}.$$

Zeigen Sie, dass L entscheidbar ist.

Beweis: unterscheide zwei Fälle:

- die Dezimaldarstellung von π enthält 1^n für jedes natürliche n
  - → dann akzeptiert die TM, die jedes Wort, das nur aus dem Zeichen 1 besteht, gerade die Sprache L

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Entscheidbarkeit

Sei

$$L = \{1^n \mid 1^n \text{ ist Teilwort der Dezimaldarstellung von } \pi\}.$$

Zeigen Sie, dass L entscheidbar ist.

Beweis: unterscheide zwei Fälle:

- die Dezimaldarstellung von π enthält 1^n für jedes natürliche n
  - → dann akzeptiert die TM, die jedes Wort, das nur aus dem Zeichen 1 besteht, gerade die Sprache L

- es gibt ein maximales $\hat{n}$, so dass die Dezimaldarstellung von π das Wort $1^{\hat{n}}$ enthält, das Wort $1^{\hat{n}+1}$ aber nicht

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Entscheidbarkeit

Sei

$$L = \{1^n \mid 1^n \text{ ist Teilwort der Dezimaldarstellung von } \pi\}.$$

Zeigen Sie, dass L entscheidbar ist.

Beweis: unterscheide zwei Fälle:

- die Dezimaldarstellung von π enthält 1^n für jedes natürliche n
  - → dann akzeptiert die TM, die jedes Wort, das nur aus dem Zeichen 1 besteht, gerade die Sprache L

- es gibt ein maximales $\hat{n}$, so dass die Dezimaldarstellung von π das Wort $1^{\hat{n}}$ enthält, das Wort $1^{\hat{n}+1}$ aber nicht
  - → dann akz. die TM, die jedes Wort, das nur aus dem Zeichen 1 besteht und Länge max. $\hat{n}$ hat, gerade die Sprache L

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Entscheidbarkeit

Sei

$$L = \{1^n \mid 1^n \text{ ist Teilwort der Dezimaldarstellung von } \pi\}.$$

Zeigen Sie, dass L entscheidbar ist.

Beweis: unterscheide zwei Fälle:

- die Dezimaldarstellung von π enthält 1^n für jedes natürliche n
- es gibt ein maximales $\hat{n}$, so dass die Dezimaldarstellung von π das Wort $1^{\hat{n}}$ enthält, das Wort $1^{\hat{n}+1}$ aber nicht

- Es ist für den Beweis egal, dass wir nicht wissen, welcher der beiden Fälle zutrifft.
- Die Sprache L ist sogar regulär!

Aufgabe – Entscheidbarkeit

Das Halteproblem definiert folgende Sprache:

$$\mathcal{H} = \{\langle w, v\rangle \mid T_w \text{ hält auf der Eingabe } v\}$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Entscheidbarkeit

Das Halteproblem definiert folgende Sprache:

$$\mathcal{H} = \{\langle w, v\rangle \mid T_w \text{ hält auf der Eingabe } v\}$$

fehlerhafter Beweisversuch, dass $\mathcal{H}$ entscheidbar ist:

Unterscheide zwei Fälle, wie bei der letzten Aufgabe:

- T_w hält auf der Eingabe v
  → dann liefert die TM, die alles akzeptiert, die richtige Antwort
- T_w stoppt bei Eingabe v niemals
  → dann liefert die TM, die alles ablehnt, die richtige Antwort

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Entscheidbarkeit

Das Halteproblem definiert folgende Sprache:

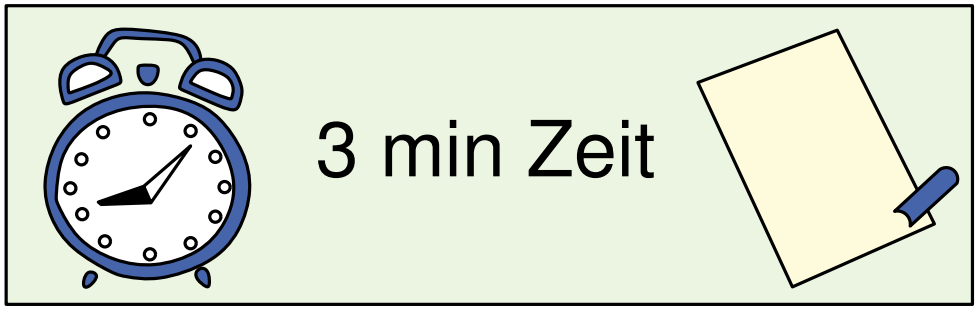
$$\mathcal{H} = \{\langle w, v\rangle \mid T_w \text{ hält auf der Eingabe } v\}$$

fehlerhafter Beweisversuch, dass $\mathcal{H}$ entscheidbar ist:

Unterscheide zwei Fälle, wie bei der letzten Aufgabe:

- T_w hält auf der Eingabe v
  → dann liefert die TM, die alles akzeptiert, die richtige Antwort
- T_w stoppt bei Eingabe v niemals
  → dann liefert die TM, die alles ablehnt, die richtige Antwort

Wieso ist dieser Beweis nicht korrekt?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Entscheidbarkeit

Das Halteproblem definiert folgende Sprache:

$$\mathcal{H} = \{\langle w, v\rangle \mid T_w \text{ hält auf der Eingabe } v\}$$

fehlerhafter Beweisversuch, dass $\mathcal{H}$ entscheidbar ist:

Unterscheide zwei Fälle, wie bei der letzten Aufgabe:

- T_w hält auf der Eingabe v
  → dann liefert die TM, die alles akzeptiert, die richtige Antwort
- T_w stoppt bei Eingabe v niemals
  → dann liefert die TM, die alles ablehnt, die richtige Antwort

Wieso ist dieser Beweis nicht korrekt?

Es ist nicht von **einer** TM **immer** entscheidbar, welcher Fall zutrifft.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

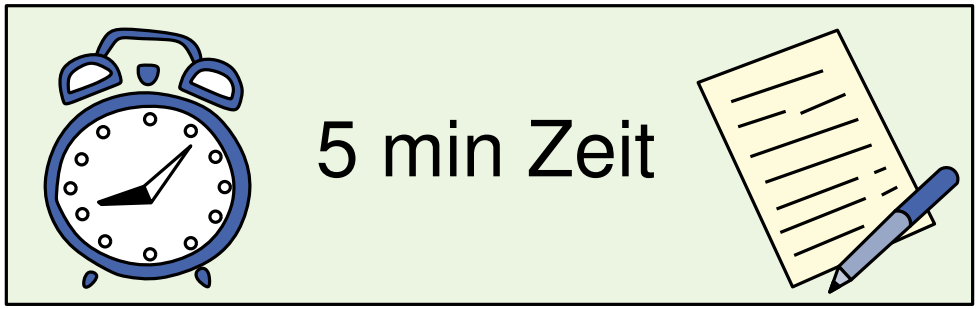
Klausuraufgabe – Entscheidbarkeit

Zeigen Sie, dass die Sprache

$$L = \{(u, v) \mid w \in L(T_u) \text{ genau dann, wenn } w^R \in L(T_v)\}$$

nicht entscheidbar ist.

Verwenden Sie nicht den Satz von Rice.



Klausuraufgabe – Entscheidbarkeit

Zeigen Sie, dass die Sprache

$$L = \{(u, v) \mid w \in L(T_u) \text{ genau dann, wenn } w^R \in L(T_v)\}$$

nicht entscheidbar ist.

Löse das Halteproblem mit einem vermeintlichen Entscheider für L:

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Klausuraufgabe – Entscheidbarkeit

Zeigen Sie, dass die Sprache

$$L = \{(u, v) \mid w \in L(T_u) \text{ genau dann, wenn } w^R \in L(T_v)\}$$

nicht entscheidbar ist.

Löse das Halteproblem mit einem vermeintlichen Entscheider für L:

- konstruiere für $\mathcal{H}$-Instanz $\langle w, v \rangle$ TM M_{wv}, die M_w mit Eingabe v simuliert und genau dann akzeptiert, wenn M_w stoppt

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Klausuraufgabe – Entscheidbarkeit

Zeigen Sie, dass die Sprache

$$L = \{(u, v) \mid w \in L(T_u) \text{ genau dann, wenn } w^R \in L(T_v)\}$$

nicht entscheidbar ist.

Löse das Halteproblem mit einem vermeintlichen Entscheider für L:

- konstruiere für $\mathcal{H}$-Instanz $\langle w, v \rangle$ TM M_{wv}, die M_w mit Eingabe v simuliert und genau dann akzeptiert, wenn M_w stoppt
- sonst akzeptiert M_{wv} nicht

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Klausuraufgabe – Entscheidbarkeit

Zeigen Sie, dass die Sprache

$$L = \{(u, v) \mid w \in L(T_u) \text{ genau dann, wenn } w^R \in L(T_v)\}$$

nicht entscheidbar ist.

Löse das Halteproblem mit einem vermeintlichen Entscheider für L:

- konstruiere für $\mathcal{H}$-Instanz $\langle w, v \rangle$ TM M_{wv}, die M_w mit Eingabe v simuliert und genau dann akzeptiert, wenn M_w stoppt
- sonst akzeptiert M_{wv} nicht
- sei M_* eine TM, die alle Eingaben akzeptiert

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Klausuraufgabe – Entscheidbarkeit

Zeigen Sie, dass die Sprache

$$L = \{(u, v) \mid w \in L(T_u) \text{ genau dann, wenn } w^R \in L(T_v)\}$$

nicht entscheidbar ist.

Löse das Halteproblem mit einem vermeintlichen Entscheider für L:

- konstruiere für $\mathcal{H}$-Instanz $\langle w, v \rangle$ TM M_{wv}, die M_w mit Eingabe v simuliert und genau dann akzeptiert, wenn M_w stoppt
- sonst akzeptiert M_{wv} nicht
- sei M_* eine TM, die alle Eingaben akzeptiert
- dann ist $(\langle M_{wv} \rangle, \langle M_* \rangle) \in L$ genau dann, wenn $\langle w, v \rangle \in \mathcal{H}$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Klausuraufgabe – Entscheidbarkeit

Zeigen Sie, dass die Sprache

$$L = \{(u, v) \mid w \in L(T_u) \text{ genau dann, wenn } w^R \in L(T_v)\}$$

nicht entscheidbar ist.

Löse das Halteproblem mit einem vermeintlichen Entscheider für L:

- konstruiere für $\mathcal{H}$-Instanz $\langle w, v \rangle$ TM M_{wv}, die M_w mit Eingabe v simuliert und genau dann akzeptiert, wenn M_w stoppt
- sonst akzeptiert M_{wv} nicht
- sei M_* eine TM, die alle Eingaben akzeptiert
- dann ist $(\langle M_{wv} \rangle, \langle M_* \rangle) \in L$ genau dann, wenn $\langle w, v \rangle \in \mathcal{H}$

Wäre also L entscheidbar, so wäre auch $\mathcal{H}$ entscheidbar. Widerspruch.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Klausuraufgabe – Entscheidbarkeit

Zeigen Sie, dass die Sprache

$$L = \{(u, v) \mid w \in L(T_u) \text{ genau dann, wenn } w^R \in L(T_v)\}$$

nicht entscheidbar ist.

Löse das Halteproblem mit einem vermeintlichen Entscheider für L:

- konstruiere für $\mathcal{H}$-Instanz $\langle w, v \rangle$ TM M_{wv}, die M_w mit Eingabe v simuliert und genau dann akzeptiert, wenn M_w stoppt
- sonst akzeptiert M_{wv} nicht
- sei M_* eine TM, die alle Eingaben akzeptiert
- dann ist $(\langle M_{wv} \rangle, \langle M_* \rangle) \in L$ genau dann, wenn $\langle w, v \rangle \in \mathcal{H}$

Wäre also L entscheidbar, so wäre auch $\mathcal{H}$ entscheidbar. Widerspruch.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgaben zu Komplexitätsklassen

Aufgabe – P und NP

Das Entscheidungsproblem Π, ob eine gegebene Zahl eine Potenz von 2 ist, ist durch die Problembeispiele $D_\Pi := \mathbb{N}$ und die Ja-Beispiele $J_\Pi := \{2^i | i \in \mathbb{N}\}$ gegeben. Seien s_b die Kodierungsschemata, die natürliche Zahlen auf ihre b-äre Repräsentation abbilden.

