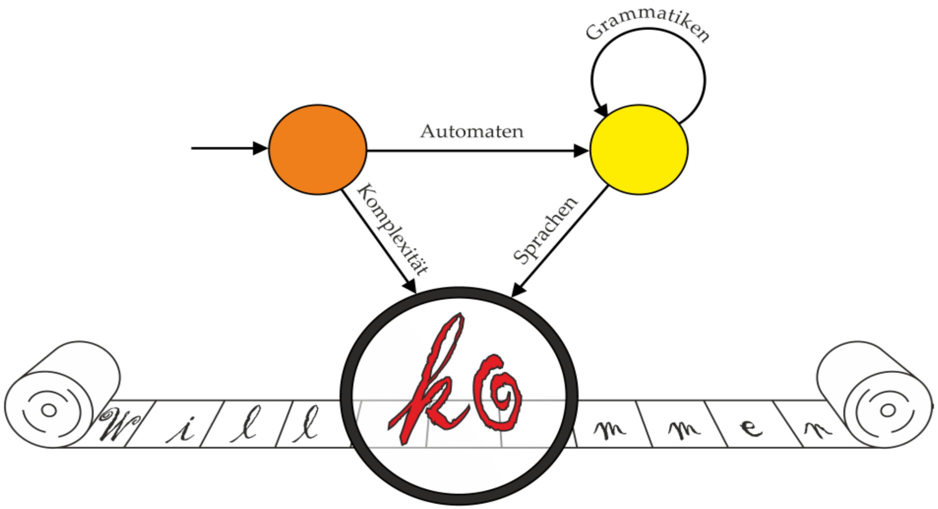
Grundlagen der Informatik II

Tutorium 4

Institut für Angewandte Informatik und Formale Beschreibungsverfahren – Prof. Tatiana von Landesberger



Vorab…

- Bonusklausur
  - Dienstag, 14.01.2019, 19:30 Uhr
  - Inhalte der Übungsblätter 1-4
  - Anmeldung: von 04.12.2019 bis 05.01.2020 über https://portal.wiwi.kit.edu/ys/3281/signup
  - **Bestehen von 2 aus 3 Aufgaben ist Voraussetzung zur Erreichung des Bonus**

Einführungsaufgabe: Kontextfreie Sprachen

- Gegeben sei eine Sprache *L* von Palindromen der Länge 8:
- $L = \{vv' \mid v \in \{a, b\}^*, |v| = 4\}$
- Es gilt beispielsweise: aabbbbaa $\in L$.
- Geben Sie eine Grammatik $G = (N, T, P, S)$ vollständig an, sodass gilt $L(G) = L$.

- **Lösung**:
- $G = (\{S, A, B, C\}, \{a, b\}, P, S)$,
- $P = \{$S → aAa | bAb,

  A → aBa | bBb,

  B → aCa | bCb,

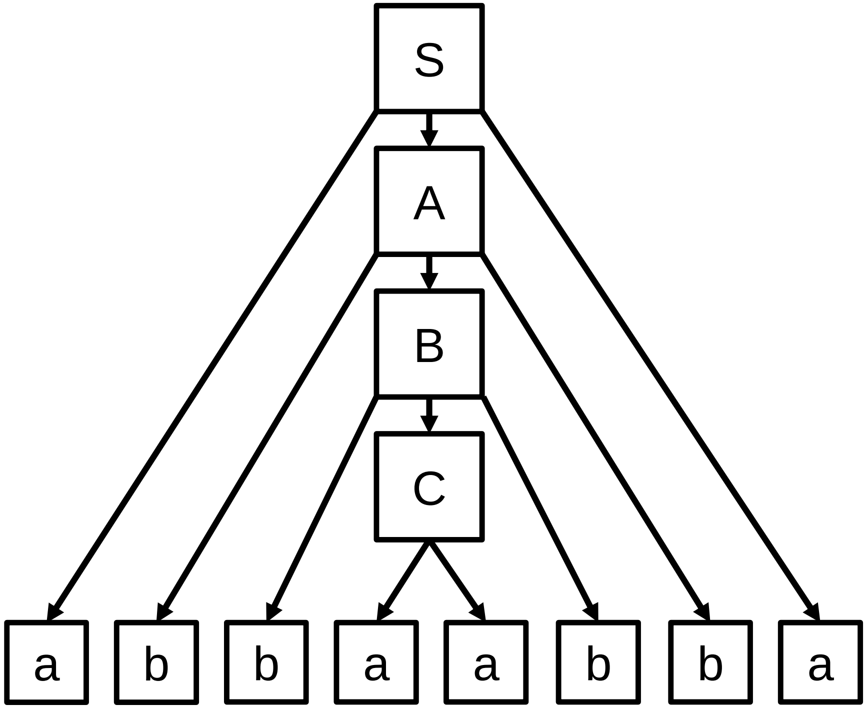
  C → aa | bb }

Einführungsaufgabe: Kontextfreie Sprachen

- 1) Geben sie den Ableitungsbaum für das Wort *abbaabba* und die Grammatik

$G = (\{S, A, B, C\}, \{a, b\}, P, S)$,

$P = \{$ S → aAa | bAb,
A → aBa | bBb,
B → aCa | bCb,
C → aa | bb }



Chomsky-Normalform

$G = (N, T, P, S)$ kontextfreie Grammatik.

