**Formale Sprachen und Automaten**

**1. Einführung**

**Berechenbarkeit:** Die Berechenbarkeit beschäftigt sich mit der Frage, welche Probleme von einem Computer berechenbar sind und welche Voraussetzungen dieser erfüllen muss, damit er alle berechenbaren Probleme lösen kann.

**Turing-Berechenbarkeit:** Ein Problem ist berechenbar, wenn es von einer Turing-Maschine berechnet werden kann.

**Church-Turing-These:** Die beiden Begriffe der Berechenbarkeit sind äquivalent, d.h. für jedes intuitiv berechenbare Problem gibt es auch eine Turingmaschine, die das Problem löst.

**Komplexitätstheorie:** Neben der Berechenbarkeit muss auch geprüft werden, ob ein Problem handhabbar ist, d.h. ob es auch für große Eingabewerte in vertretbarer Zeit gelöst werden kann:

* Algorithmen mit polynomialer Laufzeit gelten als handhabbar
* Algorithmen mit exponentieller Laufzeit gelten als nicht handhabbar

**Komplexitätsklassen:**

* P: Problem kann in polynomialer Zeit von einem deterministischen System berechnet werden
* NP: Problem kann in polynomialer Zeit von einem nichtdeterministischen System berechnet werden

**Formale Sprachen:** Formale Sprachen lassen sich als Analogie zu natürlichen Sprachen wie folgt definieren:

* Man wählt eine Menge von Buchstaben als Alphabet und erzeuge Worte durch Aneinanderhängen von Buchstaben aus dem Alphabet.
* Eine Teilmenge der gesamten Wortmenge bezeichnet eine formale Sprache.

**Automaten:** Automaten sind das akzeptierende Konzept: Eine Sprache gehört zu einer Sprachklasse, wenn es einen Automaten des vorgegebenen Automatentyps gibt, der die Sprache akzeptiert.

**Grammatik:** Grammatik ist das erzeugende Konzept: Eine Sprache gehört zur Sprachklasse, wenn es eine Grammatik des vorgegebenen Grammatiktyps gibt, der die Sprache erzeugt.

**2. Formale Sprachen und Grammatiken**