Betrachten Sie nun $L[\Pi, s_1]$ und $L[\Pi, s_2]$. Beschreiben Sie für jede der beiden Sprachen kurz die Arbeitsweise einer deterministischen TM, die sie entscheidet und geben Sie ihre Laufzeit asymptotisch an. Sind die Sprachen in P? Sind sie in NP?

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_2]$ zu Problem $\Pi = (D_\Pi := \mathbb{N}, J_{Pi} := \{2^i \mid i \in \mathbb{N}\})$.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_2]$ zu Problem $\Pi = (D_\Pi := \mathbb{N}, J_{Pi} := \{2^i \mid i \in \mathbb{N}\})$.

Konvention: Es gibt keine führenden Nullen (außer die Eingabe ist 0).

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_2]$ zu Problem $\Pi = (D_\Pi := \mathbb{N}, J_{Pi} := \{2^i \mid i \in \mathbb{N}\})$.

Konvention: Es gibt keine führenden Nullen (außer die Eingabe ist 0).

Beobachtung: Die Zweierpotenzen haben in Binärdarstellung genau die Form $10 \ldots 0$.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_2]$ zu Problem $\Pi = (D_\Pi := \mathbb{N}, J_{Pi} := \{2^i \mid i \in \mathbb{N}\})$.

Konvention: Es gibt keine führenden Nullen (außer die Eingabe ist 0).

Beobachtung: Die Zweierpotenzen haben in Binärdarstellung genau die Form $10 \ldots 0$.

- Überprüfe, ob die Eingabe 0 ist (also ob an erster Stelle eine 1 steht).

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_2]$ zu Problem $\Pi = (D_\Pi := \mathbb{N}, J_{Pi} := \{2^i \mid i \in \mathbb{N}\})$.

Konvention: Es gibt keine führenden Nullen (außer die Eingabe ist 0).

Beobachtung: Die Zweierpotenzen haben in Binärdarstellung genau die Form $10 \ldots 0$.

- Überprüfe, ob die Eingabe 0 ist (also ob an erster Stelle eine 1 steht).
- Falls ja, stoppe die Berechnung und lehne die Eingabe ab.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_2]$ zu Problem $\Pi = (D_\Pi := \mathbb{N}, J_{Pi} := \{2^i \mid i \in \mathbb{N}\})$.

Konvention: Es gibt keine führenden Nullen (außer die Eingabe ist 0).

Beobachtung: Die Zweierpotenzen haben in Binärdarstellung genau die Form $10 \ldots 0$.

- Überprüfe, ob die Eingabe 0 ist (also ob an erster Stelle eine 1 steht).
- Falls ja, stoppe die Berechnung und lehne die Eingabe ab.
- Sonst gehe schrittweise nach rechts bis zum Ende der Eingabe.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_2]$ zu Problem $\Pi = (D_\Pi := \mathbb{N}, J_{Pi} := \{2^i \mid i \in \mathbb{N}\})$.

Konvention: Es gibt keine führenden Nullen (außer die Eingabe ist 0).

Beobachtung: Die Zweierpotenzen haben in Binärdarstellung genau die Form $10 \ldots 0$.

- Überprüfe, ob die Eingabe 0 ist (also ob an erster Stelle eine 1 steht).
- Falls ja, stoppe die Berechnung und lehne die Eingabe ab.
- Sonst gehe schrittweise nach rechts bis zum Ende der Eingabe.
- Überprüfe dabei, ob nach der führenden 1 noch eine 1 vorkommt.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_2]$ zu Problem $\Pi = (D_\Pi := \mathbb{N}, J_{Pi} := \{2^i \mid i \in \mathbb{N}\})$.

Konvention: Es gibt keine führenden Nullen (außer die Eingabe ist 0).

Beobachtung: Die Zweierpotenzen haben in Binärdarstellung genau die Form $10 \ldots 0$.

- Überprüfe, ob die Eingabe 0 ist (also ob an erster Stelle eine 1 steht).
- Falls ja, stoppe die Berechnung und lehne die Eingabe ab.
- Sonst gehe schrittweise nach rechts bis zum Ende der Eingabe.
- Überprüfe dabei, ob nach der führenden 1 noch eine 1 vorkommt.
- Falls ja, stoppe die Berechnung und lehne die Eingabe ab.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_2]$ zu Problem $\Pi = (D_\Pi := \mathbb{N}, J_{Pi} := \{2^i \mid i \in \mathbb{N}\})$.

Konvention: Es gibt keine führenden Nullen (außer die Eingabe ist 0).

Beobachtung: Die Zweierpotenzen haben in Binärdarstellung genau die Form $10 \ldots 0$.

- Überprüfe, ob die Eingabe 0 ist (also ob an erster Stelle eine 1 steht).
- Falls ja, stoppe die Berechnung und lehne die Eingabe ab.
- Sonst gehe schrittweise nach rechts bis zum Ende der Eingabe.
- Überprüfe dabei, ob nach der führenden 1 noch eine 1 vorkommt.
- Falls ja, stoppe die Berechnung und lehne die Eingabe ab.
- Sonst akzeptiere die Eingabe.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_2]$ zu Problem $\Pi = (D_\Pi := \mathbb{N}, J_{Pi} := \{2^i \mid i \in \mathbb{N}\})$.

Konvention: Es gibt keine führenden Nullen (außer die Eingabe ist 0).

Beobachtung: Die Zweierpotenzen haben in Binärdarstellung genau die Form $10 \ldots 0$.

- Überprüfe, ob die Eingabe 0 ist (also ob an erster Stelle eine 1 steht).
- Falls ja, stoppe die Berechnung und lehne die Eingabe ab.
- Sonst gehe schrittweise nach rechts bis zum Ende der Eingabe.
- Überprüfe dabei, ob nach der führenden 1 noch eine 1 vorkommt.
- Falls ja, stoppe die Berechnung und lehne die Eingabe ab.
- Sonst akzeptiere die Eingabe.

$\Rightarrow$ Zeitkomplexität der TM ist linear in der Eingabegröße.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_2]$ zu Problem $\Pi = (D_\Pi := \mathbb{N}, J_{Pi} := \{2^i \mid i \in \mathbb{N}\})$.

Konvention: Es gibt keine führenden Nullen (außer die Eingabe ist 0).

Beobachtung: Die Zweierpotenzen haben in Binärdarstellung genau die Form $10 \ldots 0$.

- Überprüfe, ob die Eingabe 0 ist (also ob an erster Stelle eine 1 steht).
- Falls ja, stoppe die Berechnung und lehne die Eingabe ab.
- Sonst gehe schrittweise nach rechts bis zum Ende der Eingabe.
- Überprüfe dabei, ob nach der führenden 1 noch eine 1 vorkommt.
- Falls ja, stoppe die Berechnung und lehne die Eingabe ab.
- Sonst akzeptiere die Eingabe.

$\Rightarrow$ Zeitkomplexität der TM ist linear in der Eingabegröße.

$\Rightarrow$ $L[\Pi, s_2]$ liegt in P.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_1]$ zu dem Problem $\Pi = (D_\Pi := \mathbb{N}, J_\Pi := \{2^i \mid \in \mathbb{N}\})$.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P **und** NP

Wir betrachten $L[\Pi, s_1]$ zu dem Problem $\Pi = (D_\Pi := \mathbb{N}, J_\Pi := \{2^i \mid \in \mathbb{N}\})$.

Eine TM die $L[\Pi, s_1]$ entscheidet kann wie folgt konstruiert werden:

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_1]$ zu dem Problem $\Pi = (D_\Pi := \mathbb{N}, J_\Pi := \{2^i \mid \in \mathbb{N}\})$.

Eine TM die $L[\Pi, s_1]$ entscheidet kann wie folgt konstruiert werden:

- Durchlaufe immer wieder die Eingabe.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_1]$ zu dem Problem $\Pi = (D_\Pi := \mathbb{N}, J_\Pi := \{2^i \mid \in \mathbb{N}\})$.

Eine TM die $L[\Pi, s_1]$ entscheidet kann wie folgt konstruiert werden:

- Durchlaufe immer wieder die Eingabe.
- Bei jedem Durchlauf, merke ob gerade oder ungerade viele 1-en auf dem Band stehen.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_1]$ zu dem Problem $\Pi = (D_\Pi := \mathbb{N}, J_\Pi := \{2^i \mid \in \mathbb{N}\})$.

Eine TM die $L[\Pi, s_1]$ entscheidet kann wie folgt konstruiert werden:

- Durchlaufe immer wieder die Eingabe.
- Bei jedem Durchlauf, merke ob gerade oder ungerade viele 1-en auf dem Band stehen.
- Bei jedem Durchlauf ersetze jede zweite 1 durch eine 0.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_1]$ zu dem Problem $\Pi = (D_\Pi := \mathbb{N}, J_\Pi := \{2^i \mid \in \mathbb{N}\})$.

Eine TM die $L[\Pi, s_1]$ entscheidet kann wie folgt konstruiert werden:

- Durchlaufe immer wieder die Eingabe.
- Bei jedem Durchlauf, merke ob gerade oder ungerade viele 1-en auf dem Band stehen.
- Bei jedem Durchlauf ersetze jede zweite 1 durch eine 0.
- Wenn bei einem Durchlauf ungerade viele 1-en erkannt wurden, stoppe und lehne die Eingabe ab.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_1]$ zu dem Problem $\Pi = (D_\Pi := \mathbb{N}, J_\Pi := \{2^i \mid \in \mathbb{N}\})$.

Eine TM die $L[\Pi, s_1]$ entscheidet kann wie folgt konstruiert werden:

- Durchlaufe immer wieder die Eingabe.
- Bei jedem Durchlauf, merke ob gerade oder ungerade viele 1-en auf dem Band stehen.
- Bei jedem Durchlauf ersetze jede zweite 1 durch eine 0.
- Wenn bei einem Durchlauf ungerade viele 1-en erkannt wurden, stoppe und lehne die Eingabe ab.
- Ansonsten, falls am Ende eine 1 stehen bleibt, akzeptiere die Eingabe.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_1]$ zu dem Problem $\Pi = (D_\Pi := \mathbb{N}, J_\Pi := \{2^i \mid \in \mathbb{N}\})$.

Eine TM die $L[\Pi, s_1]$ entscheidet kann wie folgt konstruiert werden:

- Durchlaufe immer wieder die Eingabe.
- Bei jedem Durchlauf, merke ob gerade oder ungerade viele 1-en auf dem Band stehen.
- Bei jedem Durchlauf ersetze jede zweite 1 durch eine 0.
- Wenn bei einem Durchlauf ungerade viele 1-en erkannt wurden, stoppe und lehne die Eingabe ab.
- Ansonsten, falls am Ende eine 1 stehen bleibt, akzeptiere die Eingabe.

$\Rightarrow$ Zeitkomplexität der TM ist *quadratisch* in der Eingabegröße.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – P und NP

Wir betrachten $L[\Pi, s_1]$ zu dem Problem $\Pi = (D_\Pi := \mathbb{N}, J_\Pi := \{2^i \mid \in \mathbb{N}\})$.

Eine TM die $L[\Pi, s_1]$ entscheidet kann wie folgt konstruiert werden:

- Durchlaufe immer wieder die Eingabe.
- Bei jedem Durchlauf, merke ob gerade oder ungerade viele 1-en auf dem Band stehen.
- Bei jedem Durchlauf ersetze jede zweite 1 durch eine 0.
- Wenn bei einem Durchlauf ungerade viele 1-en erkannt wurden, stoppe und lehne die Eingabe ab.
- Ansonsten, falls am Ende eine 1 stehen bleibt, akzeptiere die Eingabe.

$\Rightarrow$ Zeitkomplexität der TM ist *quadratisch* in der Eingabegröße.

$\Rightarrow$ $L[\Pi, s_1]$ liegt in P

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

a) Definieren Sie PSPACE und EXPTIME, und beschreiben Sie diese Klassen mit eigenen Worten.

b) Betrachten Sie die Klassen L, NLOG, P, NP, PSPACE, NPSPACE, EXPSPACE und EXP näher:

- Stellen Sie die Beziehung zwischen diesen Klassen mithilfe eines geeigneten Diagrammtyps dar.
- Geben Sie Probleme an, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?
- Welche Folgen ergeben sich, falls P = NP? (Hinweis: Was Bedeutet das in der Praxis, Kryptographie,...)

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

a) Definieren Sie PSPACE und EXP, und beschreiben Sie diese Klassen mit eigenen Worten.

PSPACE

Klasse von Entscheidungsproblemen, die von einer DTM in polynomial viel Platz gelöst werden.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

a) Definieren Sie PSPACE und EXP, und beschreiben Sie diese Klassen mit eigenen Worten.

