Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Christian Ivicevic

Sommersemester 2015
Studentisches Tutorium
Zusatzmaterial
17. Februar 2015

Theoretische Informatik

– Inhaltliche Vorschläge für das erweiterte Gedächtnis zur Klausur –

Maschinenmodelle

- **Typ 3**: DFA, NFA, ϵ-NFA, RE, r-CFG, l-CFG
- **Typ 2**: CFG, PDA
- **Typ 0**: TM, WHILE, GOTO, μR

Ardens Lemma

$$X \equiv \alpha X | \beta \Rightarrow X \equiv \alpha^* \beta$$
$$X = AX \cup B \Rightarrow X = A^* B \text{ für } \epsilon \notin A$$
$$\text{mit } \emptyset^* = \{\epsilon\},\ \alpha|\emptyset = \alpha,\ \alpha\emptyset = \emptyset$$

Synthesealgorithmus für die Chomsky-Normalform (CNF)

- Nützlichkeitstest
- ϵ-Elimination
- Kettenproduktionen eliminieren
- bestimmte Terminale ersetzen
- Ketten packen

Ackermann-Funktion

$$a(0, n) = n + 1$$
$$a(m + 1, 0) = a(m, 1)$$
$$a(m + 1, n + 1) = a(m, a(m + 1, n))$$

PR-Schema

$$f(0, \overline{x}) = g(\overline{x})$$
$$f(m + 1, \overline{x}) = h(f(m, \overline{x}), m, \overline{x})$$

μ-rekursive Umkehrfunktionen

Sei eine Funktion $f : \mathbb{N} \to \mathbb{N}$ gegeben, dann lässt sich mit $h(n, m) = 1 \dot{-} \mathrm{eq}(f(n), m)$ eine μ-rekursive Umkehrfunktion f^{-1} zu f konstruieren mit $f^{-1}(m) = (\mu h)(m)$.

Satz von Rice

Sei $F = \{f : \Sigma^* \to \Sigma^* \mid f \text{ ist berechenbar und besitzt die Eigenschaft} \ldots\}$ eine Funktionenmenge, dann ist die Menge $C_F = \{w \in \Sigma^* \mid \varphi_w \in F\}$ entscheidbar, falls F trivial ist, d.h. $F = \emptyset$ oder $F =$ alle berechenbaren Funktionen gilt (bzw. $C_F = \emptyset$ oder $C_F = \Sigma^*$).

Problem $A \in \mathcal{NP}$

Zwei Alternativen:

1. *Gödelisiere eine gültige Belegung als Zertifikat. Ein Verifikator prüft nun, dass die Belegung tatsächlich …. Der Verifikator ist offensichtlich polynomiell beschränkt.* – Hierbei den Satz an das aktuelle Problem anpassen.

2. Beweis durch Reduktion von A auf ein beliebiges (oder gegebenes!) $\mathcal{NP}$-vollständiges Problem.

   Beispiel: $A \leq_p SAT$. Das bedeutet, dass SAT mindestens so hart ist wie A.

Problem A ist $\mathcal{NP}$-hart

Polynomielle Reduktion eines $\mathcal{NP}$-vollständigen Problems, z.B. SAT, auf A: $SAT \leq_p A$.

Problem A ist $\mathcal{NP}$-vllständig

Hierzu muss gezeigt werden, dass $A \in \mathcal{NP}$ gilt und A zudem $\mathcal{NP}$-hart ist.