G ist in **Chomsky-Normalform (CNF)** $:\Leftrightarrow\ P \subseteq N \times (NN \cup T)$ gilt,

d.h. nur Regeln der Form $A \rightarrow BC$ oder $A \rightarrow a$ mit $A, B, C \in N$, $a \in T$

Einfach gesagt sind folgende Regeln erlaubt:

1. Von Nonterminalsymbol zu 2 Nonterminalsymbolen
2. Von Nonterminalsymbol zu 1 Terminalsymbol

Von **Startsymbol** zu leerem Symbol (λ) - nur möglich wenn Startsymbol nicht auf der rechten Seite vorkommt.

Grammatiken in CNF haben folgende drei Eigenschaften

Chomsky-Normalform

G = (N, T, P, S) kontextfreie Grammatik.

G ist in **Chomsky-Normalform (CNF)** $:\Leftrightarrow$ $P \subseteq N \times (NN \cup T)$ gilt,

d.h. nur Regeln der Form $A \rightarrow BC$ oder $A \rightarrow a$ mit $A, B, C \in N$, $a \in T$

1. λ–frei

$G = (\{S, A, B\}, \{a, b\}, P, S)$,

$P = \{$ $S \rightarrow AB$,
$S \rightarrow \lambda$,
$A \rightarrow a$,
$A \rightarrow \lambda$,
$B \rightarrow b\}$

Wenn S bei keiner Regel auf der rechten Seite steht, ist der λ–Übergang $S \rightarrow \lambda$ erlaubt. Sonst könnte man mit CNF Grammatiken keine Sprachen abbilden, die das leere Wort enthalten.

Chomsky-Normalform

G = (N, T, P, S) kontextfreie Grammatik.

G ist in **Chomsky-Normalform (CNF)** :$\Leftrightarrow$ $P \subseteq N \times (NN \cup T)$ gilt,

d.h. nur Regeln der Form $A \rightarrow BC$ oder $A \rightarrow a$ mit $A, B, C \in N$, $a \in T$

2. Kettenregel-frei (keine reinen Umbenennungen)

$G = (\{S, A, B\}, \{a, b\}, P, S)$,

$P = \{$ $S \rightarrow AB$,

$S \rightarrow \lambda$,

$A \rightarrow B$,

$A \rightarrow a$,

$B \rightarrow b\}$

Chomsky-Normalform

G = (N, T, P, S) kontextfreie Grammatik.

G ist in **Chomsky-Normalform (CNF)** $:\Leftrightarrow$ $P \subseteq N \times (NN \cup T)$ gilt,

d.h. nur Regeln der Form $A \rightarrow BC$ oder $A \rightarrow a$ mit $A, B, C \in N, a \in T$

3. keine Mehrfachabzweigungen

G = ({*S*, *A*, *B*}, {*a*, *b*}, *P*, *S*),

P = { S → AB,

S → ABB,

A → a,

A → b,

B → b}

Chomsky-Normalform

G = (N, T, P, S) kontextfreie Grammatik.

G ist in **Chomsky-Normalform (CNF)** $:\Leftrightarrow$ $P \subseteq N \times (NN \cup T)$ gilt,

d.h. nur Regeln der Form $A \rightarrow BC$ oder $A \rightarrow a$ mit $A, B, C \in N$, $a \in T$

Welche der jeweiligen Mengen an Regeln gehören zu einer kontextfreien Grammatik in Chomsky-Normalform?

- ✖ $P_1 = \{ S \rightarrow a, A \rightarrow aaa \}$
- ✖ $P_2 = \{ S \rightarrow AB, AB \rightarrow a \}$
- ✖ $P_3 = \{ S \rightarrow b, B \rightarrow AaB \}$
- ✖ $P_4 = \{ S \rightarrow B, B \rightarrow A \}$
- ✖ $P_5 = \{ S \rightarrow b, B \rightarrow ABC \}$
- ✖ $P_6 = \{ S \rightarrow BCA, a \rightarrow B \}$
- ✖ $P_7 = \{ S \rightarrow \lambda, A \rightarrow \lambda \}$

Aufgabe 2: CNF Umwandlung (Präsenz)

- Gegeben sei eine kontextfreie Grammatik G.
- Um G in Chomsky-Normalform (CNF) zu überführen, wenden wir vier Schritte an.

- 1) Mache G Lambda–frei
- 2) Umbenennungen eliminieren
- 3) Alle Regeln so umformen, dass auf der rechten Seite entweder ein Terminal- oder mehrere Nonterminalsymbole stehen.
- 4) Alle Regeln, bei denen auf der rechten Seite mehr als zwei Nonterminalsymbole stehen, durch solche ersetzen, auf deren rechten Seite genau zwei Nonterminalsymbole stehen.