PSPACE

Klasse von Entscheidungsproblemen, die von einer TM in polynomial viel Platz gelöst werden.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

a) Definieren Sie PSPACE und EXP, und beschreiben Sie diese Klassen mit eigenen Worten.

PSPACE

Klasse von Entscheidungsproblemen, die von einer TM in polynomial viel Platz gelöst werden.

EXP

Klasse von Entscheidungsproblemen, die von einer DTM in $\mathcal{O}(2^{p(n)})$ Zeit gelöst werden können.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

Beziehung zwischen den Klassen: L, NL, P, NP, PSPACE, NPSPACE, EXPSPACE, EXP.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

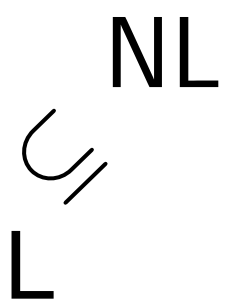
Beziehung zwischen den Klassen: L, NL, P, NP, PSPACE, NPSPACE, EXPSPACE, EXP.

L

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

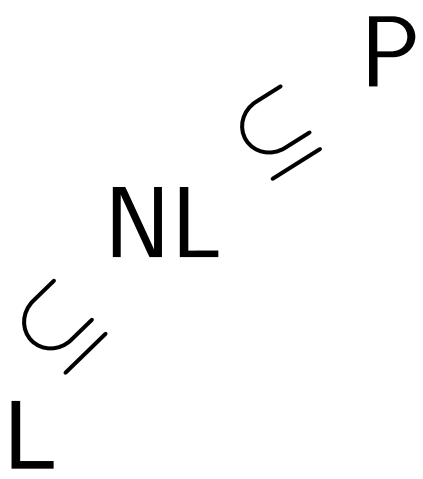
Beziehung zwischen den Klassen: L, NL, P, NP, PSPACE, NPSPACE, EXPSPACE, EXP.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

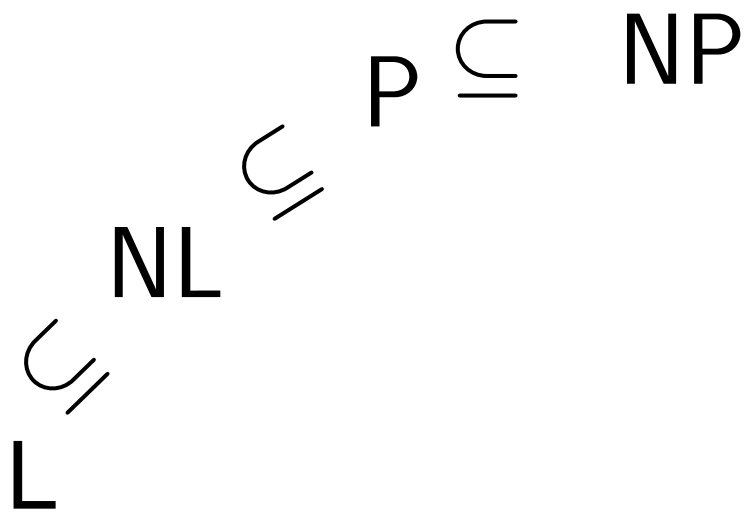
Beziehung zwischen den Klassen: L, NL, P, NP, PSPACE, NPSPACE, EXPSPACE, EXP.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

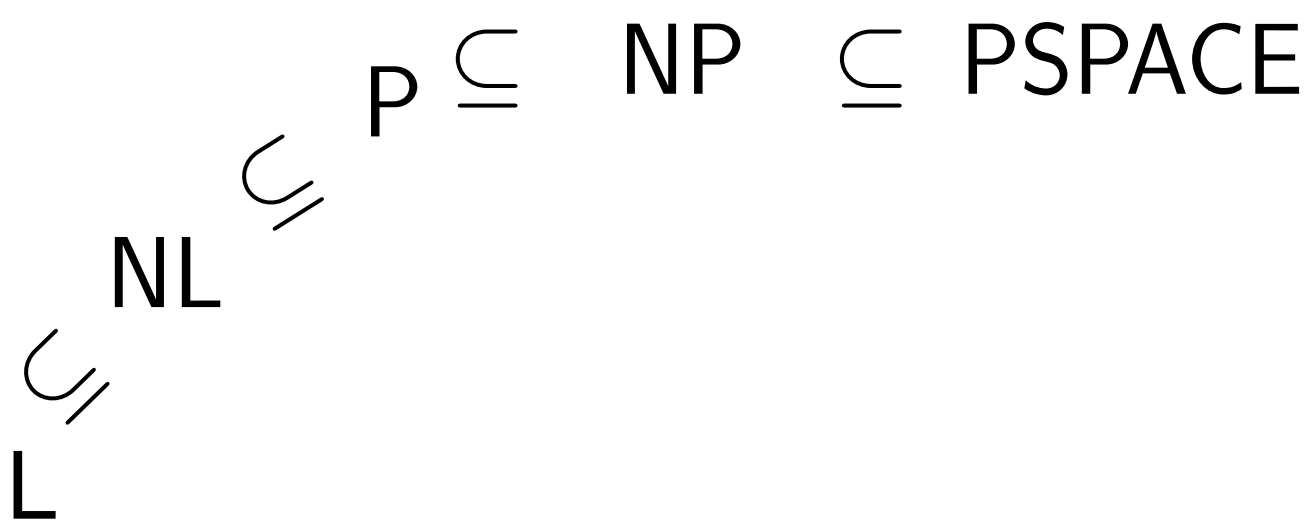
Beziehung zwischen den Klassen: L, NL, P, NP, PSPACE, NPSPACE, EXPSPACE, EXP.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

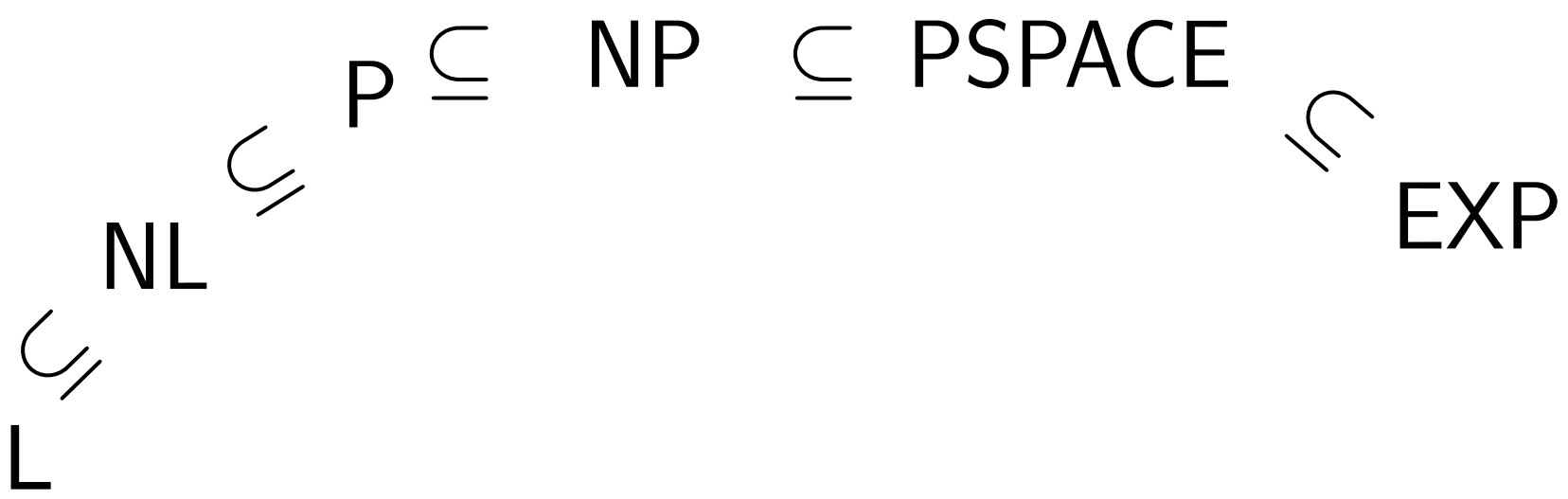
Beziehung zwischen den Klassen: L, NL, P, NP, PSPACE, NPSPACE, EXPSPACE, EXP.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

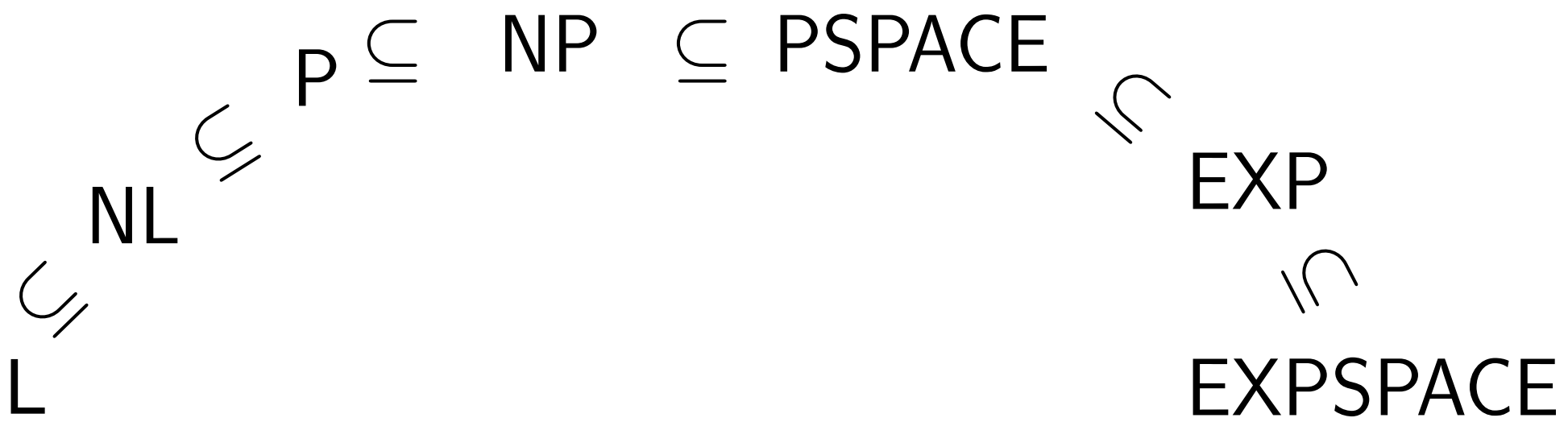
Beziehung zwischen den Klassen: L, NL, P, NP, PSPACE, NPSPACE, EXPSPACE, EXP.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

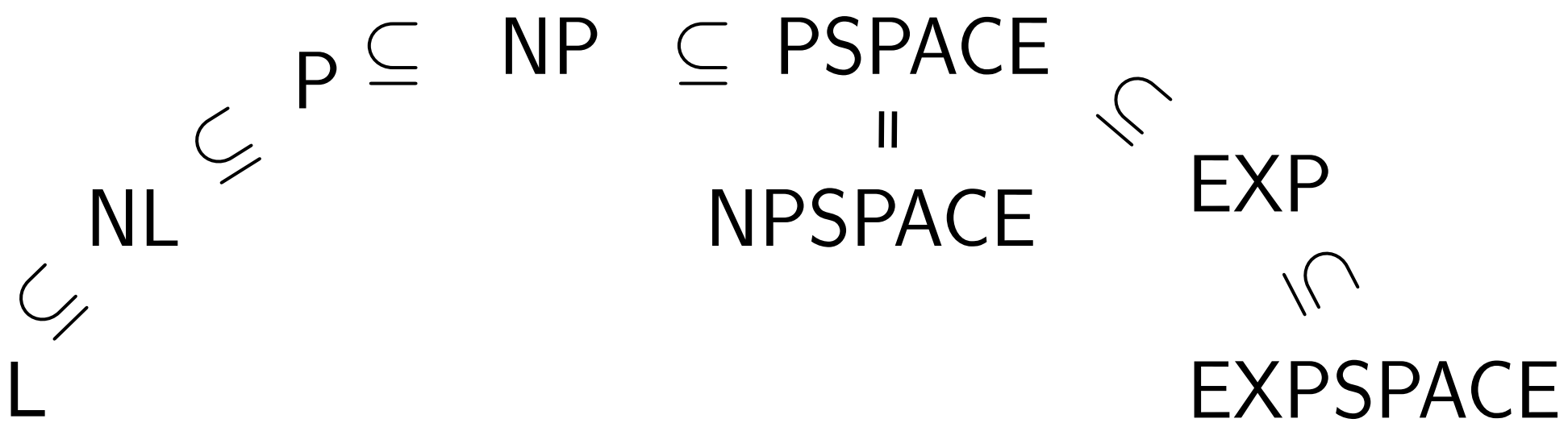
Beziehung zwischen den Klassen: L, NL, P, NP, PSPACE, NPSPACE, EXPSPACE, EXP.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

