Mathematische Grundlagen der Computerlinguistik

Abbildungen, Reguläre Sprachen und kontextfreie Spachen

Michael Staniek

ICL, Universität Heidelberg, SoSe 2019

15.05.2019

Inhalte der heutigen Vorlesung

- Abbildungen/Funktionen
- Einstieg Formale Sprachen
- Endliche Automaten und reguläre Sprachen/Grammatiken
- Kontextfreie Sprachen und Grammatiken

Funktionen

Funktionen bzw. Abbildungen

Eine Relation ist eine totale Funktion, wenn sie linkstotal (definal) und rechtseindeutig (funktional) ist.

- Jedes Element in A hat mindestens ein Partnerelement in B
- Jedes Element in A hat höchstens ein Partnerelement in B

Funktionen

Funktionen bzw. Abbildungen

Eine Relation ist eine totale Funktion, wenn sie linkstotal (definal) und rechtseindeutig (funktional) ist.

- Jedes Element in A hat mindestens ein Partnerelement in B
- Jedes Element in A hat höchstens ein Partnerelement in B

Partielle Funktionen bzw. Abbildungen

Eine Relation ist eine partielle Funktion, wenn sie rechtseindeutig (funktional) ist.

- Jedes Element in A hat höchstens ein Partnerelement in B

Funktionen/Abbildungen

- Funktionen sind linkstotale und rechtseindeutige Relationen
- Ordnen allen Elementen in A ein Element in B zu

Funktionen/Abbildungen

- Funktionen sind linkstotale und rechtseindeutige Relationen
- Ordnen allen Elementen in A ein Element in B zu
- "Abbildung f bildet A auf B ab"

Funktionen/Abbildungen

- Funktionen sind linkstotale und rechtseindeutige Relationen
- Ordnen allen Elementen in A ein Element in B zu
- "Abbildung f bildet A auf B ab"
- "$f(x) \in B$ ist das Bild von $x \in A$ unter Abbildung f"

Funktionen/Abbildungen

- Funktionen sind linkstotale und rechtseindeutige Relationen
- Ordnen allen Elementen in A ein Element in B zu
- "Abbildung f bildet A auf B ab"
- "$f(x) \in B$ ist das Bild von $x \in A$ unter Abbildung f"
- f muss für jeden Wert in A definiert sein (sonst partielle Funktion)

Funktionen/Abbildungen

- Funktionen sind linkstotale und rechtseindeutige Relationen
- Ordnen allen Elementen in A ein Element in B zu
- "Abbildung f bildet A auf B ab"
- "$f(x) \in B$ ist das Bild von $x \in A$ unter Abbildung f"
- f muss für jeden Wert in A definiert sein (sonst partielle Funktion)

Schreibweise (Beispiel)

$$f : \mathbb{Z} \to \mathbb{N}_0$$
$$x \longmapsto x^2$$

Funktionen/Abbildungen

Abbildung f:

$$f : \mathbb{Z} \to \mathbb{N}_0$$
$$x \longmapsto x^2$$

Funktionen/Abbildungen

Abbildung f:

$$\begin{aligned} f : \mathbb{Z} &\to \mathbb{N}_0 \\ x &\longmapsto x^2 \end{aligned}$$

Abbildung g:

$$\begin{aligned} g : \mathbb{R} &\to \mathbb{R} \\ x &\longmapsto x^2 \end{aligned}$$

Funktionen/Abbildungen

- Für Abbildung $f : A \to B$:

Funktionen/Abbildungen

- Für Abbildung $f : A \to B$:
    - A: Definitionsbereich von f

Funktionen/Abbildungen

- Für Abbildung $f : A \rightarrow B$:
  - A: Definitionsbereich von f
  - B: Wertebereich von f

Funktionen/Abbildungen

- Für Abbildung $f : A \to B$:
  - A: Definitionsbereich von f
  - B: Wertebereich von f
- Angabe des Definitionsbereichs und Wertebereichs einer Abbildung unbedingt notwendig

Funktionen/Abbildungen

- Für Abbildung $f : A \to B$:
  - A: Definitionsbereich von f
  - B: Wertebereich von f
- Angabe des Definitionsbereichs und Wertebereichs einer Abbildung unbedingt notwendig
- $f : \mathbb{Z} \to \mathbb{N}_0$ und $g : \mathbb{R} \to \mathbb{R}$ unterscheiden sich in wichtigen Aspekten