Aufgabe 2: CNF Umwandlung (Präsenz)

- Überführen Sie folgende kontextfreie Grammatik G in Chomsky-Normalform(CNF)

$$G = (\{S, A\}, \{a, b\}, P, S),$$

$$P = \{ \quad S \rightarrow A,$$

$$A \rightarrow aAb \mid ab \mid \lambda\}$$

Aufgabe 2: CNF Umwandlung (Präsenz)

$G = (\{S, A\}, \{a, b\}, P, S)$,
$P = \{ \quad S \rightarrow A,$
$A \rightarrow aAb \mid ab \mid \lambda\}$

- Schritt 1: Lambda-Übergänge eliminieren
- Um den Lambda-Übergang $A \rightarrow \lambda$ zu eliminieren, wird für jede Regel, bei der A auf der rechten Seite stand, eine neue eingeführt, bei der das A durch lambda ersetzt wird.
- $S \rightarrow A$ und $A \rightarrow aAb$ wird zu
- $S \rightarrow \lambda$ und $A \rightarrow a\lambda b$
- Analog für A

Aufgabe 2: CNF Umwandlung (Präsenz)

$G_1 = (\{S, A\}, \{a, b\}, P_1, S)$,
$P_1 = \{ \quad S \rightarrow A \mid \lambda,$
$\qquad A \rightarrow aAb \mid ab\}$

- Schritt 1: Lambda-Übergänge eliminieren
- Um den Lambda-Übergang $A \rightarrow \lambda$ zu eliminieren, wird für jede Regel, bei der A auf der rechten Seite stand, eine neue eingeführt, bei der das A durch lambda ersetzt wird.
- $S \rightarrow A$ und $A \rightarrow aAb$ wird zu
- $S \rightarrow \lambda$ und $A \rightarrow a\lambda b$
- Analog für A

Aufgabe 2: CNF Umwandlung (Präsenz)

$G_1 = (\{S, A\}, \{a, b\}, P_1, S)$,
$P_1 = \{ \quad S \rightarrow A \mid \lambda,$
$\quad A \rightarrow aAb \mid ab\}$

- Schritt 2: Umbenennungen eliminieren
- S → A ist eine reine Umbenennungen
- Um diese zu eliminieren führen wir neue Regeln ein
- S bildet nun zusätzlich auf alles ab, worauf A abgebildet hat
- $G_{(2)} = (\{S\}, T, P_{(2)}, S)$
- mit $P_{(2)} = \{S \rightarrow aAb \mid ab \mid \lambda, A \rightarrow aAb \mid ab\}$

Aufgabe 2: CNF Umwandlung (Präsenz)

$G_2 = (\{S, A\}, \{a, b\}, P_2, S)$,
$P_2 = \{ \quad S \rightarrow aAb \mid ab \mid \lambda,$
$\quad A \rightarrow aAb \mid ab\}$

- Schritt 2: Umbenennungen eliminieren
- $S \rightarrow A$ ist eine reine Umbenennungen
- Um diese zu eliminieren führen wir neue Regeln ein
- S bildet nun zusätzlich auf alles ab, worauf A abgebildet hat
- $G_{(2)} = (\{S\}, T, P_{(2)}, S)$
- mit $P_{(2)} = \{S \rightarrow aAb \mid ab \mid \lambda, A \rightarrow aAb \mid ab\}$

Aufgabe 2: CNF Umwandlung (Präsenz)

$G_2 = (\{S, A\}, \{a, b\}, P_2, S)$,
$P_2 = \{$ S → aAb | ab | λ,
A → aAb | ab$\}$

- Schritt 3: Alle Regeln so umformen, dass auf der rechten Seite entweder ein Terminal- oder mehrere Nonterminalsymbole stehen.

- S → aAb
- Wird zu
- S → BAC, B → a, C → b
- S → ab
- Wird zu
- S → BC, B → a, C → b

- Analog für A

Aufgabe 2: CNF Umwandlung (Präsenz)

$G_3 = (\{S, A, B, C\}, \{a, b\}, P_3, S)$,
$P_3 = \{$ S → BAC | BC | λ,
A → BAC | BC,
B → a,
C → b}

- Schritt 3: Alle Regeln so umformen, dass auf der rechten Seite entweder ein Terminal- oder mehrere Nonterminalsymbole stehen.

- S → aAb
- Wird zu
- S → BAC, B → a, C → b
- S → ab
- Wird zu
- S → BC, B → a, C → b

- Analog für A

Aufgabe 2: CNF Umwandlung (Präsenz)

$G_3 = (\{S, A, B, C\}, \{a, b\}, P_3, S)$,
$P_3 = \{$ S → BAC | BC | λ,
A → BAC | BC,
B → a,
C → b}

- Schritt 4: Alle Regeln, bei denen auf der rechten Seite mehr als zwei Nonterminalsymbole stehen, durch solche ersetzen, auf deren rechten Seite genau zwei Nonterminalsymbole stehen.

- S → BAC
- Wird zu
- S → BD, D → AC

- Analog für A

Aufgabe 2: CNF Umwandlung (Präsenz)

$G_4 = (\{S, A, B, C, D\}, \{a, b\}, P_4, S)$,

$P_4 = \{$ S → BD | BC | λ,
A → BD | BC,
D → AC,
B → a,
C → b$\}$

- Schritt 4: Alle Regeln, bei denen auf der rechten Seite mehr als zwei Nonterminalsymbole stehen, durch solche ersetzen, auf deren rechten Seite genau zwei Nonterminalsymbole stehen.