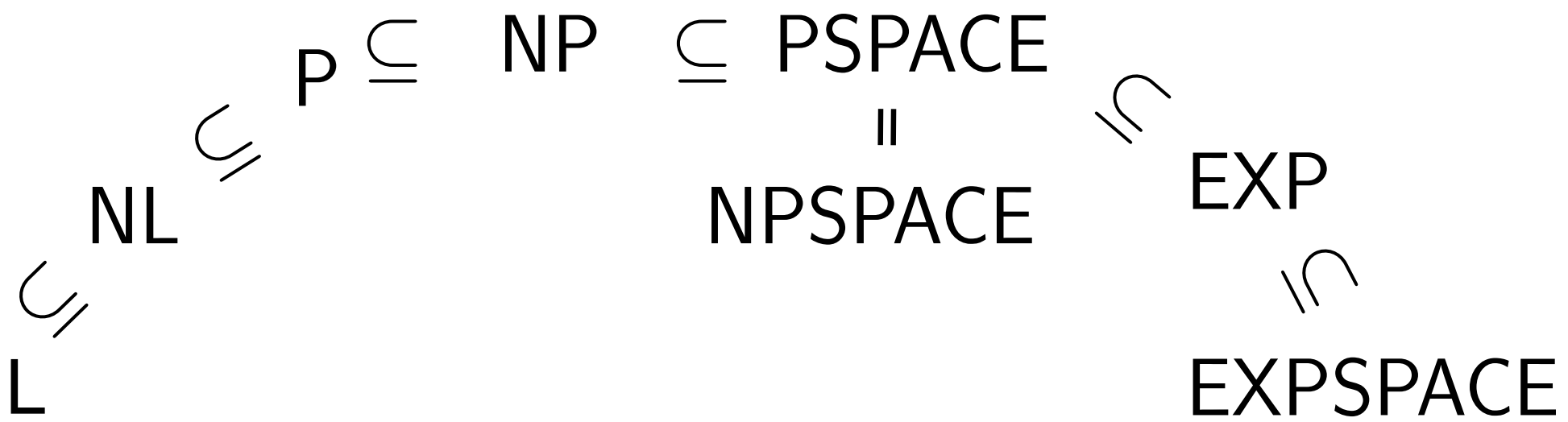
Beziehung zwischen den Klassen: L, NL, P, NP, PSPACE, NPSPACE, EXPSPACE, EXP.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

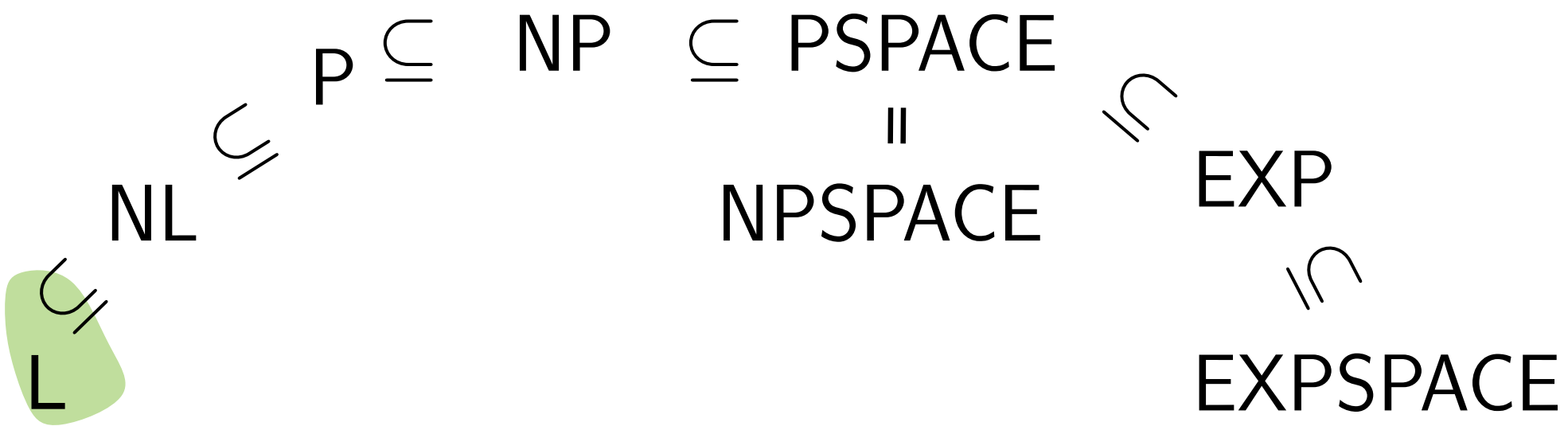
Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

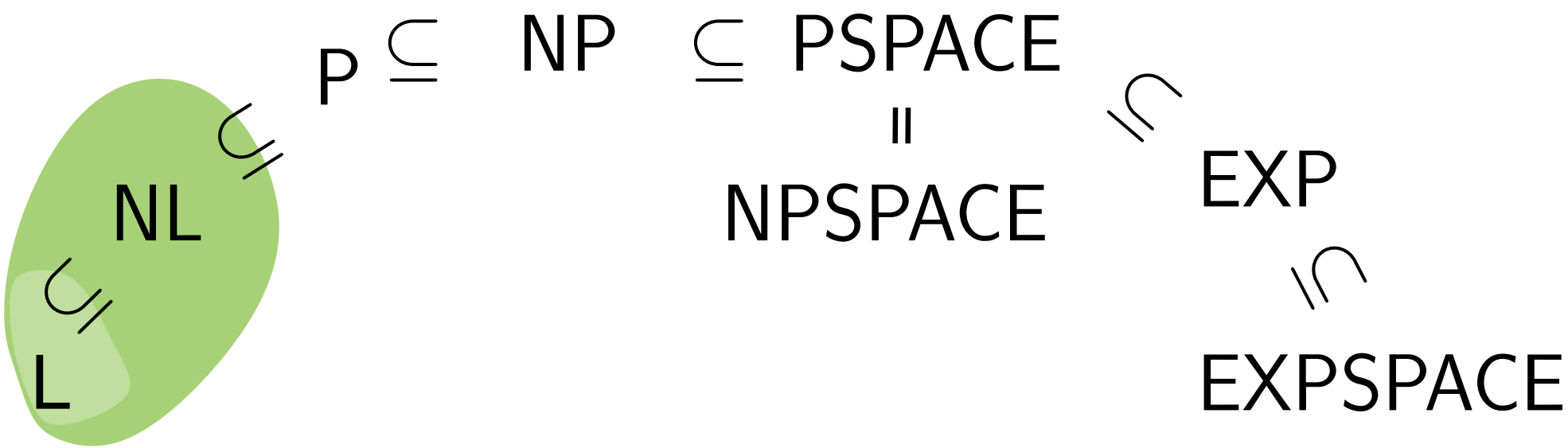
Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

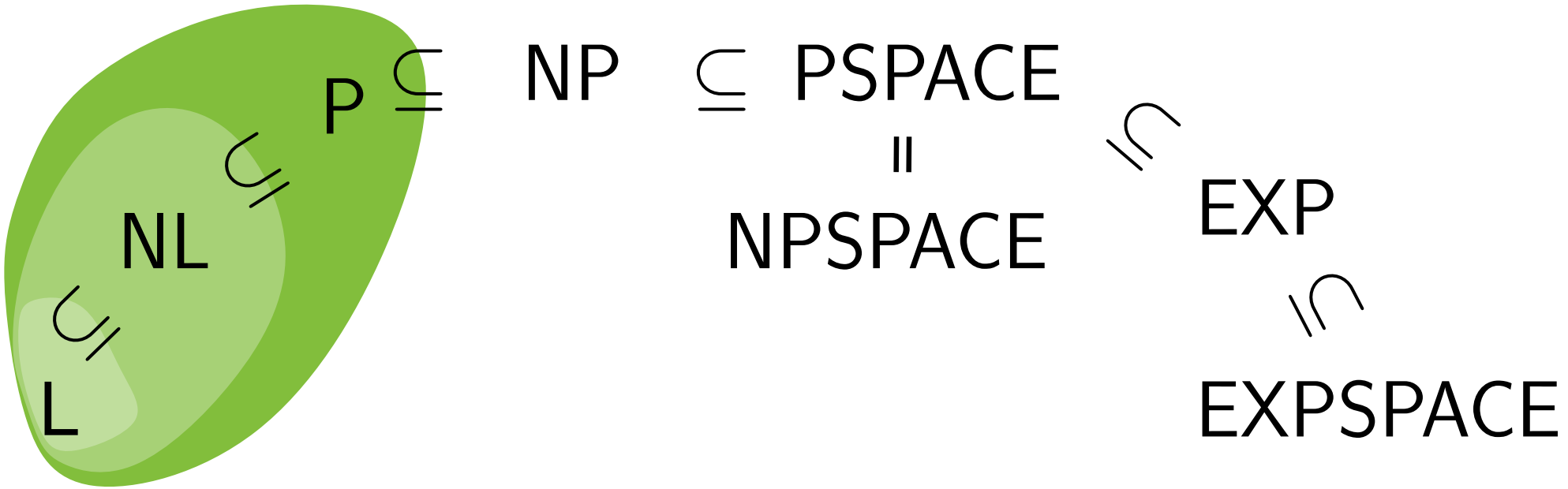
Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

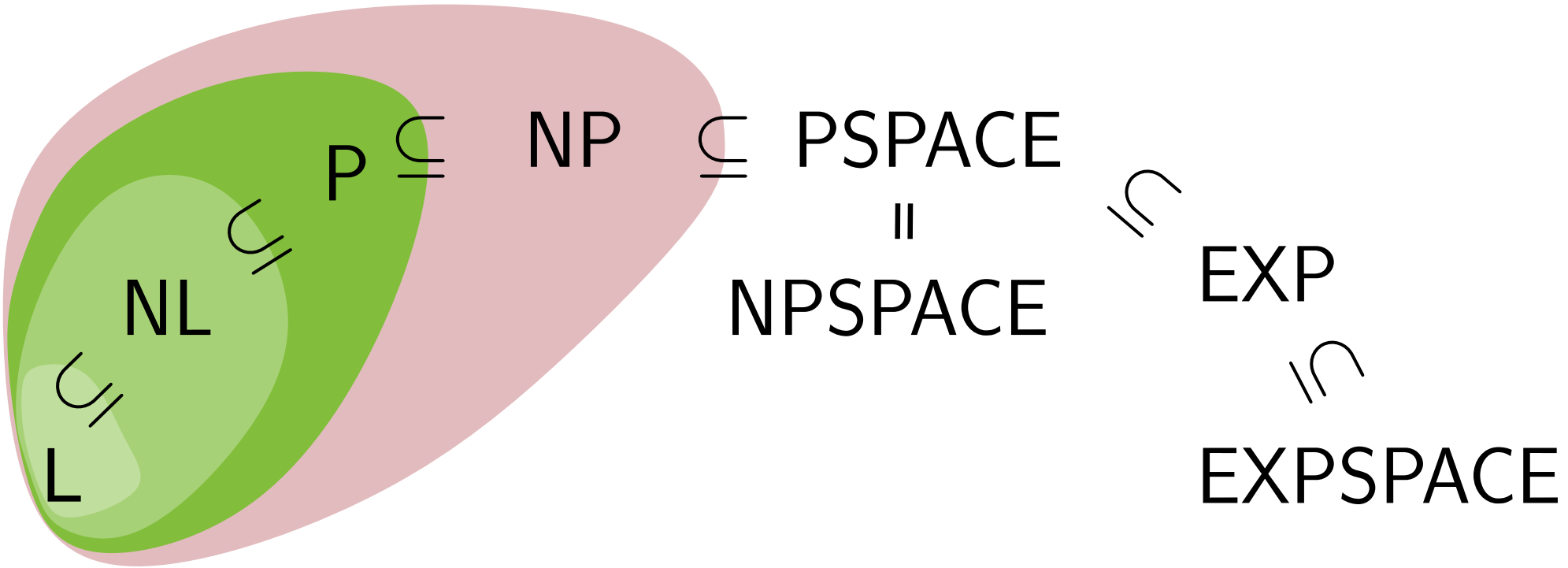
Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