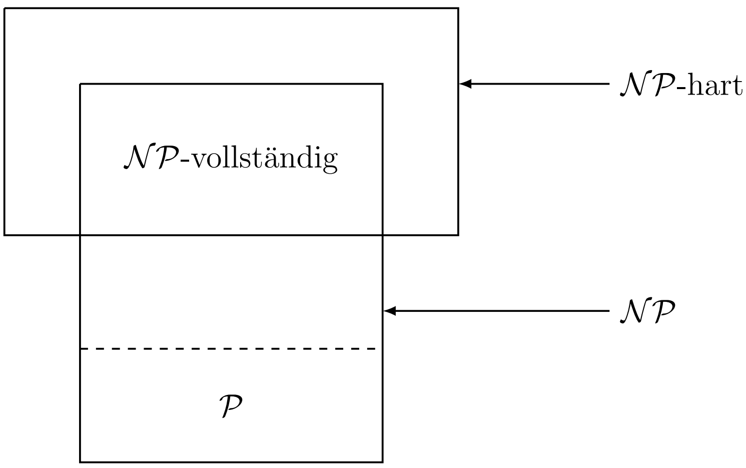

Abbildung 1: $\mathcal{P}$ und $\mathcal{NP}$-Hierarchie

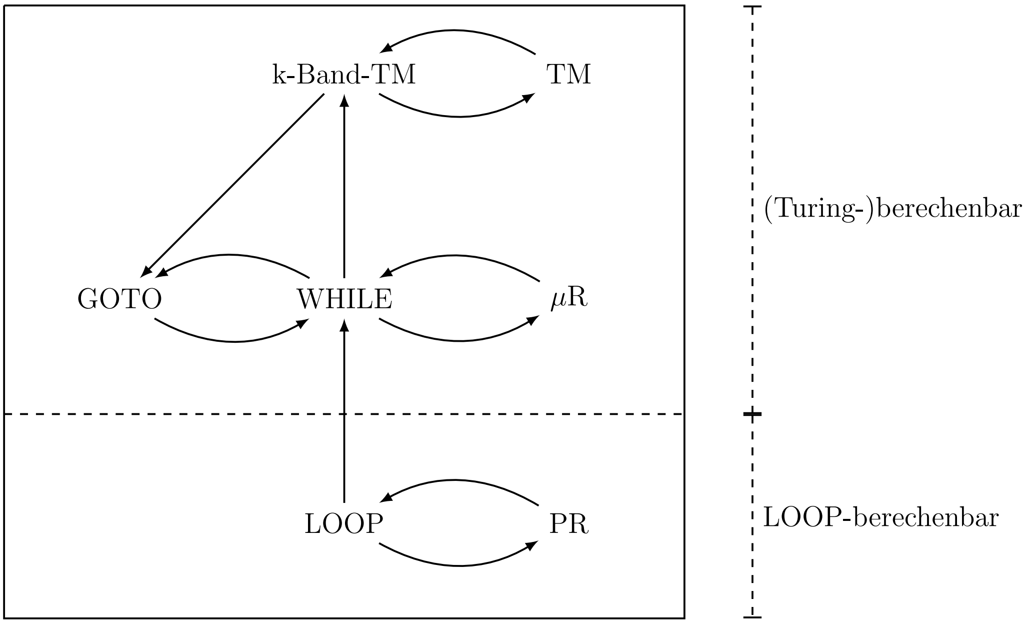

Abbildung 2: Berechenbarkeitsmodelle und ihre Transformationsrichtungen