- S → BAC
- Wird zu
- S → BD, D → AC

- Analog für A

Aufgabe 2: CNF Umwandlung (Präsenz)

- Jetzt ist G in Chomsky-Normalform

$$G_{CNF} = (\{S, A, B, C, D\}, \{a, b\}, P_{CNF}, S),$$

$$\begin{aligned} P_{CNF} = \{S &\rightarrow BD \mid BC \mid \lambda, \\ A &\rightarrow BD \mid BC, \\ D &\rightarrow AC \\ B &\rightarrow a, \\ C &\rightarrow b\} \end{aligned}$$

Grammatiken

- Bisher haben wir mit Grammatiken nur Wörter erzeugt.
- Startwort S → AB → ... → Wort w

- Wenn man w gegeben hat:
  - Kann w mit den Regeln der Grammatik G aus S entstanden sein?
  - Wenn ja, wie kann man das zeigen?
  - Einen Ableitungsbaum für komplexe Grammatiken zu finden ist sehr mühsam.
  - Mit dem Cocke-Younger-Kasami Algorithmus geht es besser

- Vorteil von Grammatik gegenüber Automaten:
  - Nicht nur ja/nein $w \in L$?
  - Sondern auch wie ist w entstanden

Cocke-Younger-Kasami Algorithmus

- Wie leitet man das Wort *abbab* für die kontextfreie Grammatik G ab?

G = ({A, B, C, D, S}, {a, b}, P, S)

P = { S → AB | CD | AD,
A → a,
B → b,
C → AB | SC,
D → BC | AD | BB}

- **Wie geht man vor?**
  1. Kontextfreie Grammatik G in CNF bringen.
  2. Cocke-Younger-Kasami Algorithmus!

CYK Algorithmus

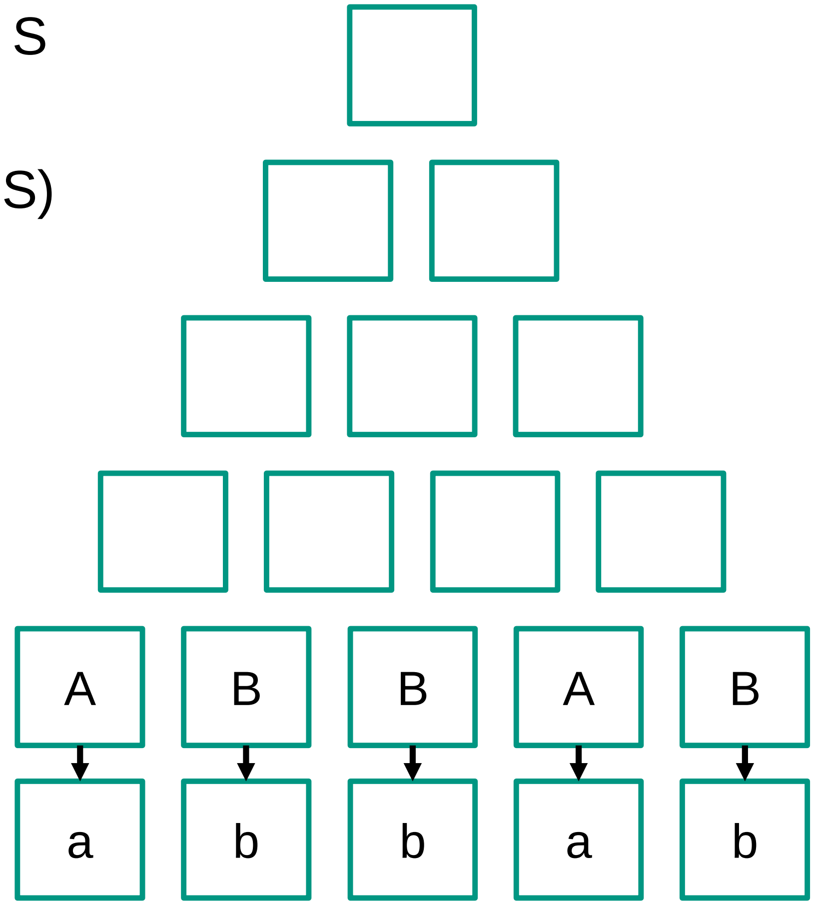
- Löst Wortproblem für kontextfreie Grammatiken
- Grammatik muss in Chomsky-Normalform sein
- Algorithmus am Beispiel erklärt
- Input: Wort w, Grammatik G, Startwort S

CYK Algorithmus

- Input: Wort w, Grammatik G, Startwort S

- Grammatik G=({S, A, B, C, D},{a, b},P,S)
- Produktionsfunktion P:
- P={ S → AB | CD | AD,
  C → AB | SC,
  D → BC | AD | BB,
  A → a,
  B → b }