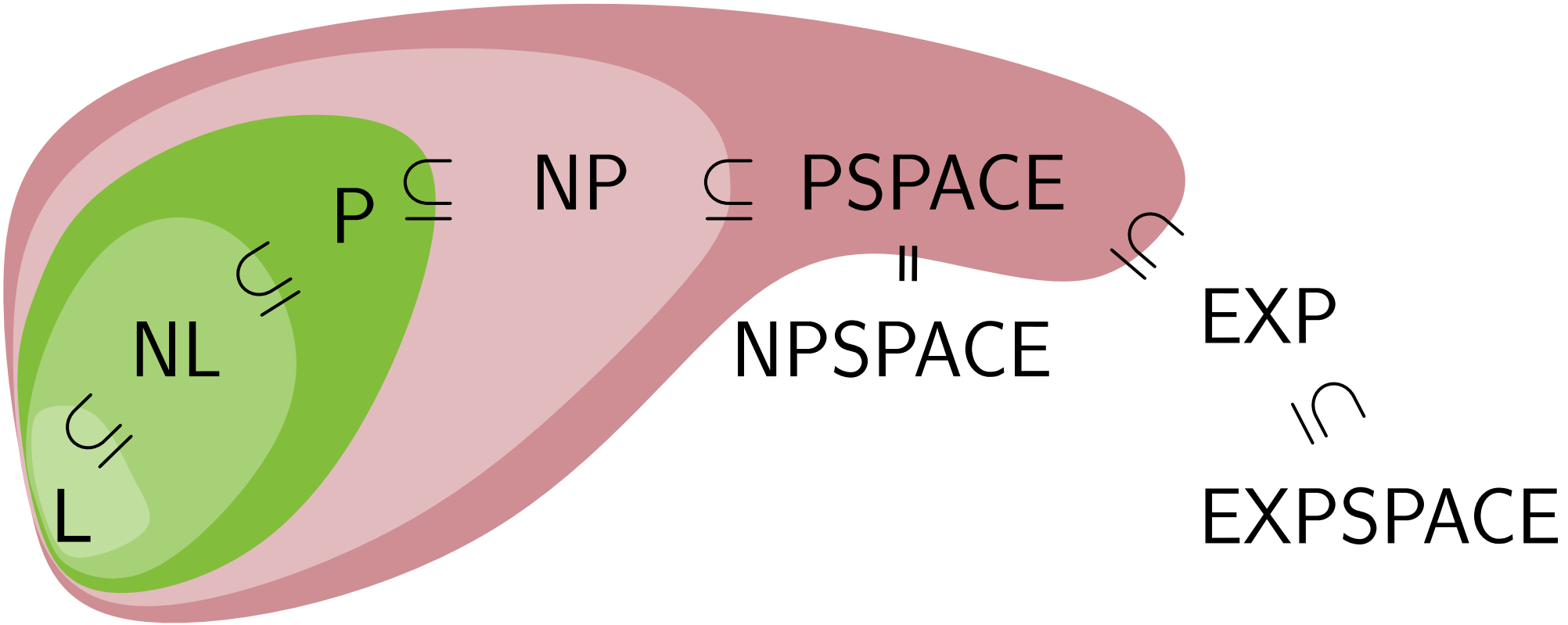
Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

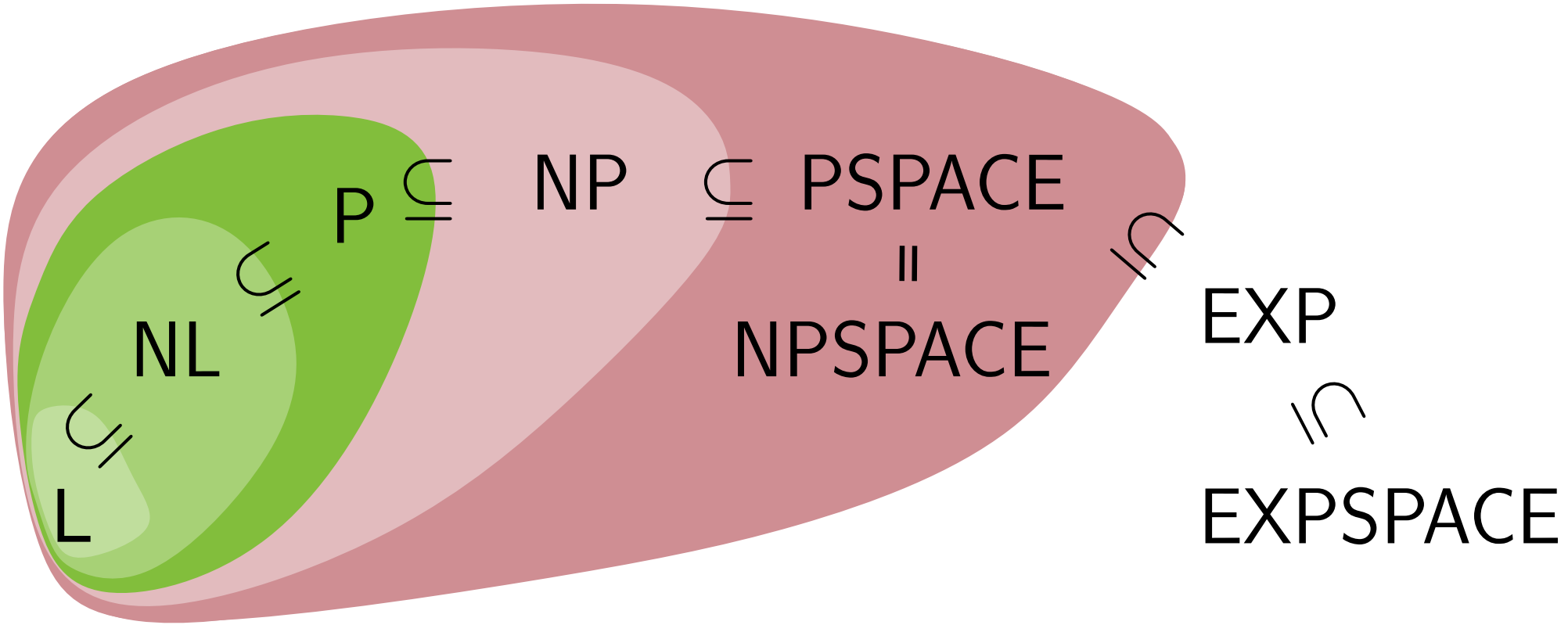
Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

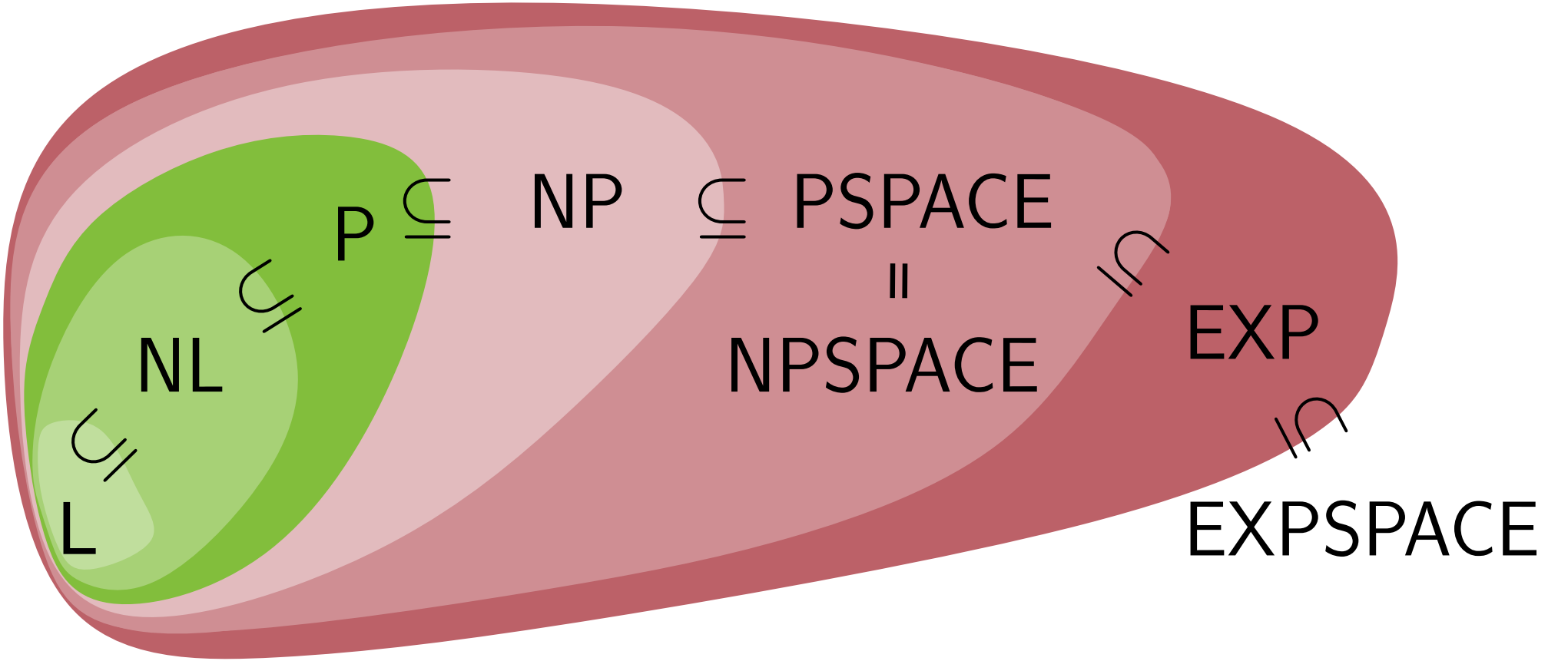
Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

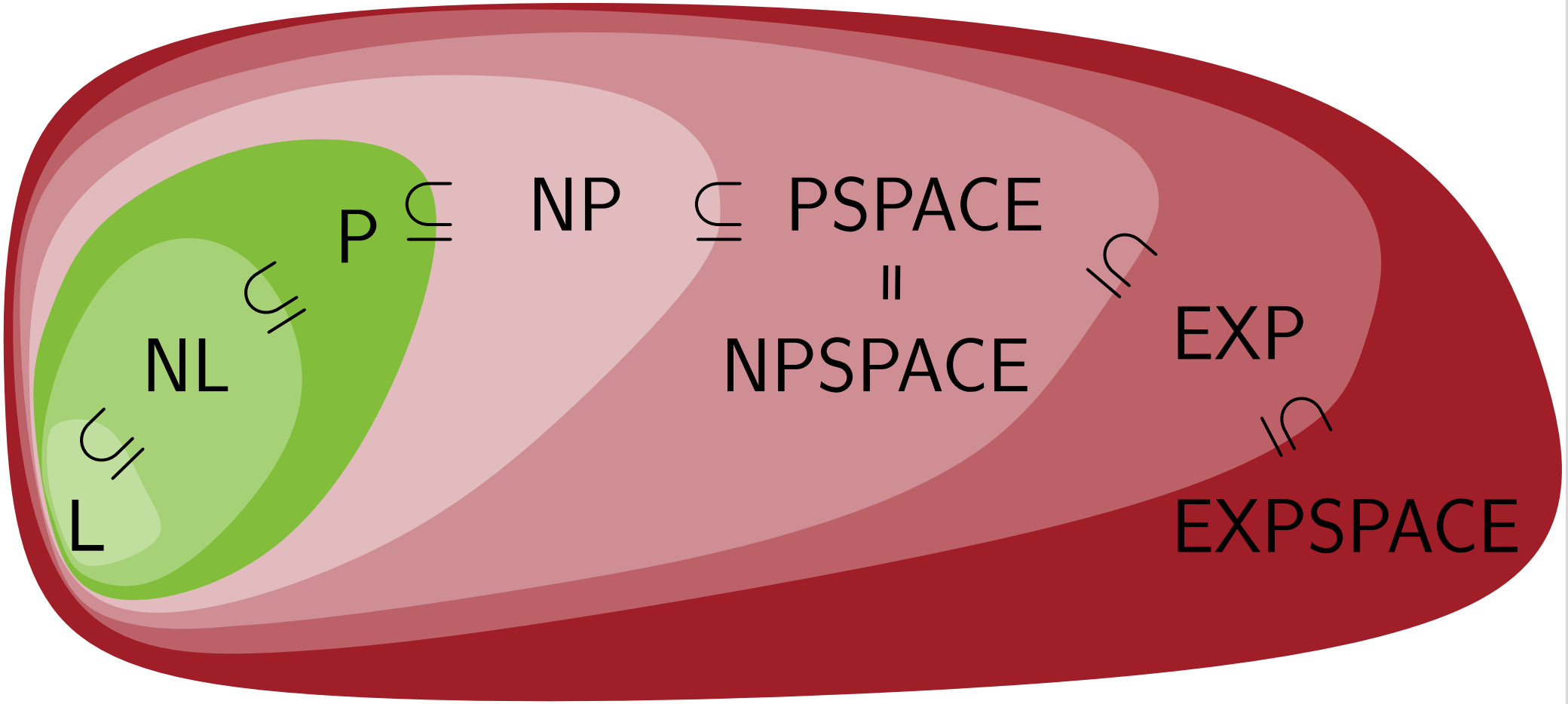
Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