Funktionen/Abbildungen

- Für Abbildung $f : A \to B$:
  - A: Definitionsbereich von f
  - B: Wertebereich von f
- Angabe des Definitionsbereichs und Wertebereichs einer Abbildung unbedingt notwendig
- $f : \mathbb{Z} \to \mathbb{N}_0$ und $g : \mathbb{R} \to \mathbb{R}$ unterscheiden sich in wichtigen Aspekten
  - Abzählbarkeit Definitionsbereich

Funktionen/Abbildungen

- Für Abbildung $f : A \to B$:
  - A: Definitionsbereich von f
  - B: Wertebereich von f
- Angabe des Definitionsbereichs und Wertebereichs einer Abbildung unbedingt notwendig
- $f : \mathbb{Z} \to \mathbb{N}_0$ und $g : \mathbb{R} \to \mathbb{R}$ unterscheiden sich in wichtigen Aspekten
  - Abzählbarkeit Definitionsbereich
  - Abzählbarkeit Wertebereich

Funktionen/Abbildungen

- Für Abbildung $f : A \to B$:
  - A: Definitionsbereich von f
  - B: Wertebereich von f
- Angabe des Definitionsbereichs und Wertebereichs einer Abbildung unbedingt notwendig
- $f : \mathbb{Z} \to \mathbb{N}_0$ und $g : \mathbb{R} \to \mathbb{R}$ unterscheiden sich in wichtigen Aspekten
  - Abzählbarkeit Definitionsbereich
  - Abzählbarkeit Wertebereich
  - Stetigkeit und Differenzierbarkeit in $\mathbb{R}$

Funktionen/Abbildungen

- Für Abbildung $f : A \to B$:
  - A: Definitionsbereich von f
  - B: Wertebereich von f
- Angabe des Definitionsbereichs und Wertebereichs einer Abbildung unbedingt notwendig
- $f : \mathbb{Z} \to \mathbb{N}_0$ und $g : \mathbb{R} \to \mathbb{R}$ unterscheiden sich in wichtigen Aspekten
  - Abzählbarkeit Definitionsbereich
  - Abzählbarkeit Wertebereich
  - Stetigkeit und Differenzierbarkeit in $\mathbb{R}$
- Wichtige Eigenschaften unter optimierungstheoretischen Gesichtspunkten

Bild und Urbild

Bildmenge und Urbildmenge

Sei $f : X \to Y$ eine Abbildung und $A \subseteq X$ und $B \subseteq Y$. Dann heißt die Menge

$$f(A) := \{f(x)|x \in A\}$$

die Bildmenge von A oder "das Bild von A" und die Menge

Bild und Urbild

Bildmenge und Urbildmenge

Sei $f : X \to Y$ eine Abbildung und $A \subseteq X$ und $B \subseteq Y$. Dann heißt die Menge

$$f(A) := \{f(x) | x \in A\}$$

die Bildmenge von A oder "das Bild von A" und die Menge

$$f^{-1}(B) := \{x | f(x) \in B\}$$

die Urbildmenge von B oder "das Urbild von B."

Bild und Urbild

Bildmenge und Urbildmenge

Sei $f : X \to Y$ eine Abbildung und $A \subseteq X$ und $B \subseteq Y$. Dann heißt die Menge

$$f(A) := \{f(x) | x \in A\}$$

die Bildmenge von A oder "das Bild von A" und die Menge

$$f^{-1}(B) := \{x | f(x) \in B\}$$

die Urbildmenge von B oder "das Urbild von B."

- Bildmenge und Urbildmenge sind immer definiert mit Bezug auf eine Teilmenge des Definitions- bzw. Wertebereichs

Bild und Urbild

Bildmenge und Urbildmenge

Sei $f : X \to Y$ eine Abbildung und $A \subseteq X$ und $B \subseteq Y$. Dann heißt die Menge

$$f(A) := \{f(x) | x \in A\}$$

die Bildmenge von A oder "das Bild von A" und die Menge

$$f^{-1}(B) := \{x | f(x) \in B\}$$

die Urbildmenge von B oder "das Urbild von B."