- Wort w=abbab

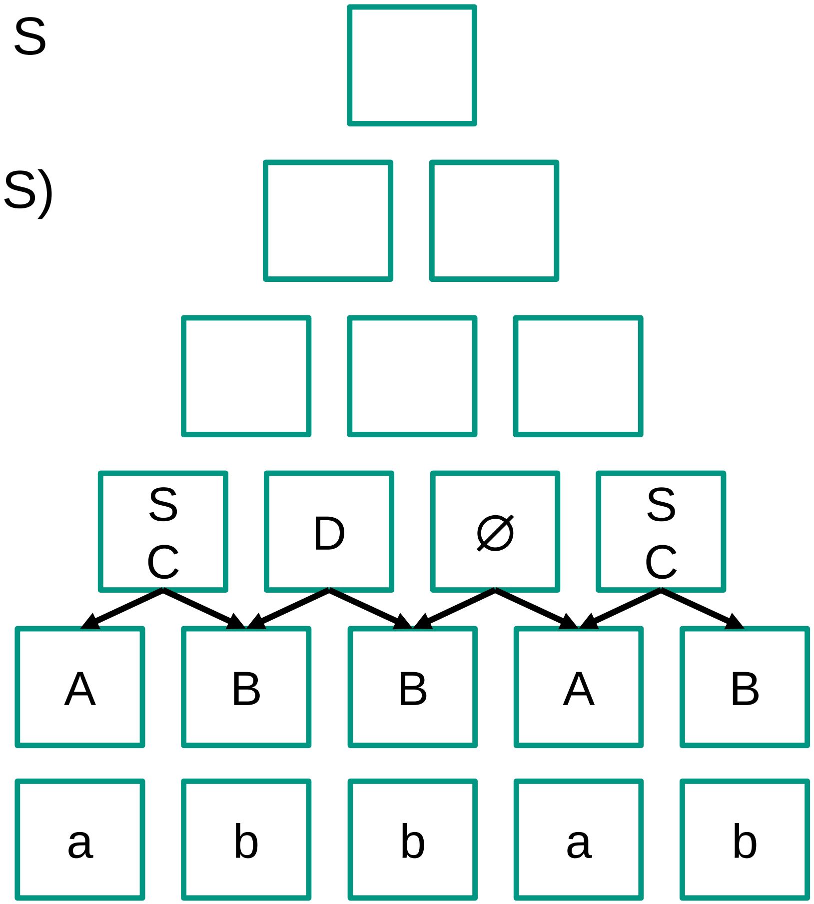


CYK Algorithmus

- Input: Wort w, Grammatik G, Startwort S

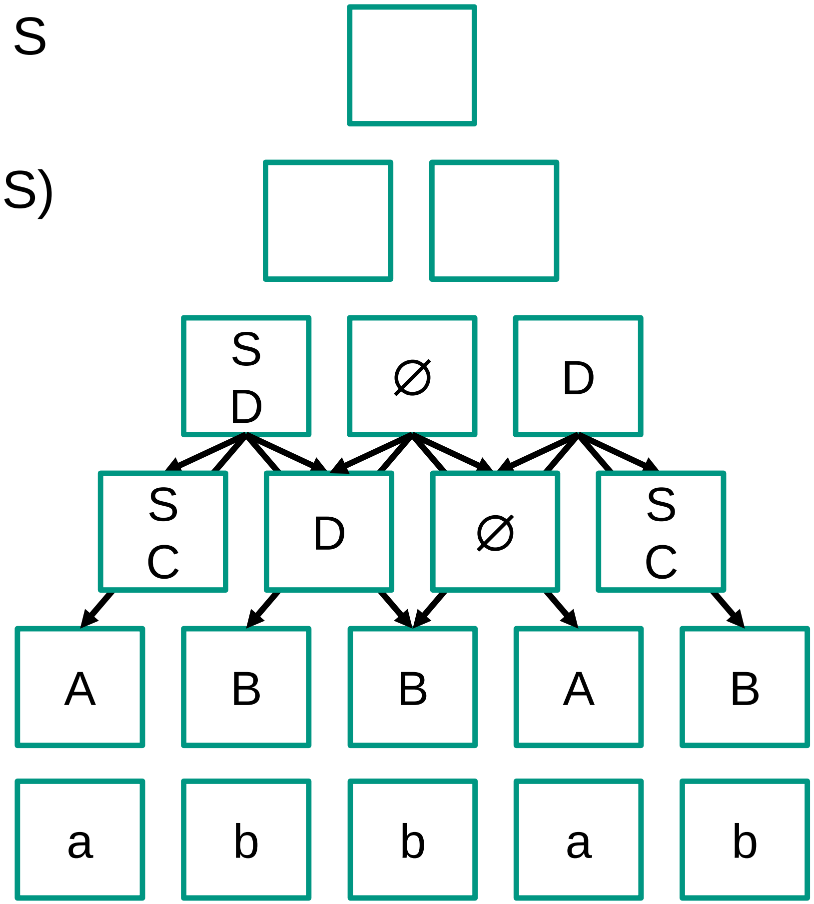
- Grammatik G=({S, A, B, C, D},{a, b},P,S)
- Produktionsfunktion P:
- P={ S → AB | CD | AD,
  C → AB | SC,
  D → BC | AD | BB,
  A → a,
  B → b }

- Wort w=abbab



CYK Algorithmus

- Input: Wort w, Grammatik G, Startwort S
- Grammatik G=({S, A, B, C, D},{a, b},P,S)
- Produktionsfunktion P:
- P={ S → AB | CD | AD,
  C → AB | SC,
  D → BC | AD | BB,
  A → a,
  B → b }
- Wort w=abbab

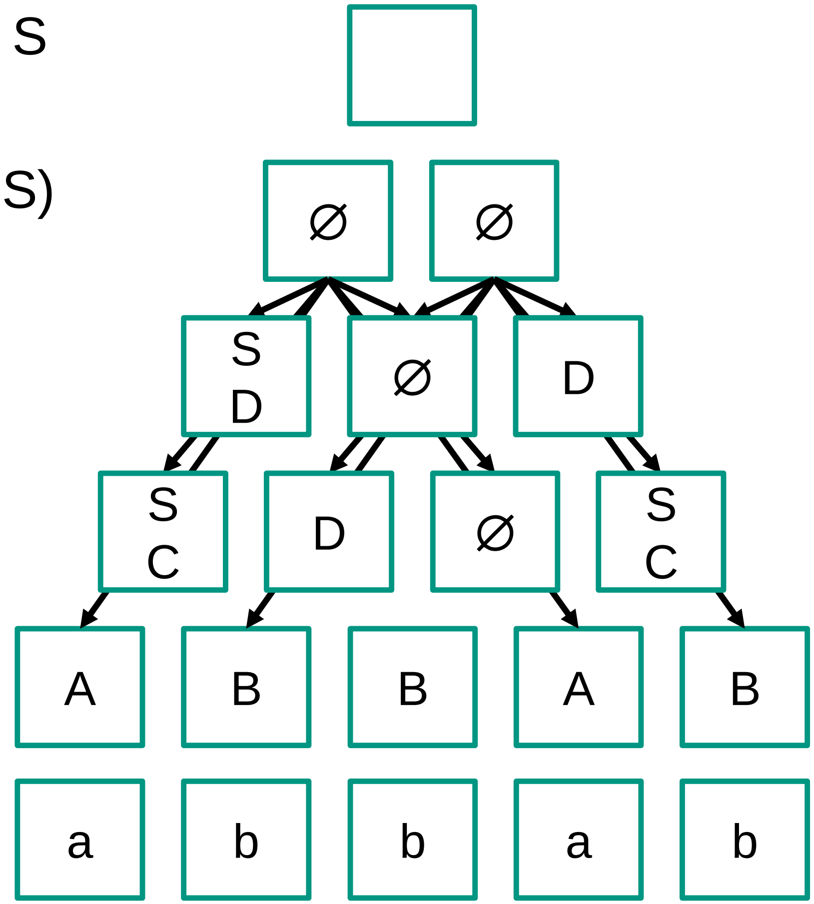


CYK Algorithmus

- Input: Wort w, Grammatik G, Startwort S

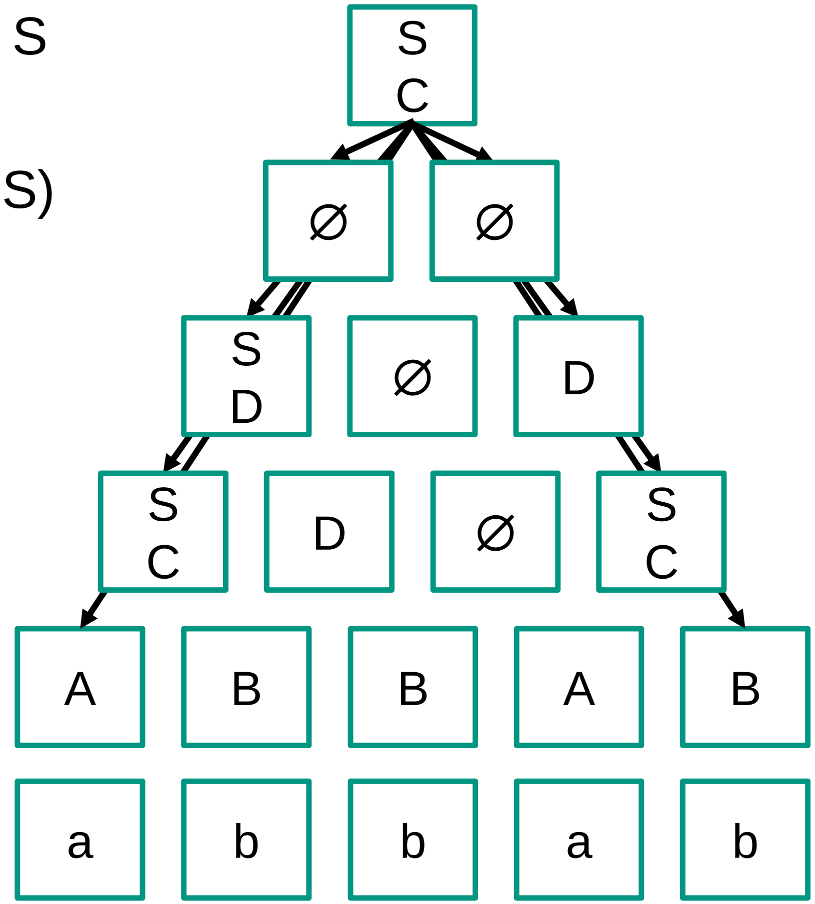
- Grammatik G=({S, A, B, C, D},{a, b},P,S)
- Produktionsfunktion P:
- P={ $S \rightarrow AB \mid CD \mid AD,$
  $C \rightarrow AB \mid SC,$
  $D \rightarrow BC \mid AD \mid BB,$
  $A \rightarrow a,$
  $B \rightarrow b$ }

- Wort w=abbab



CYK Algorithmus

- Input: Wort w, Grammatik G, Startwort S
- Grammatik G=({S, A, B, C, D},{a, b},P,S)
- Produktionsfunktion P:
- P={ S → AB | CD | AD,
  C → AB | SC,
  D → BC | AD | BB,
  A → a,
  B → b }
- Wort w=abbab



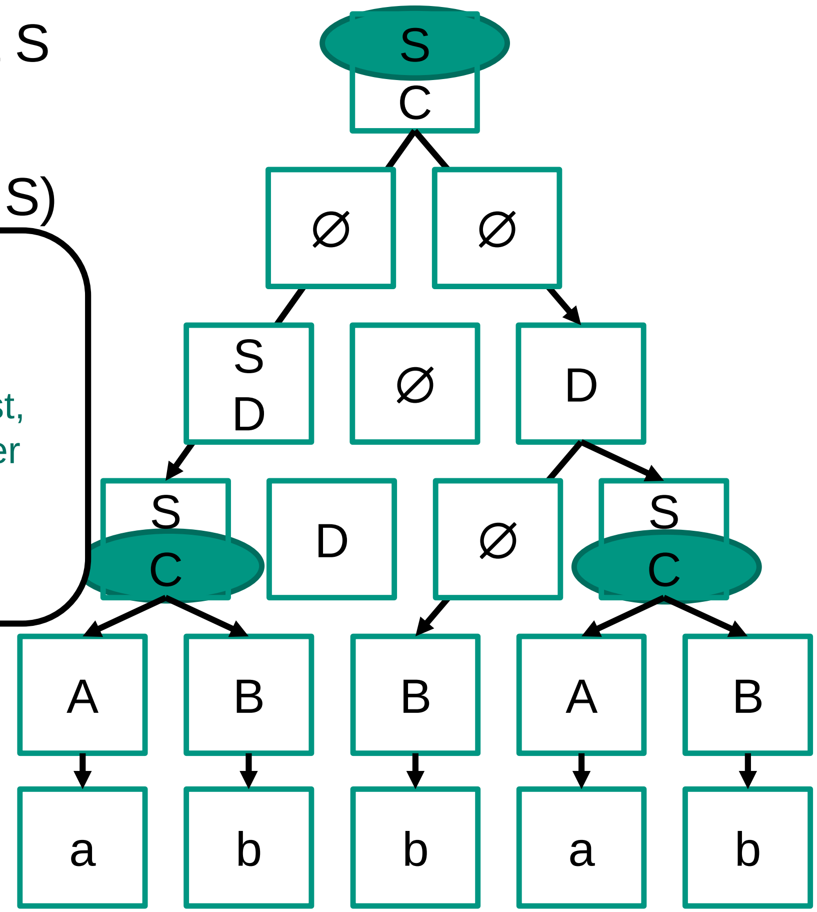
CYK Algorithmus

- Input: Wort w, Grammatik G, Startwort S

- Grammatik G=({S, A, B, C, D},{a, b},P,S)
- Produktionsfunktion
- P={ S → AB | CD |
  C → AB | SC,
  D → BC | AD |
  A → a,
  B → b }

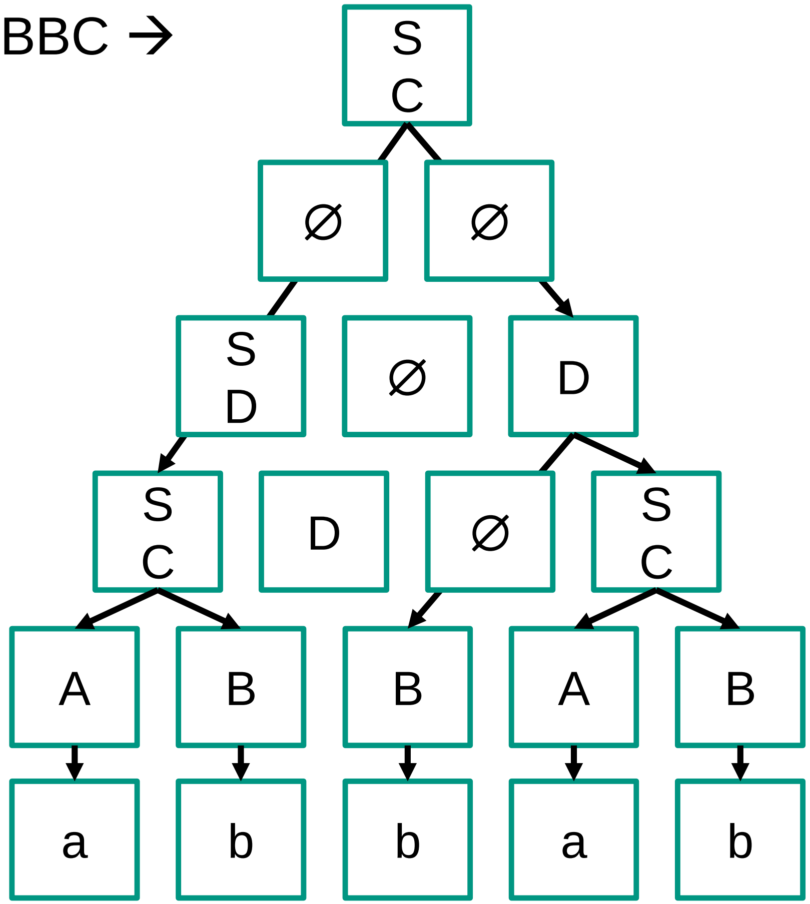
- Wort w=abbab

Da das Startsymbol dieser Grammatik S ist, kann die Ableitung hier nur mit S beginnen.



CYK Algorithmus

- Ableitung von w: S → CD → ABD → ABBC → ABBAB →aBBAB → … → abbab



Aufgabe 3: CYK Algorithmus (Präsenz)

- Gegeben sei die kontextfreie Grammatik G = (N, T, P, S) mit
- *N* = {S, A};
- *T* = {a, b};
- *P* = { S → A,

  A → aAb | ab}

- Überprüfen Sie mithilfe des Cocke-Younger-Kasami (CYK) Algorithmus, ob durch *G* das Wort *w* = *aaabbb* erzeugt werden kann.
- **Erster Schritt**: G in CNF bringen.

Aufgabe 3: CYK Algorithmus (Präsenz)

- **G in Chomsky Normalform**:
- $G_{CNF} = (N_{CNF}, T, P_{CNF}, S)$ mit
- $N_{CNF} = \{S, B, C, D\}$
- $T = \{a, b\}$
- $P_{CNF} = \{ \quad S \rightarrow BD \mid BC,$

  $D \rightarrow SC,$

  $B \rightarrow a,$

  $C \rightarrow b\}$

Aufgabe 3: CYK Algorithmus (Präsenz)

Karlsruhe Institute of Technology

- **G in Chomsky Normalform**:
- $G_{CNF} = (N_{CNF}, T, P_{CNF}, S)$ mit
- $N_{CNF} = \{S, B, C, D\}$
- $T = \{a, b\}$
- $P_{CNF} = \{ S \rightarrow BD \mid BC, D \rightarrow SC, B \rightarrow a, C \rightarrow b\}$

Visualisierung hier gespiegelt zur 'Pyramide' aus der Vorlesung

	a	a	a	b	b	b
$m = 1$						
$m = 2$						
$m = 3$						
$m = 4$						
$m = 5$						
$m = 6$						

Aufgabe 3: CYK Algorithmus (Präsenz)

- **G in Chomsky Normalform**:
- $G_{CNF} = (N_{CNF}, T, P_{CNF}, S)$ mit
- $N_{CNF} = \{S, B, C, D\}$
- $T = \{a, b\}$
- $P_{CNF} = \{ S \rightarrow BD \mid BC, D \rightarrow SC, B \rightarrow a, C \rightarrow b\}$

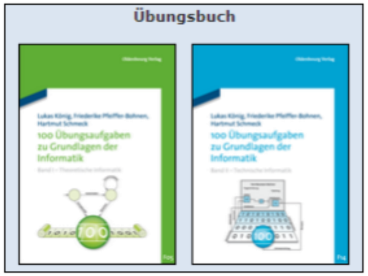
	a	a	a	b	b	b
$m = 1$	B	B	B	C	C	C
$m = 2$	$-$	$-$	S	$-$	$-$	
$m = 3$	$-$	$-$	D	$-$		
$m = 4$	$-$	S	$-$			
$m = 5$	$-$	D				
$m = 6$	S					

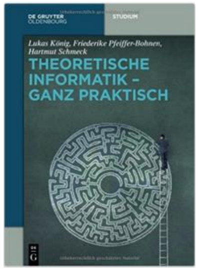
Auch klausur-relevant…

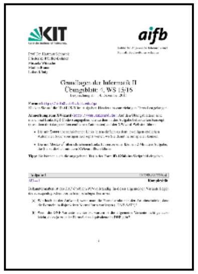
Weitere Aufgaben zu den Themen dieses Tutoriums

- Aus dem **Aufgabenpool** bzw. **Übungsbuch**:
  - Kapitel 6: Kontextfreie Grammatiken (KON-AB, KON-AC, KON-AD).

- Bei Fragen oder Kommentaren zu **allen Aufgaben**
  - nutzen Sie das **Q/A-Forum** oder
  - fragen Sie Ihren Tutor / Ihre Tutorin.







Fragen zu den letzten Heimaufgaben

?

Während die Tutoriumsaufgaben einen Einstieg in ein Thema ermöglichen, dienen die Heimaufgaben dem tieferen Verständnis und insbesondere auch der Klausurvorbereitung

Aufgabe 4: Kontextfreie Sprachen (Heim)

- Gesucht ist eine kontextfreie Grammatik *G*, für die gilt:

  $L(G) = L$ mit

  $L = \{a^n b^n c^m \mid m, n \geq 0\}$.

Geben Sie die Grammatik G = (N, T, P, S) vollständig an.

Lösung:
G = (N, T, P, S)
N = { S, A, B }
T = { a, b, c }
$P = \{ S \rightarrow AB,\ A \rightarrow aAb \mid \lambda,\ B \rightarrow cB \mid \lambda \}$

Aufgabe 4: Kontextfreie Sprachen (Heim)

- Gesucht ist eine kontextfreie Grammatik *G*, für die gilt:

  L(G) = L mit

  $L = \{a^n b^n c^m \mid m, n \geq 0\}$.

  Geben Sie die Grammatik G = (N, T, P, S) vollständig an.

  Lösung:
  G = (N, T, P, S)
  N = { S, A, B }
  T = { a, b, c }
  $P = \{ S \rightarrow AB,\ A \rightarrow aAb \mid \lambda,\ B \rightarrow cB \mid \lambda \}$

Aufgabe 4: Kontextfreie Sprachen (Heim)

- Geben Sie für folgende Grammatik die Produktion des Testwortes *aaaaabbbbbcc*

- G = (N, T, P, S)
- N = { S, A, B }
- T = { a, b, c }
- P = { S → AB, A → aAb | λ, B → cB | λ }

Lösung:
S ➔ AB ➔ aAbB ➔ aaAbbB ➔ aaaAbbbB ➔ aaaaAbbbbB ➔ aaaaaAbbbbbB ➔ aaaaabbbbbB ➔ aaaaabbbbbcB ➔ aaaaabbbbbccB ➔ aaaaabbbbbcc