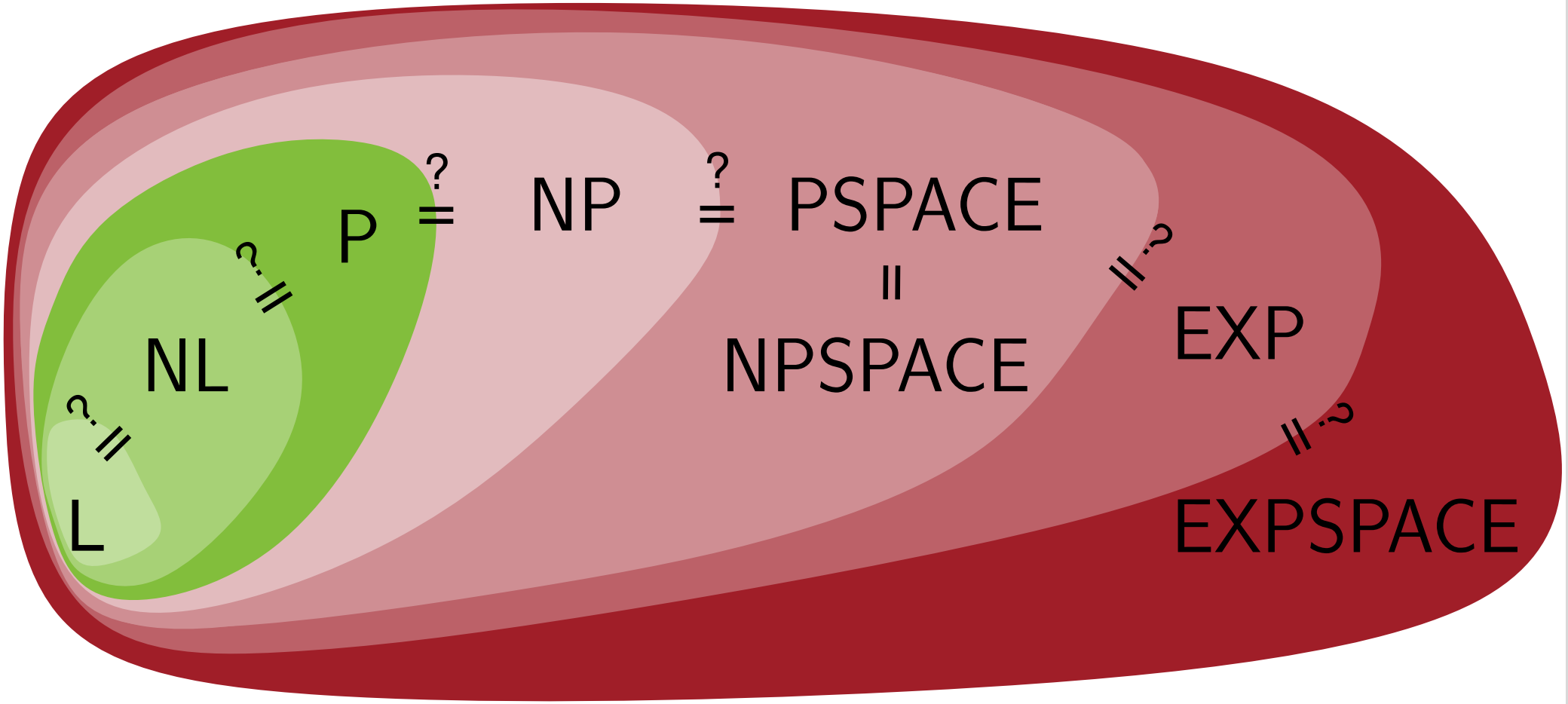
Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

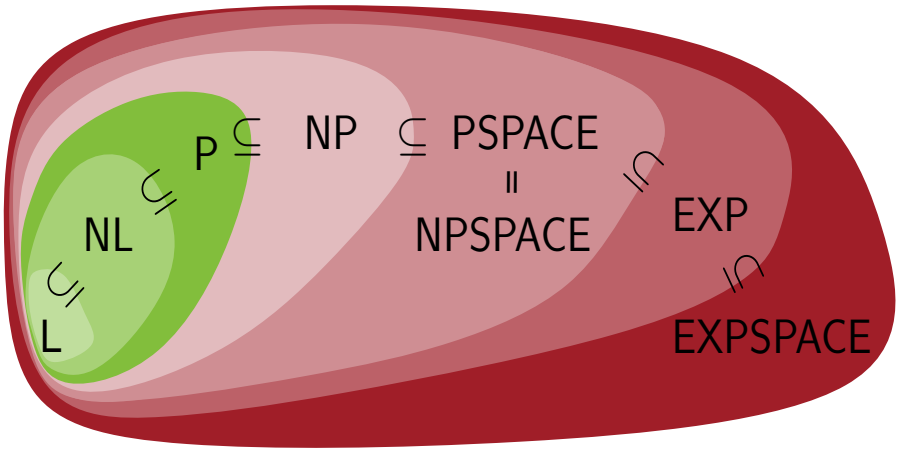
Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?

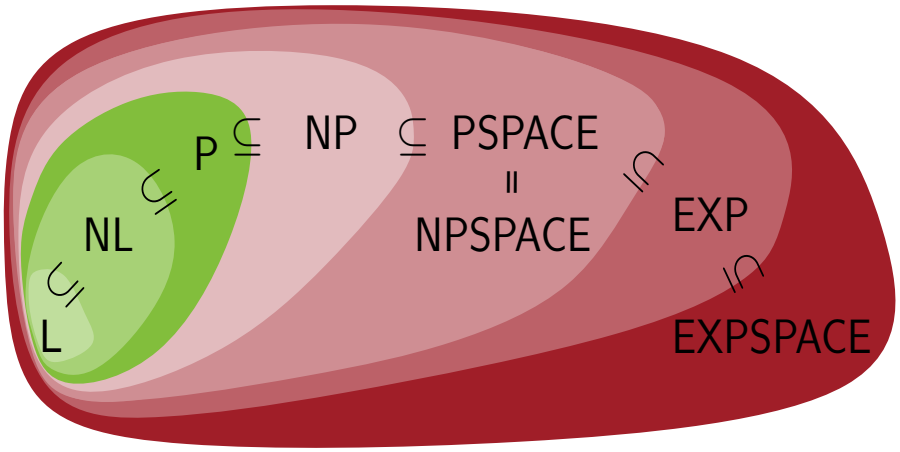


Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?

Problem des Handlungsreisenden



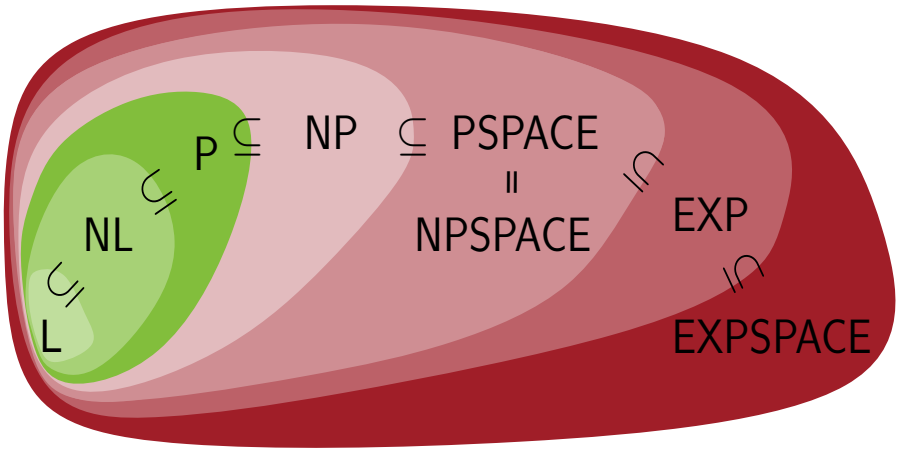
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?

Problem des Handlungsreisenden

- Mögliches exaktes Lösungsverfahren: Alle Weglängen aller möglichen Rundreisen berechnen



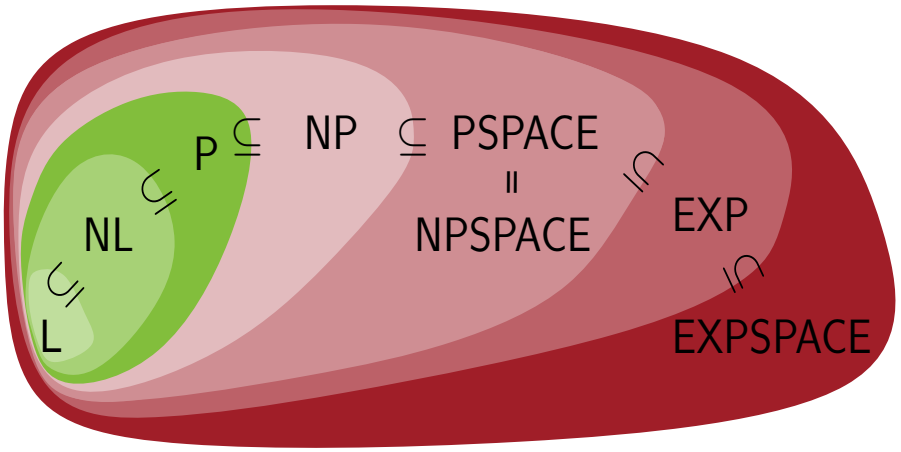
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?

Problem des Handlungsreisenden

- Mögliches exaktes Lösungsverfahren: Alle Weglängen aller möglichen Rundreisen berechnen
- Schon bei kleiner Anzahl von Städten unpraktikabel



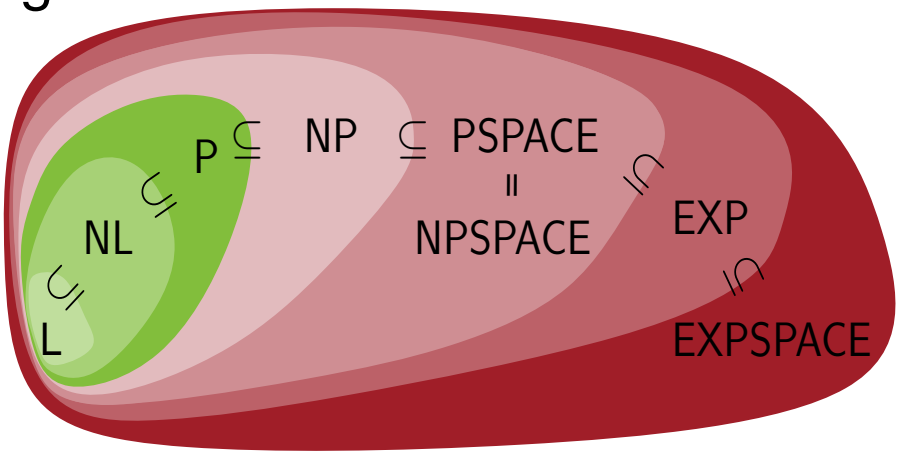
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?