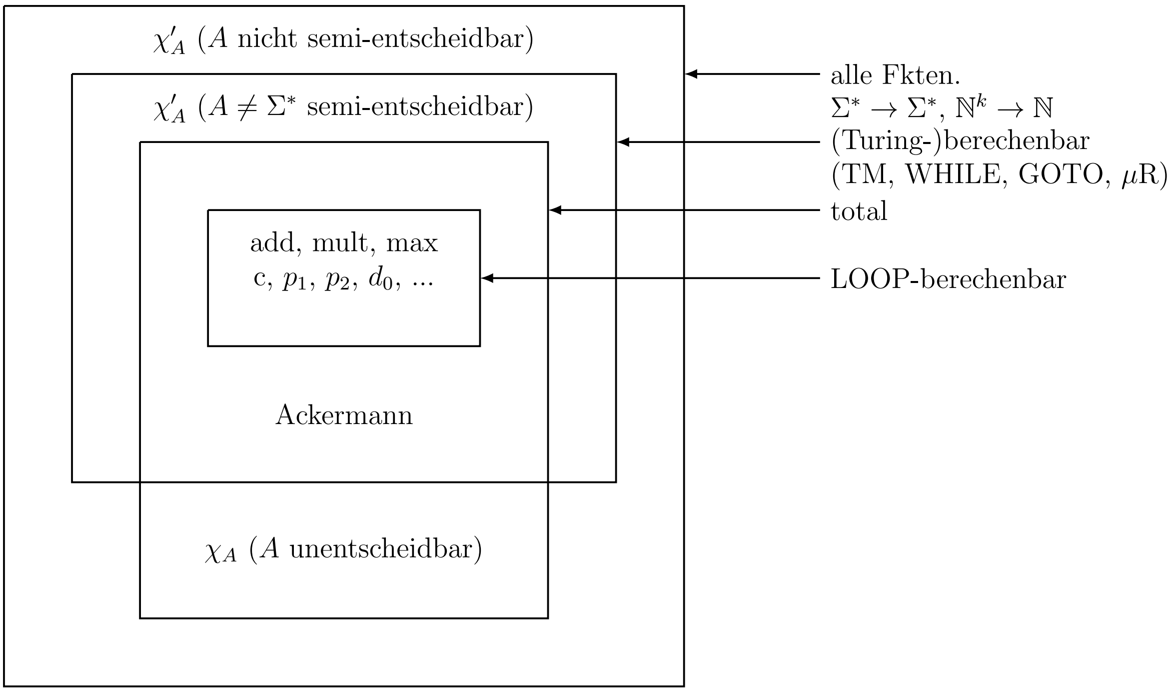

Abbildung 3: Berechenbarkeit und Entscheidbarkeit

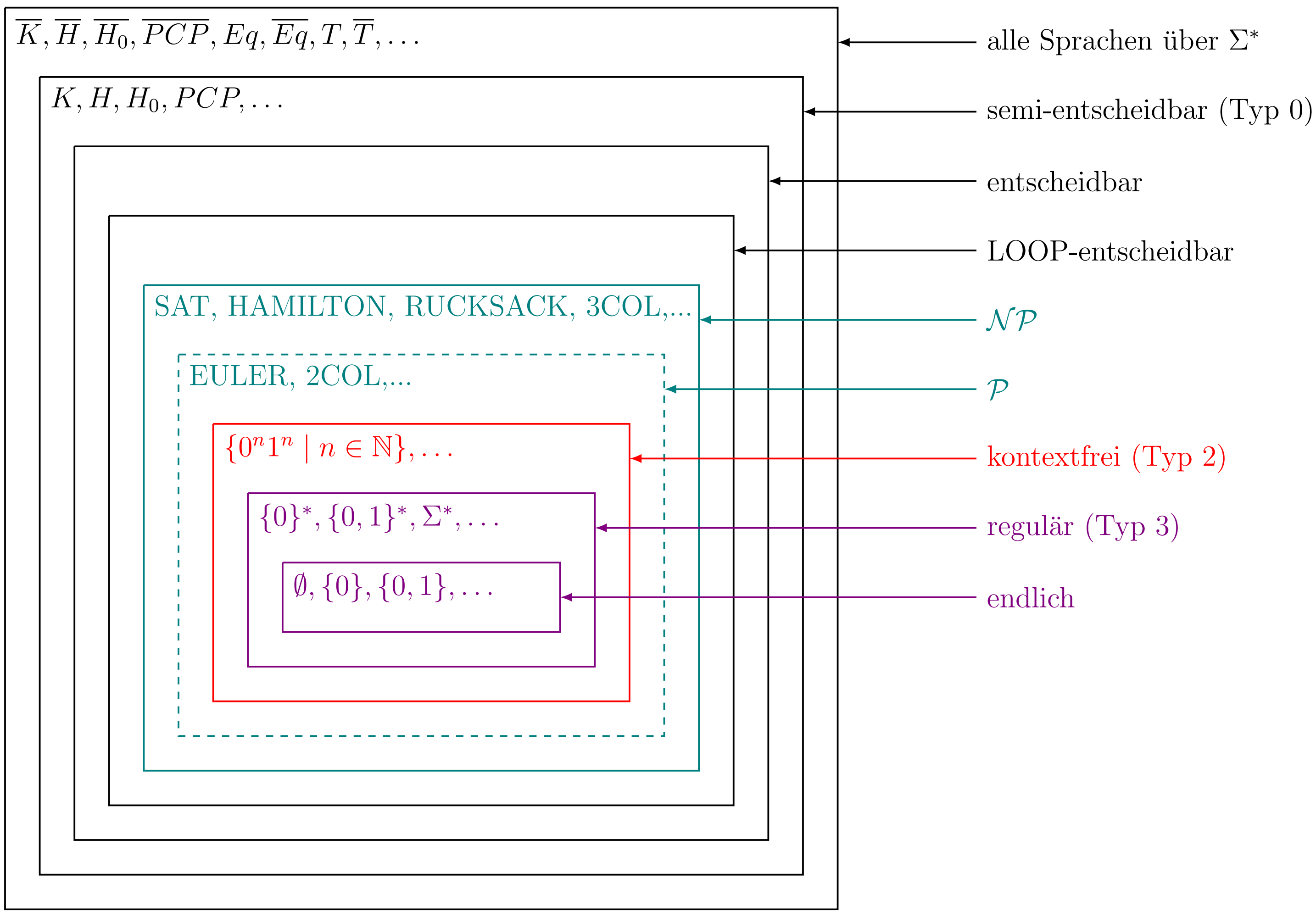

Abbildung 4: Hierarchie der wichtigsten Sprachen und Probleme