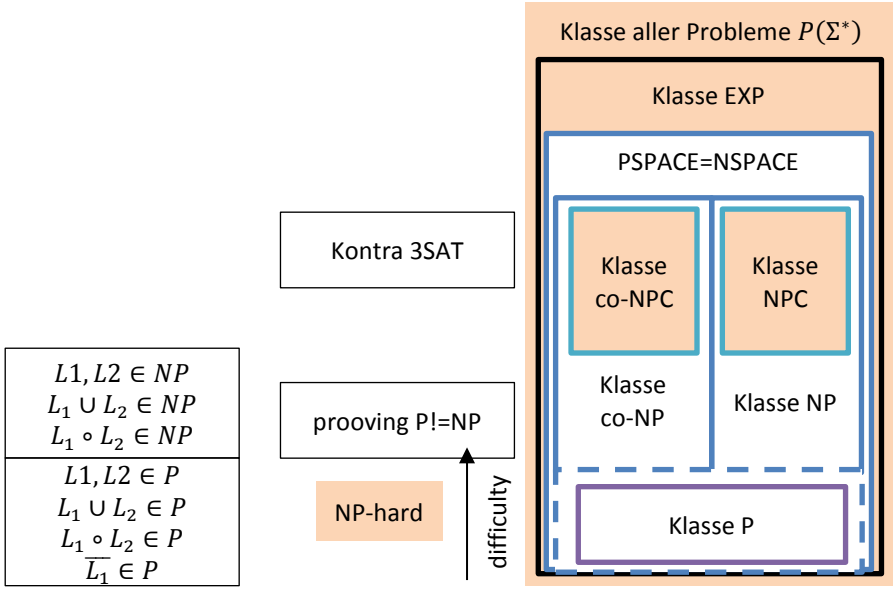
THEORETISCHE INFORMATIK OVERVIEW

Specification	Chomsky Typ	L: Sprachen	M: Machine (Implementation)	Komplexität (Zeit, Platz)	Tool zur Widerlegung
Regulärer Ausdruck/Grammatik RE: $a, \epsilon, \emptyset, \cup, \cdot, ^*$ (reg. Oper.)	3	reguläre Sprache $0^*1^*; 1^*; (01)^n$	DFA $Q, \Sigma, \delta: (Q \times \Sigma) \to Q, s \in Q, F \subseteq Q$ NFA: $Q, \Sigma, \delta: (Q \times \Sigma_\epsilon) \to \mathrm{P}(Q), s, F$	$t_M = O(n)$ $s_M = O(1)$	Pumping-Lemma
kontextfreie Grammatik $CFG: V, \Sigma, P(V \times (V \cup \Sigma)^*), S \in V$	2	kontextfreie Sprache $0^n1^n; 0^{n+1}1^n; 0^n\#1^n; 0^n10^n1$	PDA/Kellerautomat: (NFA mit Stack) «nicht behandelt»	$\max t_M = O(n^3)$ $\max s_M = O(n^2)$	Pumping-Lemma analoge Fassung
$A: \Sigma_u^* \xrightarrow{total} \{acc, rej\}$	1	entscheidbare A_{TM}: Akzeptanzproblem (rekursiv) $\{w\#w \mid w \in \{0,1\}\}; 1^{n^2}; a^nb^nc^n$	A: Akzeptanzmaschine (no Impl.) TM (Erkenner, Entscheider)	$\max t_M = O(n^x)$ $s_M = O(n)$	Diagonalisierung oder Reduktion
			TM: Turing Maschine $Q, \Sigma, \Gamma, \delta, q_s, q_{acc}, q_{rej}$		
$U: \Sigma_u^* \xrightarrow{total} \{acc, rej, \perp\}$		Turing-erkennbare A_{TM}: Akzeptanzproblem $HALT_{TM}$	U: Universalmaschine TM (Erkenner, kein Entscheider)		Komplement
	0	$\overline{A_{TM}}$ nicht Turing-erkennbar			

M akzeptiert ein Wort
M erkennt eine Sprache



Probleme	Komplexität	
Chess	verifizieren: $O(p(x^n))$	
Tic-tac-toe	Zeit: ∞ Platz: $O(p(n))$	
SAT, 3SAT, VC, IS, Clique, 3GC, Job-sequencing, max-cut, sudoku, protein folding, tetris, crossword-puzzles, knapsack		$L_1 \leq_p L_2$ $L_2 \leq_p L_1$
public-key crypt, prooving P==NP	lösen: schwierig verifizieren: $O(p(n))$	$\boldsymbol{NP} := \bigcup_{k>0} NTIME(n^k)$
Multipl, sort, search, Primes, rubic-cube, 2SAT, 2GC	lösen: $O(p(n))$	$\boldsymbol{P} := \bigcup_{k>0} DTIME(n^k)$