Problem des Handlungsreisenden

- Mögliches exaktes Lösungsverfahren: Alle Weglängen aller möglichen Rundreisen berechnen
- Schon bei kleiner Anzahl von Städten unpraktikabel
- Bei n Städten $\rightarrow \frac{(n-1)!}{2}$ verschieden mögliche Rundreisen



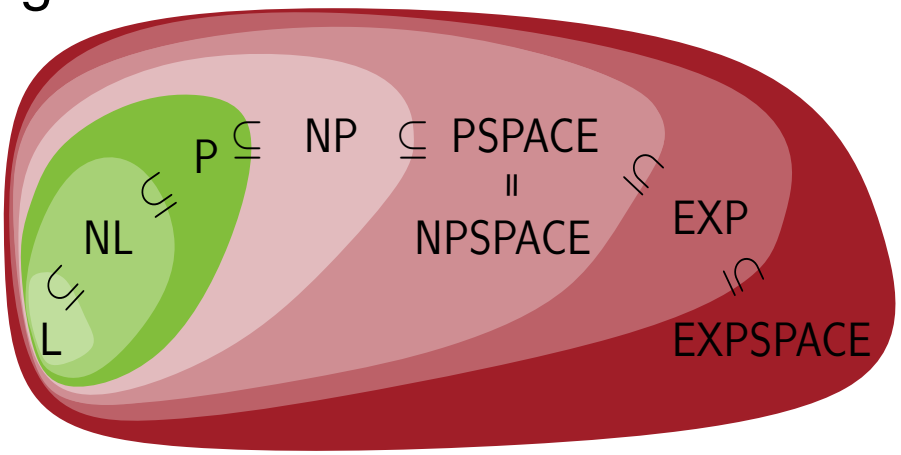
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

Probleme, die für heutige Computer bzgl. sinnvoll verwendeter Rechenzeit und Speicherplatz nicht lösbar sind. Zu welchen Komplexitätsklassen gehören diese Probleme?

Problem des Handlungsreisenden

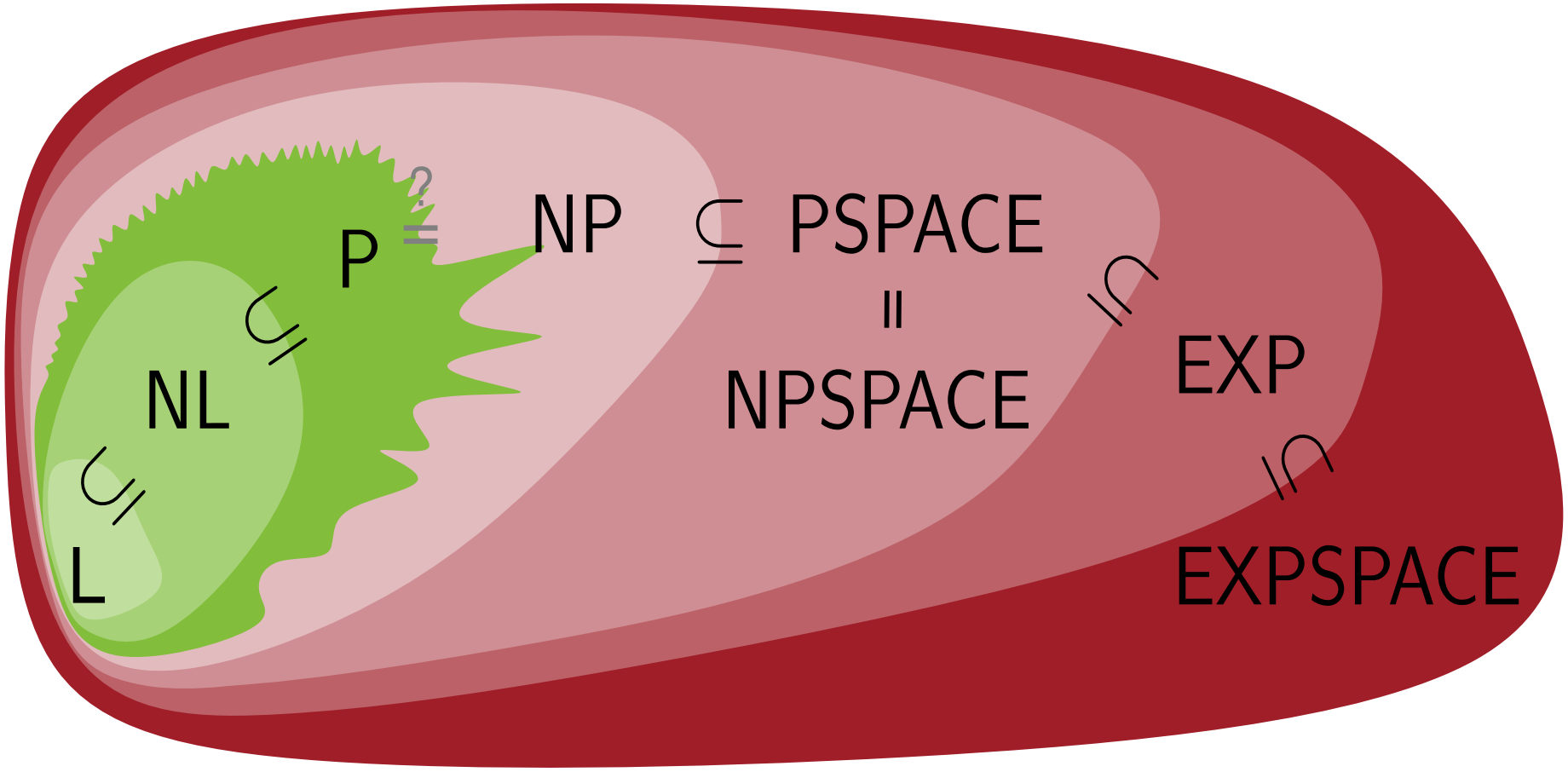
- Mögliches exaktes Lösungsverfahren: Alle Weglängen aller möglichen Rundreisen berechnen
- Schon bei kleiner Anzahl von Städten unpraktikabel
- Bei n Städten $\rightarrow \frac{(n-1)!}{2}$ verschieden mögliche Rundreisen
- Für 16 Städte $\rightarrow$ 653 Mrd. Rundreisen



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$

- Eines der Millenium-Preis-Probleme

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$

- Eines der Millenium-Preis-Probleme
- Können Probleme in NP genauso effizient gelöst werden, wie in P?

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$

- Eines der Millenium-Preis-Probleme
- Können Probleme in NP genauso effizient gelöst werden, wie in P?
- Wenn der Beweis nicht konstruktiv ist, weiß man die Auswirkung nicht

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$

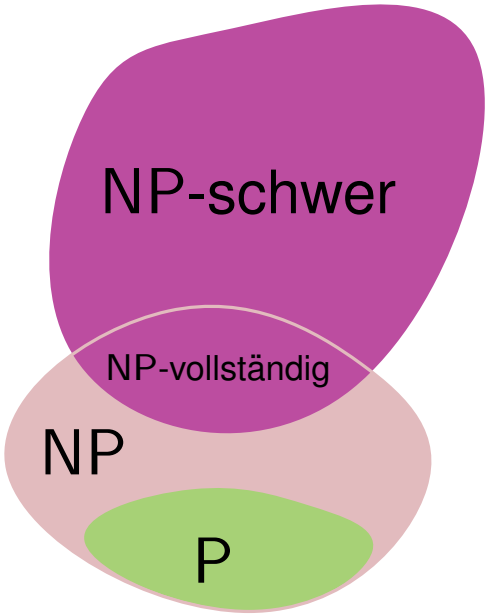
- Eines der Millenium-Preis-Probleme
- Können Probleme in NP genauso effizient gelöst werden, wie in P?
- Wenn der Beweis nicht konstruktiv ist, weiß man die Auswirkung nicht
- Viele Probleme sind NP-vollständig

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$

- Eines der Millenium-Preis-Probleme
- Können Probleme in NP genauso effizient gelöst werden, wie in P?
- Wenn der Beweis nicht konstruktiv ist, weiß man die Auswirkung nicht
- Viele Probleme sind NP-vollständig

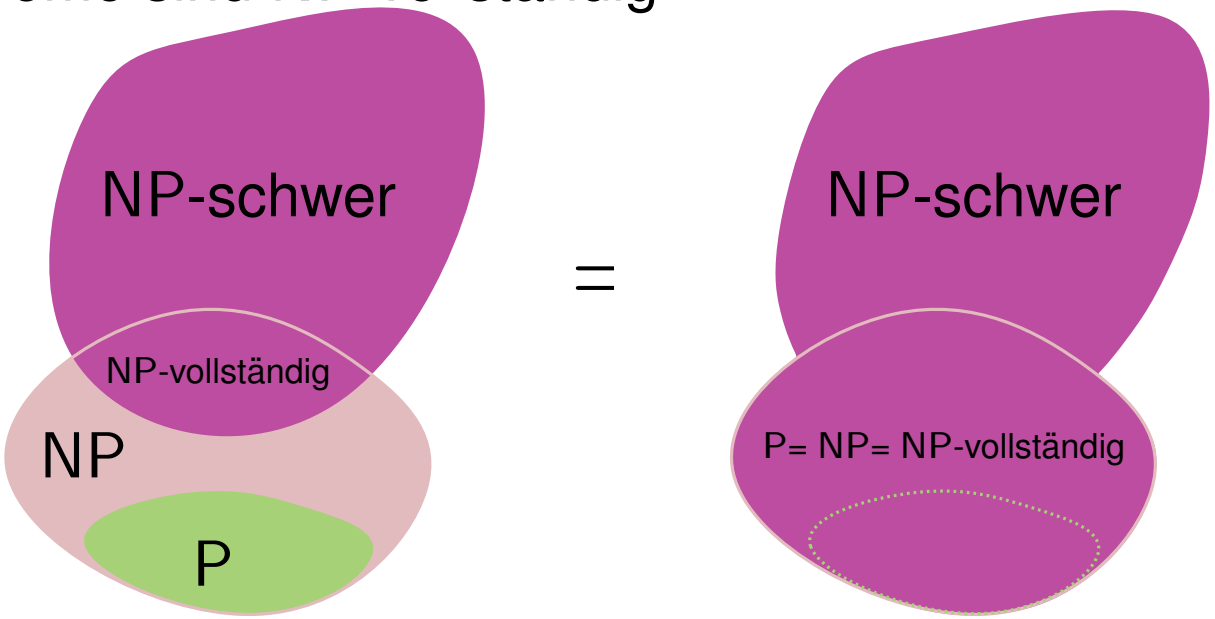


Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$

- Eines der Millenium-Preis-Probleme
- Können Probleme in NP genauso effizient gelöst werden, wie in P?
- Wenn der Beweis nicht konstruktiv ist, weiß man die Auswirkung nicht
- Viele Probleme sind NP-vollständig



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$

Beispiel Kryptographie:

- aus P = NP würde folgen, dass einige kryptographische Primitive nicht existieren, z.B. Einwegfunktionen

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$

Beispiel Kryptographie:

- aus P = NP würde folgen, dass einige kryptographische Primitive nicht existieren, z.B. Einwegfunktionen

Aber Achtung:

- es gilt nicht: „Kryptographie ist möglich $\iff$ P $\neq$ NP“

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$

Beispiel Kryptographie:

- aus P = NP würde folgen, dass einige kryptographische Primitive nicht existieren, z.B. Einwegfunktionen

Aber Achtung:

- es gilt nicht: „Kryptographie ist möglich $\iff$ $P \neq NP$“
- bei P und NP geht es um asymptotisches Laufzeitverhalten

  → aber z.B. bei AES: Schlüsselgröße 256 bits, Blockgröße 128, also feste Instanzgröße

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$

Beispiel Kryptographie:

- aus P = NP würde folgen, dass einige kryptographische Primitive nicht existieren, z.B. Einwegfunktionen

Aber Achtung:

- es gilt nicht: „Kryptographie ist möglich $\iff$ $P \neq NP$“
- bei P und NP geht es um asymptotisches Laufzeitverhalten
- worst-case-Laufzeit vs. average-case-Laufzeit

  → damit ein Problem NP-vollständig ist, reicht es, wenn manche Instanzen schwierig zu lösen sind. Es sollten aber alle verschlüsselten Nachrichten schwierig zu entschlüsseln sein!