- Bildmenge und Urbildmenge sind immer definiert mit Bezug auf eine Teilmenge des Definitions- bzw. Wertebereichs
- Nicht zu verwechseln mit Definitions- bzw. Wertebereich selbst

Funktionen/Abbildungen

Zusammengesetzte Abbildungen

Sind $f : X \to Y$ und $g : Y \to Z$ Abbildungen, so sei die zusammengetzte Abbildung $g \circ f$ durch

$$X \to Z$$

$$x \longmapsto g(f(x))$$

definiert.

Funktionen/Abbildungen

Zusammengesetzte Abbildungen

Sind $f : X \to Y$ und $g : Y \to Z$ Abbildungen, so sei die zusammengetzte Abbildung $g \circ f$ durch

$$X \to Z$$

$$x \longmapsto g(f(x))$$

definiert.

- Definitionsbereich von g muss Wertebereich von f sein (für totale Funktion)

Abbildungen: Umkehrrelation (allgemein)

Umkehrrelation

Sei $R \subseteq A \times B$ eine Relation.
Die Relation

$$R^{-1} := \{(b, a) \in B \times A | (a, b) \in R\}$$

heißt Umkehrrelation oder konverse Relation von R.

Zur Erinnerung: Bejektivität bei Relationen

Bijektive Relationen

Eine Relation $R \subseteq A \times B$ heißt bijektiv genau dann, wenn:

$$\forall b \in B : \exists! a \in A : (a, b) \in R$$

- jedes Element in B hat genau ein Partnerelement in A

Zur Erinnerung: Bejektivität bei Relationen

Bijektive Relationen

Eine Relation $R \subseteq A \times B$ heißt bijektiv genau dann, wenn:

$$\forall b \in B : \exists! a \in A : (a, b) \in R$$

- jedes Element in B hat genau ein Partnerelement in A
- Ist eine bijektive Relation außerdem linkstotal und rechtseindeutig, so spricht man von einer bijektiven Funktion oder Abbildung

Abbildungen: Umkehrabbildung

Umkehrabbildung

Ist die Abbildung $f : X \to Y$ bijektiv, so heißt:

$$f^{-1} : Y \to X$$
$$f(x) \longmapsto x$$

die Umkehrabbildung von f.

Abbildungen: Umkehrabbildung

Umkehrabbildung

Ist die Abbildung $f : X \to Y$ bijektiv, so heißt:

$$f^{-1} : Y \to X$$
$$f(x) \longmapsto x$$

die Umkehrabbildung von f.

- Umkehrabbildung eine Abbildung f ist nicht zu verwechseln mit dem Urbild einer bestimmten Menge unter f

Abbildungen: Umkehrabbildung

Umkehrabbildung

Ist die Abbildung $f : X \to Y$ bijektiv, so heißt:

$$f^{-1} : Y \to X$$

$$f(x) \longmapsto x$$

die Umkehrabbildung von f.

- Umkehrabbildung eine Abbildung f ist nicht zu verwechseln mit dem Urbild einer bestimmten Menge unter f
- Für nicht-bijektive Funktionen lässt sich keine Umkehrabbildung definieren

Abbildungen: Umkehrabbildung

Umkehrabbildung

Ist die Abbildung $f : X \to Y$ bijektiv, so heißt:

$$f^{-1} : Y \to X$$

$$f(x) \longmapsto x$$

die Umkehrabbildung von f.

- Umkehrabbildung eine Abbildung f ist nicht zu verwechseln mit dem Urbild einer bestimmten Menge unter f
- Für nicht-bijektive Funktionen lässt sich keine Umkehrabbildung definieren
- Aber: Umkehrrelation der entsprechenden Relation

Formale Sprachen

- Was ist eine Sprache?

Formale Sprachen

- Was ist eine Sprache?
- Nur im Kontext einer Disziplin sinnvoll zu beantworten

Formale Sprachen

- Was ist eine Sprache?
- Nur im Kontext einer Disziplin sinnvoll zu beantworten
- Formal gesehen: Menge von Zeichenketten

Alphabet

Ein Alphabet ist eine endliche Menge von Zeichen.

$$\{a, b, ..., z\}$$

$$\{Mary, had, a, little, lamb\}$$

Alphabete formaler Sprachen

Alphabete formaler Sprachen

- Alphabete müssen endlich sein
- Zeichen eines Alphabets sollten unterscheidbar und hinsichtlich der gewünschten Beschreibungsebene“atomar” sein

Alphabete formaler Sprachen

- Alphabete müssen endlich sein
- Zeichen eines Alphabets sollten unterscheidbar und hinsichtlich der gewünschten Beschreibungsebene“atomar” sein
  - z.B.: Wörter bestehen aus Buchstaben, aber beim Parsing/ der syntaktischen Analyse einer natürlichen Sprache interessiert uns nur die Struktur des Satzes, nicht die der Wörter
  - anders bei Morphologie (Analyse der Wortstruktur), etc.

Alphabete formaler Sprachen

- Alphabete müssen endlich sein
- Zeichen eines Alphabets sollten unterscheidbar und hinsichtlich der gewünschten Beschreibungsebene“atomar” sein
  - z.B.: Wörter bestehen aus Buchstaben, aber beim Parsing/ der syntaktischen Analyse einer natürlichen Sprache interessiert uns nur die Struktur des Satzes, nicht die der Wörter
  - anders bei Morphologie (Analyse der Wortstruktur), etc.
  - Indiz: Menschen, die nicht Lesen oder Schreiben können, können trotzdem “richtige” von “falschen” Sätzen in ihrer Muttersprache unterscheiden
    - Ich gehe heute ins Kino
    - *Kino gehe ins ich heute
- Sonstige Eigenschaften des Alphabets aus formaler Sicht irrelevant

Formale Sprachen

Reguläre Grammatik

Eine reguläre Grammatik G ist ein 4-Tupel (N, Σ, R, S) wobei

- Σ eine endliche Menge von Terminalsymbolen, das Alphabet von G ist

Formale Sprachen

Reguläre Grammatik

Eine reguläre Grammatik G ist ein 4-Tupel (N, Σ, R, S) wobei

- Σ eine endliche Menge von Terminalsymbolen, das Alphabet von G ist
- N eine endliche Menge von Nichtterminalsymbolen ist

Formale Sprachen

Reguläre Grammatik

Eine reguläre Grammatik G ist ein 4-Tupel (N, Σ, R, S) wobei

- Σ eine endliche Menge von Terminalsymbolen, das Alphabet von G ist
- N eine endliche Menge von Nichtterminalsymbolen ist
- $S \in N$ das Startsymbol von G ist

Formale Sprachen

Reguläre Grammatik

Eine reguläre Grammatik G ist ein 4-Tupel (N, Σ, R, S) wobei

- Σ eine endliche Menge von Terminalsymbolen, das Alphabet von G ist
- N eine endliche Menge von Nichtterminalsymbolen ist
- $S \in N$ das Startsymbol von G ist
- R eine Menge von Produktionsregeln der Form
  - $R \subseteq N \times (\{\epsilon\} \cup \Sigma \cup (\Sigma \circ N))$ (rechtsreguläre Grammatik)
  - $R \subseteq N \times (\{\epsilon\} \cup \Sigma \cup (N \circ \Sigma))$ (linksreguläre Grammatik)
- $\circ$: Verkettung von Sprachen: $A \circ B := \{ab | a \in A, b \in B\}$
- ϵ: Leeres Wort ("empty string")

Formale Sprachen

Reguläre Grammatik

Eine reguläre Grammatik G ist ein 4-Tupel (N, Σ, R, S) wobei

- Σ eine endliche Menge von Terminalsymbolen, das Alphabet von G ist
- N eine endliche Menge von Nichtterminalsymbolen ist
- $S \in N$ das Startsymbol von G ist
- R eine Menge von Produktionsregeln der Form
  - $R \subseteq N \times (\{\epsilon\} \cup \Sigma \cup (\Sigma \circ N))$ (rechtsreguläre Grammatik)
  - $R \subseteq N \times (\{\epsilon\} \cup \Sigma \cup (N \circ \Sigma))$ (linksreguläre Grammatik)
- $\circ$: Verkettung von Sprachen: $A \circ B := \{ab | a \in A, b \in B\}$
- ϵ: Leeres Wort ("empty string")
- Alle Zeichenketten, die durch rekursive Anwendungen von Regeln in R auf S erzeugt werden können, sind Teil der von G definierten regulären Sprache

Formale Sprachen

Erlaubte Regelformen in regulären Grammatiken:

$$X, Y \in N$$

$$a \in \Sigma$$

rechtsreguläre Grammatiken:

$$X \rightarrow aY$$

$$X \rightarrow a$$

$$X \rightarrow \epsilon$$

linksreguläre Grammatiken:

$$X \rightarrow Ya$$

$$X \rightarrow a$$

$$X \rightarrow \epsilon$$

Recap: Nichtdeterministische endliche Automaten

NEA

Ein nichtdeterministischer endlicher Automat A ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$ wobei

Recap: Nichtdeterministische endliche Automaten

NEA

Ein nichtdeterministischer endlicher Automat A ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$ wobei

- Q eine endliche Zustandsmenge,

Recap: Nichtdeterministische endliche Automaten

NEA

Ein nichtdeterministischer endlicher Automat A ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$ wobei

- Q eine endliche Zustandsmenge,
- Σ ein endliches Alphabet,

Recap: Nichtdeterministische endliche Automaten

NEA

Ein nichtdeterministischer endlicher Automat A ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$ wobei

- Q eine endliche Zustandsmenge,
- Σ ein endliches Alphabet,
- $\delta : (Q \times \Sigma) \times Q$ eine Übergangsrelation,

Recap: Nichtdeterministische endliche Automaten

NEA

Ein nichtdeterministischer endlicher Automat A ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$ wobei

- Q eine endliche Zustandsmenge,
- Σ ein endliches Alphabet,
- $\delta : (Q \times \Sigma) \times Q$ eine Übergangsrelation,
- q_0 ein Startzustand und

Recap: Nichtdeterministische endliche Automaten

NEA

Ein nichtdeterministischer endlicher Automat A ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$ wobei

- Q eine endliche Zustandsmenge,
- Σ ein endliches Alphabet,
- $\delta : (Q \times \Sigma) \times Q$ eine Übergangsrelation,
- q_0 ein Startzustand und
- $F \subseteq Q$ eine Menge an Endzuständen ist

Recap: Nichtdeterministische endliche Automaten

NEA

Ein nichtdeterministischer endlicher Automat A ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$ wobei

- Q eine endliche Zustandsmenge,
- Σ ein endliches Alphabet,
- $\delta : (Q \times \Sigma) \times Q$ eine Übergangsrelation,
- q_0 ein Startzustand und
- $F \subseteq Q$ eine Menge an Endzuständen ist
- Übergangsrelation bestimmt, in welchen Zuständen welche Zeichen "erkannt" werden und definiert somit eine Sprache

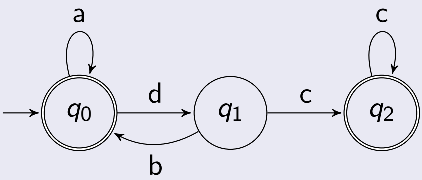
Recap: Deterministische endliche Automaten

DEA

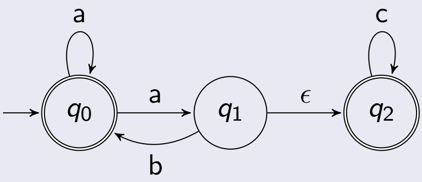
Ein deterministischer endlicher Automat A ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$ wobei

- Q eine endliche Zustandsmenge,
- Σ ein endliches Alphabet,
- $\delta : (Q \times \Sigma) \rightarrow Q$ eine Übergangsfunktion,
- q_0 ein Startzustand und
- $F \subseteq Q$ eine Menge an Endzuständen ist
- Jeder NEA kann äquivalent als DEA formuliert werden (Teilmengenkonstruktion; ohne Beweis)

Beispiel: Deterministischer Endlicher Automat



Beispiel: Nichtdeterministischer Endlicher Automat



Sprachen, Grammatiken und Automaten

Sprachen und Grammatiken

- Eine Grammatik beschreibt eine Sprache aus allen Wörtern, die durch rekursive Anwendung der Regeln auf das Startsymbol erzeugt werden können, wobei die erzeugten Wörter
  - nur aus Terminalsymbolen bestehen und

Sprachen, Grammatiken und Automaten

Sprachen und Grammatiken

- Eine Grammatik beschreibt eine Sprache aus allen Wörtern, die durch rekursive Anwendung der Regeln auf das Startsymbol erzeugt werden können, wobei die erzeugten Wörter
  - nur aus Terminalsymbolen bestehen und
  - am Ende keine nicht-expandierten Nichtterminalsymbole vorhanden sein dürfen

Sprachen, Grammatiken und Automaten

Sprachen, Grammatiken und Automaten

Sprachen und Automaten

- Ein endlicher Automat beschreibt eine Sprache aus allen Wörtern, die entstehen,

Sprachen, Grammatiken und Automaten

Sprachen und Automaten

- Ein endlicher Automat beschreibt eine Sprache aus allen Wörtern, die entstehen,
  - wenn man beim Startzustand beginnt,

Sprachen, Grammatiken und Automaten

Sprachen und Automaten

- Ein endlicher Automat beschreibt eine Sprache aus allen Wörtern, die entstehen,
  - wenn man beim Startzustand beginnt,
  - Übergängen folgt,

Sprachen, Grammatiken und Automaten

Sprachen und Automaten

- Ein endlicher Automat beschreibt eine Sprache aus allen Wörtern, die entstehen,
  - wenn man beim Startzustand beginnt,
  - Übergängen folgt,
  - dabei die Symbole der Übergänge ausgibt

Sprachen, Grammatiken und Automaten

Sprachen und Automaten

- Ein endlicher Automat beschreibt eine Sprache aus allen Wörtern, die entstehen,
  - wenn man beim Startzustand beginnt,
  - Übergängen folgt,
  - dabei die Symbole der Übergänge ausgibt
  - und in einen Endzustand gelangt.

Sprachen, Grammatiken und Automaten

- Gibt es für jeden endlichen Automaten eine entsprechende Grammatik?

Sprachen, Grammatiken und Automaten

- Gibt es für jeden endlichen Automaten eine entsprechende Grammatik?
- Gibt es für jede Grammatik einen entsprechenden endlichen Automaten?

DEAs/NEAs und reguläre Grammatiken

Äquivalenzbeweis NEAs und reg. Grammatiken (Beweisskizze)

Überführung eines NEA in eine (rechts-)reguläre Grammatik:

DEAs/NEAs und reguläre Grammatiken

Äquivalenzbeweis NEAs und reg. Grammatiken (Beweisskizze)

Überführung eines NEA in eine (rechts-)reguläre Grammatik:

- $N := Q$

DEAs/NEAs und reguläre Grammatiken

Äquivalenzbeweis NEAs und reg. Grammatiken (Beweisskizze)

Überführung eines NEA in eine (rechts-)reguläre Grammatik:

- $N := Q$
- $S := q_0$

DEAs/NEAs und reguläre Grammatiken

Äquivalenzbeweis NEAs und reg. Grammatiken (Beweisskizze)

Überführung eines NEA in eine (rechts-)reguläre Grammatik:

- $N := Q$
- $S := q_0$
- $\Sigma_A := \Sigma_G$

DEAs/NEAs und reguläre Grammatiken

Äquivalenzbeweis NEAs und reg. Grammatiken (Beweisskizze)

Überführung eines NEA in eine (rechts-)reguläre Grammatik:

- $N := Q$
- $S := q_0$
- $\Sigma_A := \Sigma_G$
- Für jeden Übergang $((q_n, a), q_f), q_f \in F$, füge R die Regel $q_n \rightarrow a$ hinzu.

DEAs/NEAs und reguläre Grammatiken

Äquivalenzbeweis NEAs und reg. Grammatiken (Beweisskizze)

Überführung eines NEA in eine (rechts-)reguläre Grammatik:

- $N := Q$
- $S := q_0$
- $\Sigma_A := \Sigma_G$
- Für jeden Übergang $((q_n, a), q_f), q_f \in F$, füge R die Regel $q_n \rightarrow a$ hinzu.
- Für jeden Übergang $((q_n, a), q_m)$, füge R die Regel $q_n \rightarrow aq_m$ hinzu.

DEAs/NEAs und reguläre Grammatiken

Äquivalenzbeweis NEAs und reg. Grammatiken (Beweisskizze)

Überführung eines NEA in eine (rechts-)reguläre Grammatik:

- $N := Q$
- $S := q_0$
- $\Sigma_A := \Sigma_G$
- Für jeden Übergang $((q_n, a), q_f), q_f \in F$, füge R die Regel $q_n \to a$ hinzu.
- Für jeden Übergang $((q_n, a), q_m)$, füge R die Regel $q_n \to aq_m$ hinzu.
- Für jeden Zustand $q_f \in F$, füge R die Regel $q_f \to \epsilon$ hinzu.

DEAs/NEAs und reguläre Grammatiken

Äquivalenzbeweis NEAs und reg. Grammatiken (Beweisskizze)

Überführung einer (rechts-)regulären Grammatik in einen NEA:

DEAs/NEAs und reguläre Grammatiken

Äquivalenzbeweis NEAs und reg. Grammatiken (Beweisskizze)

Überführung einer (rechts-)regulären Grammatik in einen NEA:

- $Q := N \cup \{q_f\}$

DEAs/NEAs und reguläre Grammatiken

Äquivalenzbeweis NEAs und reg. Grammatiken (Beweisskizze)

Überführung einer (rechts-)regulären Grammatik in einen NEA:

- $Q := N \cup \{q_f\}$
- $q_0 := S$

DEAs/NEAs und reguläre Grammatiken

Äquivalenzbeweis NEAs und reg. Grammatiken (Beweisskizze)

Überführung einer (rechts-)regulären Grammatik in einen NEA:

- $Q := N \cup \{q_f\}$
- $q_0 := S$
- $F := \{q_f\}$

DEAs/NEAs und reguläre Grammatiken

Äquivalenzbeweis NEAs und reg. Grammatiken (Beweisskizze)

Überführung einer (rechts-)regulären Grammatik in einen NEA:

- $Q := N \cup \{q_f\}$
- $q_0 := S$
- $F := \{q_f\}$
- $\Sigma_G := \Sigma_A$

DEAs/NEAs und reguläre Grammatiken

Äquivalenzbeweis NEAs und reg. Grammatiken (Beweisskizze)

Überführung einer (rechts-)regulären Grammatik in einen NEA:

- $Q := N \cup \{q_f\}$
- $q_0 := S$
- $F := \{q_f\}$
- $\Sigma_G := \Sigma_A$
- Jede Regel der Grammatik hat eine der folgenden Formen:
  - I: $q_n \to a, a \in \Sigma \cup \{\epsilon\}$
  - II: $q_n \to aB, a \in \Sigma, B \in N$
- Erweitere δ für jede Regel vom Typ I um den Übergang $((q_n, a), q_f)$
- Erweitere δ für jede Regel vom Typ II um den Übergang $((q_n, a), B)$

Formale Sprachen - kontextfreie Sprachen

Kontextfreie Grammatik

Eine kontextfreie Grammatik G ist ein 4-Tupel (N, Σ, R, S) wobei

- Σ eine endliche Menge von Terminalsymbolen, das Alphabet von G ist

Formale Sprachen - kontextfreie Sprachen

Kontextfreie Grammatik

Eine kontextfreie Grammatik G ist ein 4-Tupel (N, Σ, R, S) wobei

- Σ eine endliche Menge von Terminalsymbolen, das Alphabet von G ist
- N eine endliche Menge von Nichtterminalsymbolen ist

Formale Sprachen - kontextfreie Sprachen

Kontextfreie Grammatik

Eine kontextfreie Grammatik G ist ein 4-Tupel (N, Σ, R, S) wobei

- Σ eine endliche Menge von Terminalsymbolen, das Alphabet von G ist
- N eine endliche Menge von Nichtterminalsymbolen ist
- $S \in N$ das Startsymbol von G ist

Formale Sprachen - kontextfreie Sprachen

Kontextfreie Grammatik

Eine kontextfreie Grammatik G ist ein 4-Tupel (N, Σ, R, S) wobei

- Σ eine endliche Menge von Terminalsymbolen, das Alphabet von G ist
- N eine endliche Menge von Nichtterminalsymbolen ist
- $S \in N$ das Startsymbol von G ist
- R eine endliche Menge von Produktionsregeln ist:
  - $R \subseteq N \times (\Sigma \cup N)^*$
- A^* bezeichnet die Kleenesche Hülle einer Sprache A:

$$A^* := \cup_{i \in N_0} A^i, A^0 := \epsilon, A^{n+1} := A^n \circ A$$

Formale Sprachen - kontextfreie Sprachen

Kontextfreie Grammatik

Eine kontextfreie Grammatik G ist ein 4-Tupel (N, Σ, R, S) wobei

- Σ eine endliche Menge von Terminalsymbolen, das Alphabet von G ist
- N eine endliche Menge von Nichtterminalsymbolen ist
- $S \in N$ das Startsymbol von G ist
- R eine endliche Menge von Produktionsregeln ist:
  - $R \subseteq N \times (\Sigma \cup N)^*$
- A^* bezeichnet die Kleenesche Hülle einer Sprache A:

$$A^* := \cup_{i \in N_0} A^i, A^0 := \epsilon, A^{n+1} := A^n \circ A$$

- Alle Zeichenketten, die durch rekursive Anwendungen von Regeln in R auf S erzeugt werden können, sind Teil der von G definierten kontextfreien Sprache

Formale Sprachen - kontextfreie Sprachen

Regelformen in kontextfreien Grammatiken:

$$X, A, B, C \in N$$

$$a, b, c \in \Sigma$$

Erlaubte Regeln (Beispiele):

$$X \rightarrow \epsilon$$

$$X \rightarrow a$$

$$X \rightarrow aB$$

$$X \rightarrow CaB$$

$$X \rightarrow X$$

$$X \rightarrow XCa$$

Formale Sprachen - kontextfreie Sprachen

Regelformen in kontextfreien Grammatiken:

$$X, A, B, C \in N$$
$$a, b, c \in \Sigma$$

Keine erlaubten Regeln (Beispiele):

$$XA \to a$$

$$XA \to XA$$
$$XB \to A$$

$$a \to C$$

$$X \to (aB)^*$$

Formale Sprachen - kontextfreie Sprachen

Chomsky-Normalform

Eine kontextfreie Grammatik entspricht dann der Chomsky-Normalform, wenn ihre Regeln folgende Form haben:

- $X \rightarrow YZ, X, Y, Z \in N$

Formale Sprachen - kontextfreie Sprachen

Chomsky-Normalform

Eine kontextfreie Grammatik entspricht dann der Chomsky-Normalform, wenn ihre Regeln folgende Form haben:

- $X \to YZ, X, Y, Z \in N$
- $Y \to a, X \in N, a \in \Sigma$

Formale Sprachen - kontextfreie Sprachen

Chomsky-Normalform

Eine kontextfreie Grammatik entspricht dann der Chomsky-Normalform, wenn ihre Regeln folgende Form haben:

- $X \rightarrow YZ, X, Y, Z \in N$
- $Y \rightarrow a, X \in N, a \in \Sigma$
- $S \rightarrow \epsilon$

Formale Sprachen - kontextfreie Sprachen

Chomsky-Normalform

Eine kontextfreie Grammatik entspricht dann der Chomsky-Normalform, wenn ihre Regeln folgende Form haben:

- $X \rightarrow YZ, X, Y, Z \in N$
- $Y \rightarrow a, X \in N, a \in \Sigma$
- $S \rightarrow \epsilon$
- Auf der rechten Regelseite darf ein Terminalsymbol oder zwei Nichtterminalsymbole stehen
- Das leere Wort darf nur dann auf der rechten Regelseite stehen, wenn links das Startsymbol steht

Formale Sprachen - kontextfreie Sprachen

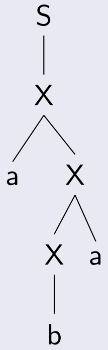
Eindeutige und mehrdeutige Grammatiken

- Wenn eine Grammatik G ein Wort w durch unterschiedliche Sequenzen an Regelanwendungen erzeugen und dabei unterschiedliche Syntaxbäume generieren kann, dann ist G *mehrdeutig hinsichtlich* w,
- ansonsten ist G *eindeutig hinsichtlich* w.
- Gibt es mindestens ein w, hinsichtlich dessen G mehrdeutig ist, so ist G mehrdeutig
- Gibt es keine solchen Wörter in der von G erkannten Sprache, so ist G eindeutig

Formale Sprachen - kontextfreie Sprachen

Mehrdeutige Grammatik:

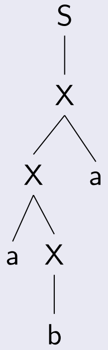
$$G := (\{S, X\}, \{a, b\}, R, S)$$
$$R := \{S \to X, X \to aX, X \to Xa,\ X \to b\}$$



Formale Sprachen - kontextfreie Sprachen

Mehrdeutige Grammatik:

$$G := (\{S, X\}, \{a, b\}, R, S)$$
$$R := \{S \to X, X \to aX, X \to Xa,\ X \to b\}$$



Noch Fragen?

Weiterführende Literatur

Dan Jurafsky und James H. Martin , *Speech and Language Processing*, dritte Ausgabe. 11. Kapitel: `https://web.stanford.edu/~jurafsky/slp3/11.pdf`

Klaus Jänich, *Lineare Algebra*, Springer, 2017. Seiten 1-19

Vielen Dank für die Aufmerksamkeit!