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$

Beispiel Kryptographie:

- aus P = NP würde folgen, dass einige kryptographische Primitive nicht existieren, z.B. Einwegfunktionen

Aber Achtung:

- es gilt nicht: „Kryptographie ist möglich $\iff$ P $\neq$ NP“
- bei P und NP geht es um asymptotisches Laufzeitverhalten
- worst-case-Laufzeit vs. average-case-Laufzeit
- viele als schwierig angenommene krytpographische Probleme sind nicht als NP-vollständig bekannt, z.B. Ganzzahlfaktorisierung

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – Komplexitätsklassen

$$P \stackrel{?}{=} NP$$

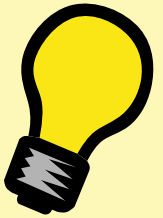
Beispiel Kryptographie:

- aus P = NP würde folgen, dass einige kryptographische Primitive nicht existieren, z.B. Einwegfunktionen

Aber Achtung:

- komplizierte und nuancierte Situation
- Fragestellungen weit über $P \stackrel{?}{=} NP$ hinaus
- bei Interesse:

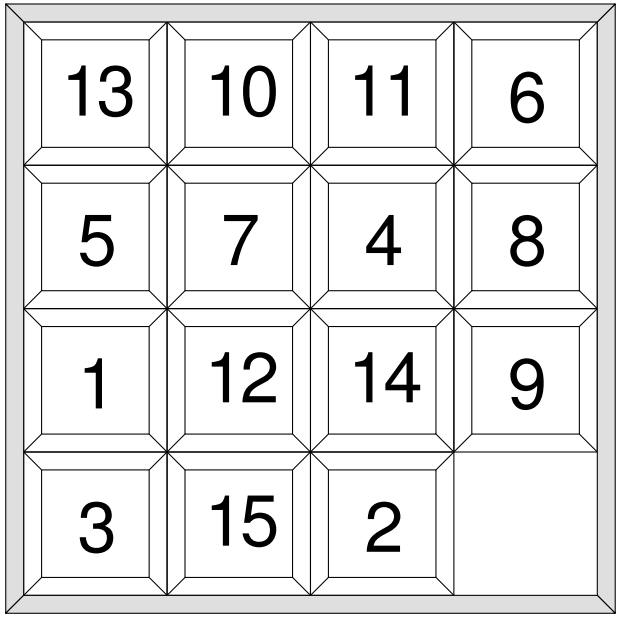
  → Vorlesung „Komplexitätstheorie, mit Anwendungen in der Kryptographie“

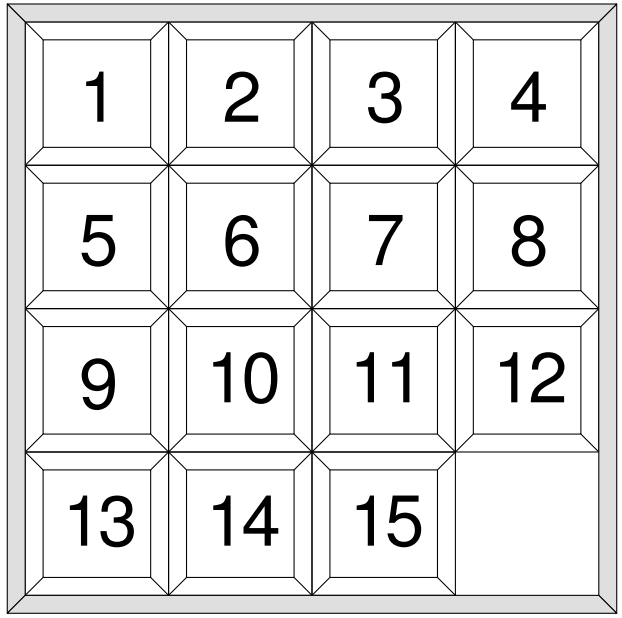


Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – 15 Puzzle

15-Puzzle



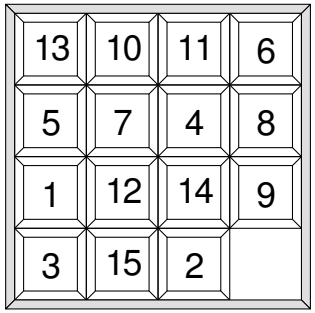


- Formulieren Sie das Problem 15-Puzzle als Optimierungs-, Optimalwert- und Entscheidungsproblem
- Geben Sie ein Kodierungsschema an und bestimmen Sie die Kodierungslänge der Instanzen.

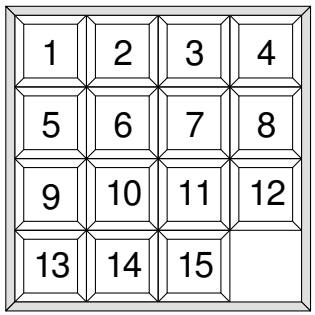
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – 15 Puzzle

Optimierungsproblem:



Optimalwertproblem:



Entscheidungsproblem:

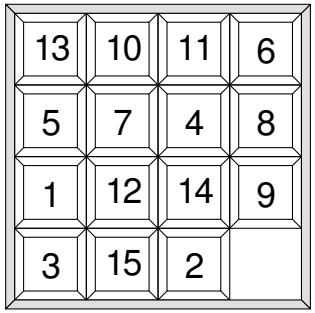
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – 15 Puzzle

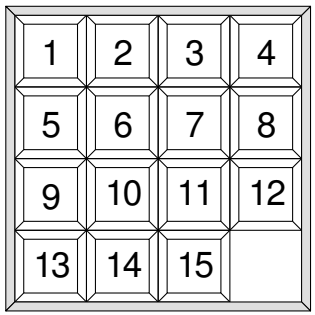
Optimierungsproblem:

Gegeben: Anfangskonfiguration von 15-Puzzle

Gesucht: Folge von Zügen mit minimaler Länge, die die Anfangskonfiguration in die Zielkonfiguration überführt.



Optimalwertproblem:



Entscheidungsproblem:

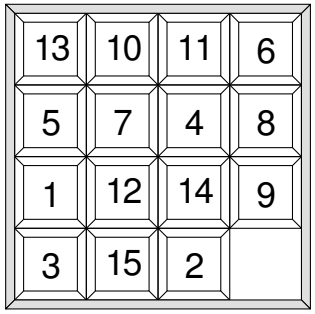
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – 15 Puzzle

Optimierungsproblem:

Gegeben: Anfangskonfiguration von 15-Puzzle

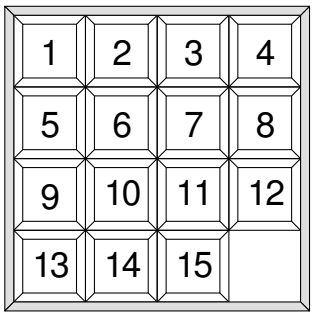
Gesucht: Folge von Zügen mit minimaler Länge, die die Anfangskonfiguration in die Zielkonfiguration überführt.



Optimalwertproblem:

Gegeben: Anfangskonfiguration von 15-Puzzle

Gesucht: Die Länge einer kürzesten Folge von Zügen, die die Anfangskonfiguration in die Zielkonfiguration überführt.



Entscheidungsproblem:

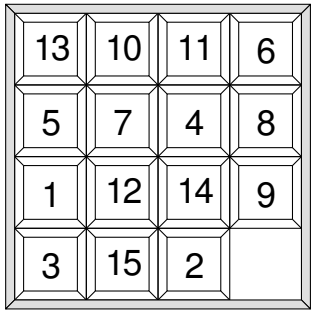
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – 15 Puzzle

Optimierungsproblem:

Gegeben: Anfangskonfiguration von 15-Puzzle

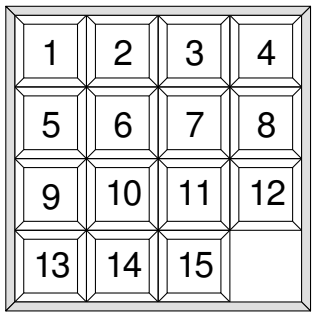
Gesucht: Folge von Zügen mit minimaler Länge, die die Anfangskonfiguration in die Zielkonfiguration überführt.



Optimalwertproblem:

Gegeben: Anfangskonfiguration von 15-Puzzle

Gesucht: Die Länge einer kürzesten Folge von Zügen, die die Anfangskonfiguration in die Zielkonfiguration überführt.



Entscheidungsproblem:

Gegeben: Anfangskonfiguration von 15-Puzzle und ein Parameter k

Gesucht: Gibt es eine Folge von Zügen der Länge $\leq k$, die die Anfangskonfiguration in die Zielkonfiguration überführt.

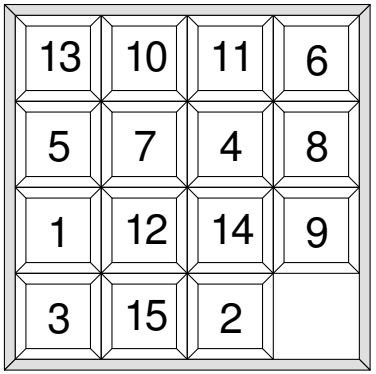
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Aufgabe – 15 Puzzle

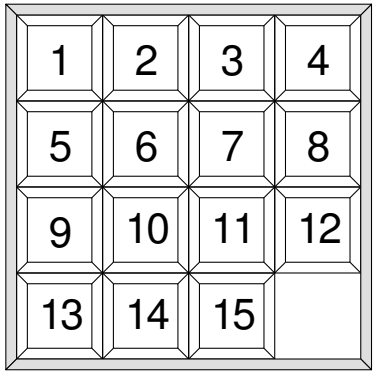
Mögliche Kodierung:

- Eine Konfiguration als Folge $v_1, v_2, \ldots, v_{16}$ der 16 Kacheln (inklusive der leeren Kachel)
- Die Kachel mit Nummer i als Hexadezimalzahl i, und
- das leere Feld mit der Hexadezimalzahl 0.

Die Länge der Kodierung ist somit $\sum_{i=1}^{16} \langle v_i \rangle = \sum_{i=1}^{16} 1 = 16$



D A B 6 5 7 4 8 1 C E 9 3 F 2 0



1 2 3 4 5 6 7 8 9 A B C D E F 0

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Allgemeine Wiederholung

Polynomiale Transformation

Eine **polynomiale Transformation** einer Sprache $L_1 \subseteq \Sigma_1^*$ in eine Sprache $L_2 \subseteq \Sigma_2^*$ ist eine Funktion $f : \Sigma_1^* \to \Sigma_2^*$ mit den Eigenschaften:

- es existiert eine polynomiale deterministische Turing-Maschine, die f berechnet;
- für alle $x \in \Sigma_1^*$ gilt: $x \in L_1 \Leftrightarrow f(x) \in L_2$.

Wir schreiben dann $L_1 \propto L_2$ (L_1 ist polynomial transformierbar in L_2).

Eine Sprache L heißt NP**–vollständig**, falls gilt:

- $L \in$ NP und
- für alle $L' \in$ NP gilt $L' \propto L$.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Polynomiale Transformation

Eine **polynomiale Transformation** einer Sprache $L_1 \subseteq \Sigma_1^*$ in eine Sprache $L_2 \subseteq \Sigma_2^*$ ist eine Funktion $f : \Sigma_1^* \to \Sigma_2^*$ mit den Eigenschaften:

- es existiert eine polynomiale deterministische Turing-Maschine, die f berechnet;
- für alle $x \in \Sigma_1^*$ gilt: $x \in L_1 \Leftrightarrow f(x) \in L_2$.

Wir schreiben dann $L_1 \propto L_2$ (L_1 ist polynomial transformierbar in L_2).

Eine Sprache L heißt NP**–vollständig**, falls gilt:

- $L \in$ NP und
- für alle $L' \in$ NP gilt $L' \propto L$.

→ Wenn ein NP-vollständiges Problem Π in P liegt, dann gilt NP=P.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Nichtdeterminismus

- Nicht praktischer Natur, da man nichtdet. Maschinen nicht wirklich bauen kann,
- theoretische Konstrukte sind hilfreich, da die Laufzeit von nichtdet. Maschinen deutlich geringer sind,
- liefern wichtige Hinweise auf effizientere Lösungen vieler praktischer Probleme (helfen mögliche Zunahme von Komplexität zu zeigen),
- det. Maschinen $\Rightarrow$ Berechnung bei gegebener Eingabe klar definiert,
- bei nicht deterministischen TM bei nur einem gelesenen Zeichen mehrere Nachfolgezustände möglich,
- keine einzelne Berechnung, sondern viele verschiedene Möglichkeiten der Berechnung,
- min. eine Berechnung führt in einen akzeptierenden Zustand.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Orakel

- Kann man sich als Black Box vorstellen,
- kann von einer TM befragt werden und Probleme in einem Schritt lösen.
- Orakel dienen dazu die Hierarchien von Berechenbarkeit und Komplexität zu definieren und deren Eigenschaften zu studieren.
- Ein geeignetes Orakel $\Rightarrow$ Berechenbarkeit verstärken oder Komplexität verringern.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik