

6b

Der CYK-Algorithmus

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 4. April 2024

Basiert auf Folien von PD Dr. David Sabel



Definition

Eine Sprache L ist **entscheidbar**, wenn es einen Algorithmus gibt, der bei Eingabe eines Wortes w in endlicher Zeit feststellt, ob $w \in L$ gilt oder nicht.

Wortproblem für Typ 2-Grammatiken

Definition

Das **Wortproblem** für Typ i -Grammatiken ist die Frage, ob für eine gegebene Typ i -Grammatik $G = (V, \Sigma, P, S)$ und ein Wort $w \in \Sigma^*$ $w \in L(G)$ gilt oder nicht.

Wortproblem für Typ 2-Grammatiken

Definition

Das **Wortproblem** für Typ i -Grammatiken ist die Frage, ob für eine gegebene Typ i -Grammatik $G = (V, \Sigma, P, S)$ und ein Wort $w \in \Sigma^*$ $w \in L(G)$ gilt oder nicht.

Satz

Das Wortproblem für Typ 2-Grammatiken ist entscheidbar:
Es gibt einen Algorithmus, der bei Eingabe von Typ 2-Grammatik G und Wort w nach endlicher Zeit entscheidet, ob $w \in L(G)$ gilt oder nicht. Zudem entscheidet er das Wortproblem in Polynomialzeit.

Grundgedanke des Algorithmus

Der **CYK-Algorithmus** von Cocke, Younger und Kasami ist ein Polynomialzeitalgorithmus für das Wortproblem für Typ 2-Grammatiken.

Grundgedanke des Algorithmus

Der **CYK-Algorithmus** von Cocke, Younger und Kasami ist ein Polynomialzeitalgorithmus für das Wortproblem für Typ 2-Grammatiken.

Eingabe: Eine CFG $G = (V, \Sigma, P, S)$ in Chomsky-Normalform und ein Wort $w \in \Sigma^+$.

Grundgedanke des Algorithmus

Der **CYK-Algorithmus** von Cocke, Younger und Kasami ist ein Polynomialzeitalgorithmus für das Wortproblem für Typ 2-Grammatiken.

Eingabe: Eine CFG $G = (V, \Sigma, P, S)$ in Chomsky-Normalform und ein Wort $w \in \Sigma^+$.

G ist in **Chomsky-Normalform**, wenn für jede Produktion $A \rightarrow w \in P$ gilt:
 $w = a \in \Sigma$ oder $w = BC$ mit $B, C \in V$.

Für jede CFG G mit $\varepsilon \notin L(G)$ kann eine Chomsky-Normalform G' mit $L(G') = L(G)$ berechnet werden.

Grundgedanke des Algorithmus

Der **CYK-Algorithmus** von Cocke, Younger und Kasami ist ein Polynomialzeitalgorithmus für das Wortproblem für Typ 2-Grammatiken.

Eingabe: Eine CFG $G = (V, \Sigma, P, S)$ in Chomsky-Normalform und ein Wort $w \in \Sigma^+$.

G ist in **Chomsky-Normalform**, wenn für jede Produktion $A \rightarrow w \in P$ gilt:
 $w = a \in \Sigma$ oder $w = BC$ mit $B, C \in V$.

Für jede CFG G mit $\varepsilon \notin L(G)$ kann eine Chomsky-Normalform G' mit $L(G') = L(G)$ berechnet werden.

Ausgabe: **ja**, wenn $w \in L(G)$, sonst **nein**.

Grundgedanke des Algorithmus

Der **CYK-Algorithmus** von Cocke, Younger und Kasami ist ein Polynomialzeitalgorithmus für das Wortproblem für Typ 2-Grammatiken.

Eingabe: Eine CFG $G = (V, \Sigma, P, S)$ in Chomsky-Normalform und ein Wort $w \in \Sigma^+$.

G ist in **Chomsky-Normalform**, wenn für jede Produktion $A \rightarrow w \in P$ gilt:
 $w = a \in \Sigma$ oder $w = BC$ mit $B, C \in V$.

Für jede CFG G mit $\varepsilon \notin L(G)$ kann eine Chomsky-Normalform G' mit $L(G') = L(G)$ berechnet werden.

Ausgabe: **ja**, wenn $w \in L(G)$, sonst **nein**.

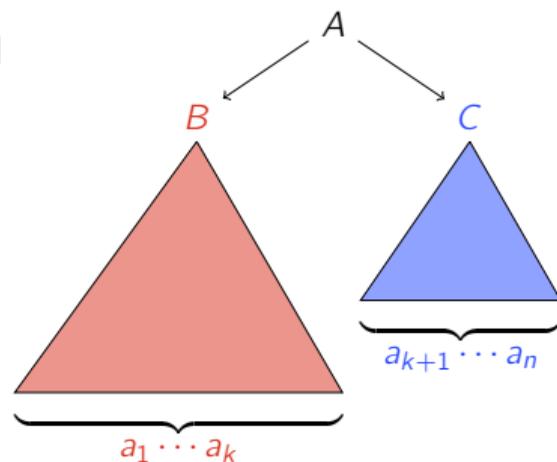
Grober Ansatz:

1. Teste für jede Variable A und Teilwort u von w , ob sie es erzeugt.
2. Verwende Test zum Prüfen, ob das Startsymbol S das Wort w erzeugt.

Grundgedanke des CYK-Algorithmus

Prüfe, ob $A \in V$ ein Wort $u = a_1 \cdots a_n$ ($n \geq 1$) erzeugt:

- ▶ Wenn $u = a_1 \in \Sigma$, dann prüfe ob $A \rightarrow a_1 \in P$.
- ▶ Anderenfalls ($n > 1$) kann u nur erzeugt werden, wenn
 - ▶ es eine Produktion $A \rightarrow BC \in P$ gibt und
 - ▶ es einen Index $1 \leq k < n$ gibt, sodass
 - B erzeugt $a_1 \cdots a_k$ und
 - C erzeugt $a_{k+1} \cdots a_n$.



- ▶ Daher prüfe für **alle** $A \rightarrow BC \in P$ und **alle** k mit $1 \leq k < n$ **rekursiv**, ob B das Wort $a_1 \cdots a_k$ und C das Wort $a_{k+1} \cdots a_n$ erzeugt.

Beispiel für naives rekursives Suchen

Seien die CFG $G = (V, \Sigma, P, S)$ mit

$$P = \{S \rightarrow AB \mid BA, \\ A \rightarrow AA \mid AB \mid a, \\ B \rightarrow BB \mid b\}$$

und das Wort $bbbaab$.

Beispiel für naives rekursives Suchen

Seien die CFG $G = (V, \Sigma, P, S)$ mit S erzeugt *bbbaab*, denn $S \rightarrow BA \in P$ und

$$P = \{S \rightarrow AB \mid BA, \\ A \rightarrow AA \mid AB \mid a, \\ B \rightarrow BB \mid b\}$$

und das Wort *bbbaab*.

- ▶ B erzeugt *bbb*, denn $B \rightarrow BB \in P$ und
 - ▶ B erzeugt *bb*, denn $B \rightarrow BB \in P$ und
 - ▶ B erzeugt *b*, denn $B \rightarrow b \in P$
 - ▶ B erzeugt *b*, denn $B \rightarrow b \in P$
 - ▶ B erzeugt *b*, denn $B \rightarrow b \in P$
- ▶ A erzeugt *aab*, denn $A \rightarrow AB \in P$ und
 - ▶ A erzeugt *aa*, denn $A \rightarrow AA \in P$ und
 - ▶ A erzeugt *a*, denn $A \rightarrow a \in P$ und
 - ▶ A erzeugt *a*, denn $A \rightarrow a \in P$
 - ▶ B erzeugt *b*, denn $B \rightarrow b \in P$.

Beispiel für naives rekursives Suchen

Seien die CFG $G = (V, \Sigma, P, S)$ mit S erzeugt $bbbaab$, denn $S \rightarrow BA \in P$ und

$$P = \{S \rightarrow AB \mid BA, \\ A \rightarrow AA \mid AB \mid a, \\ B \rightarrow BB \mid b\}$$

und das Wort $bbbaab$.

- ▶ B erzeugt bbb , denn $B \rightarrow BB \in P$ und
 - ▶ B erzeugt bb , denn $B \rightarrow BB \in P$ und
 - ▶ B erzeugt b , denn $B \rightarrow b \in P$
 - ▶ B erzeugt b , denn $B \rightarrow b \in P$
 - ▶ B erzeugt b , denn $B \rightarrow b \in P$
- ▶ A erzeugt aab , denn $A \rightarrow AB \in P$ und
 - ▶ A erzeugt aa , denn $A \rightarrow AA \in P$ und
 - ▶ A erzeugt a , denn $A \rightarrow a \in P$ und
 - ▶ A erzeugt a , denn $A \rightarrow a \in P$
 - ▶ B erzeugt b , denn $B \rightarrow b \in P$.

Beispiel für naives rekursives Suchen

Seien die CFG $G = (V, \Sigma, P, S)$ mit S erzeugt $bbbaab$, denn $S \rightarrow BA \in P$ und

$$P = \{S \rightarrow AB \mid BA, \\ A \rightarrow AA \mid AB \mid a, \\ B \rightarrow BB \mid b\}$$

und das Wort $bbbaab$.

- ▶ B erzeugt bbb , denn $B \rightarrow BB \in P$ und
 - ▶ B erzeugt bb , denn $B \rightarrow BB \in P$ und
 - ▶ B erzeugt b , denn $B \rightarrow b \in P$
 - ▶ B erzeugt b , denn $B \rightarrow b \in P$
 - ▶ B erzeugt b , denn $B \rightarrow b \in P$
- ▶ A erzeugt aab , denn $A \rightarrow AB \in P$ und
 - ▶ A erzeugt aa , denn $A \rightarrow AA \in P$ und
 - ▶ A erzeugt a , denn $A \rightarrow a \in P$ und
 - ▶ A erzeugt a , denn $A \rightarrow a \in P$
 - ▶ B erzeugt b , denn $B \rightarrow b \in P$.

Um an Effizienz zu gewinnen:

Statt Rekursion verwende [dynamische Programmierung](#).

Grundgedanke des CYK-Algorithmus

Der Algorithmus berechnet Mengen $V(i, j) \subseteq V$, sodass

$$V(i, j) := \{A \in V \mid A \Rightarrow^* a_i \cdots a_{i+j-1}\}$$

Informell: $V(i, j)$ enthält alle Variablen $A \in V$, die $a_i \cdots a_{i+j-1}$ (= das Teilwort von w ab Position i mit Länge j) erzeugen.

Grundgedanke des CYK-Algorithmus

Der Algorithmus berechnet Mengen $V(i, j) \subseteq V$, sodass

$$V(i, j) := \{A \in V \mid A \Rightarrow^* a_i \cdots a_{i+j-1}\}$$

Informell: $V(i, j)$ enthält alle Variablen $A \in V$, die $a_i \cdots a_{i+j-1}$ (= das Teilwort von w ab Position i mit Länge j) erzeugen.

Schritte:

1. Beginne mit $V(i, 1) = \{A \mid A \rightarrow a_i \in P\}$.

Grundgedanke des CYK-Algorithmus

Der Algorithmus berechnet Mengen $V(i, j) \subseteq V$, sodass

$$V(i, j) := \{A \in V \mid A \Rightarrow^* a_i \cdots a_{i+j-1}\}$$

Informell: $V(i, j)$ enthält alle Variablen $A \in V$, die $a_i \cdots a_{i+j-1}$ (= das Teilwort von w ab Position i mit Länge j) erzeugen.

Schritte:

1. Beginne mit $V(i, 1) = \{A \mid A \rightarrow a_i \in P\}$.
2. Berechne $V(i, j)$ für $j = 2$ bis n . Für $j > 1$ gilt:

$$A \in V(i, j) \text{ g.d.w. es gibt } k \in \{1, 2, \dots, j-1\}, \text{ sodass} \\ A \rightarrow BC \in P, B \in V(i, k) \text{ und } C \in V(i+k, j-k)$$

Grundgedanke des CYK-Algorithmus

Der Algorithmus berechnet Mengen $V(i, j) \subseteq V$, sodass

$$V(i, j) := \{A \in V \mid A \Rightarrow^* a_i \cdots a_{i+j-1}\}$$

Informell: $V(i, j)$ enthält alle Variablen $A \in V$, die $a_i \cdots a_{i+j-1}$ (= das Teilwort von w ab Position i mit Länge j) erzeugen.

Schritte:

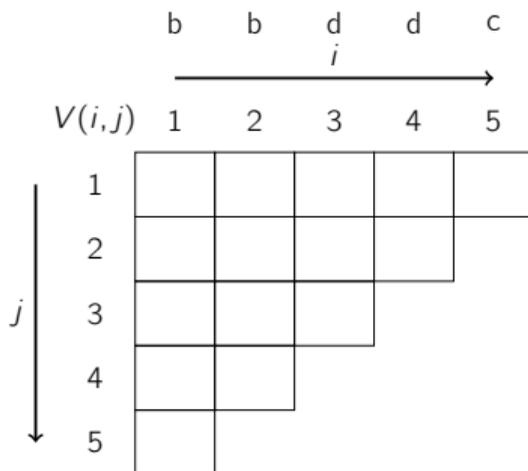
1. Beginne mit $V(i, 1) = \{A \mid A \rightarrow a_i \in P\}$.
2. Berechne $V(i, j)$ für $j = 2$ bis n . Für $j > 1$ gilt:

$$A \in V(i, j) \text{ g.d.w. es gibt } k \in \{1, 2, \dots, j-1\}, \text{ sodass} \\ A \rightarrow BC \in P, B \in V(i, k) \text{ und } C \in V(i+k, j-k)$$

3. Prüfe, ob $S \in V(1, n)$.

Beispiel für den CYK-Algorithmus

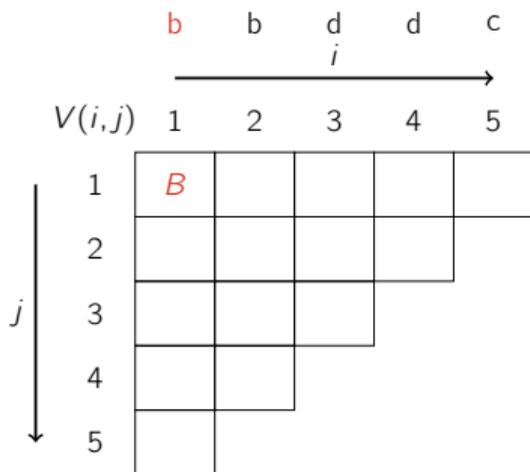
Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$



Füllen der $V(i, 1)$ -Einträge

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$



Füllen der $V(i, 1)$ -Einträge

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
				i		
		→				
$V(i,j)$		1	2	3	4	5
	1	B	B			
	2					
	3					
	4					
	5					

j ↓

Füllen der $V(i, 1)$ -Einträge

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
				i		
		→				
$V(i,j)$		1	2	3	4	5
	1	B	B	D		
	2					
	3					
	4					
	5					

j ↓

Füllen der $V(i, 1)$ -Einträge

Beispiel für den CYK-Algorithmus

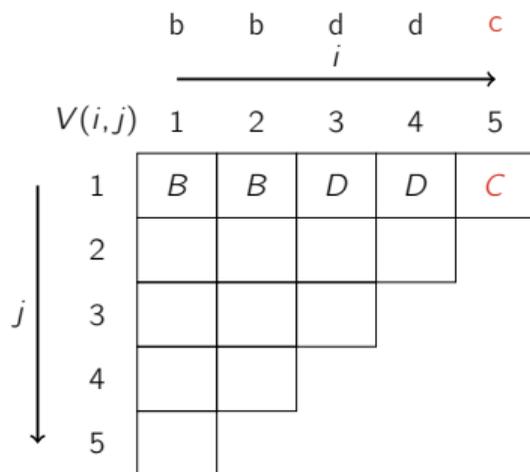
Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		\xrightarrow{i}				
	$V(i,j)$	1	2	3	4	5
$\downarrow j$	1	B	B	D	D	
	2					
	3					
	4					
	5					

Füllen der $V(i, 1)$ -Einträge

Beispiel für den CYK-Algorithmus

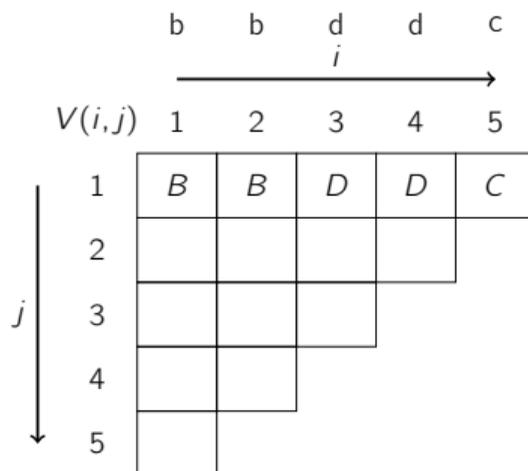
Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$



Füllen der $V(i, 1)$ -Einträge

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$



3-fache Schleife:

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2						
3						
4						
5						

3-fache Schleife: $j = 2, i = 1, k = 1: V(1, 2) := V(1, 2) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 1), \\ C \in V(2, 1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3						
4						
5						

3-fache Schleife: $j = 2, i = 2, k = 1: V(2, 2) := V(2, 2) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(2, 1), \\ C \in V(3, 1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3						
4						
5						

3-fache Schleife: $j = 2, i = 3, k = 1: V(3, 2) := V(3, 2) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(3, 1), \\ C \in V(4, 1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A		D		
3						
4						
5						

3-fache Schleife: $j = 2, i = 4, k = 1: V(4, 2) := V(4, 2) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(4, 1), \\ C \in V(5, 1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3						
4						
5						

3-fache Schleife: $j = 3, i = 1, k = 1: V(1, 3) := V(1, 3) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 1), \\ C \in V(2, 2) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

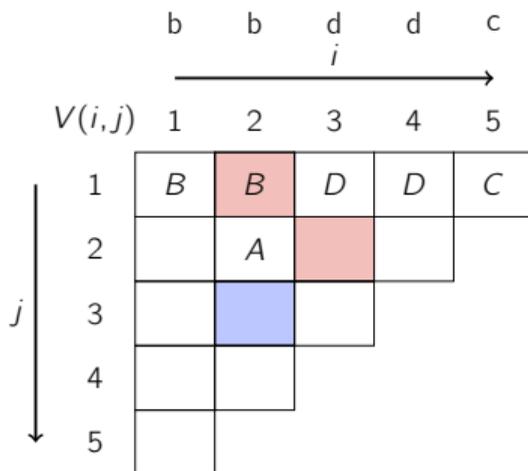
Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		\xrightarrow{i}				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3						
4						
5						

3-fache Schleife: $j = 3, i = 1, k = 2: V(1, 3) := V(1, 3) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 2), \\ C \in V(3, 1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$



3-fache Schleife: $j = 3, i = 2, k = 1: V(2, 3) := V(2, 3) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(2, 1), \\ C \in V(3, 2) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3		E				
4						
5						

3-fache Schleife: $j = 3, i = 2, k = 2: V(2, 3) := V(2, 3) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(2, 2), \\ C \in V(4, 1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3		E				
4						
5						

3-fache Schleife: $j = 3, i = 3, k = 1: V(3, 3) := V(3, 3) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(3, 1), \\ C \in V(4, 2) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3		E				
4						
5						

3-fache Schleife: $j = 3, i = 3, k = 2: V(3, 3) := V(3, 3) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(3, 2), \\ C \in V(5, 1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3		E				
4	A					
5						

3-fache Schleife: $j = 4, i = 1, k = 1: V(1, 4) := V(1, 4) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 1), \\ C \in V(2, 3) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3		E				
4	A					
5						

$j \downarrow$

3-fache Schleife: $j = 4, i = 1, k = 2: V(1, 4) := V(1, 4) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 2), \\ C \in V(3, 2) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3		E				
4	A					
5						

3-fache Schleife: $j = 4, i = 1, k = 3: V(1, 4) := V(1, 4) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 3), \\ C \in V(4, 1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		\xrightarrow{i}				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3		E				
4	A					
5						

3-fache Schleife: $j = 4, i = 2, k = 1: V(2, 4) := V(2, 4) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(2, 1), \\ C \in V(3, 3) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3		E				
4	A					
5						

3-fache Schleife: $j = 4, i = 2, k = 2: V(2, 4) := V(2, 4) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(2, 2), \\ C \in V(4, 2) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3		E				
4	A					
5						

3-fache Schleife: $j = 4, i = 2, k = 3: V(2, 4) := V(2, 4) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(2, 3), \\ C \in V(5, 1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3		E				
4	A					
5						

3-fache Schleife: $j = 5, i = 1, k = 1: V(1, 5) := V(1, 5) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 1), \\ C \in V(2, 4) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3		E				
4	A					
5						

3-fache Schleife: $j = 5, i = 1, k = 2: V(1, 5) := V(1, 5) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 2), \\ C \in V(3, 3) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3		E				
4	A					
5						

3-fache Schleife: $j = 5, i = 1, k = 3: V(1, 5) := V(1, 5) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 3), \\ C \in V(4, 2) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	B	B	D	D	C	
2		A				
3		E				
4	A					
5	S					

3-fache Schleife: $j = 5, i = 1, k = 4: V(1, 5) := V(1, 5) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 4), \\ C \in V(5, 1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = b b d d c$ und $G = (\{S, A, B, C, D, E\}, \{b, c, d\}, P, S)$ mit
 $P = \{S \rightarrow AC, A \rightarrow BE, A \rightarrow BD, E \rightarrow AD, C \rightarrow c, B \rightarrow b, D \rightarrow d\}$

		b	b	d	d	c
		\xrightarrow{i}				
	$V(i,j)$	1	2	3	4	5
$\downarrow j$	1	B	B	D	D	C
	2		A			
	3		E			
	4	A				
	5	S				

Da $S \in V(1, 5)$, gilt $w \in L(G)$.

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	↓	T	U	T		
2						
3						
4						
5						

Füllen der $V(i, 1)$ -Einträge

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

a b a b a
 i
 →

$V(i,j)$ 1 2 3 4 5

1	T	U	T	U	
2					
3					
4					
5					

j
↓

Füllen der $V(i, 1)$ -Einträge

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		\xrightarrow{i}				
$V(i,j)$		1	2	3	4	5
$\downarrow j$	1	T	U	T	U	T
	2					
	3					
	4					
	5					

Füllen der $V(i, 1)$ -Einträge

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		\xrightarrow{i}				
$V(i,j)$		1	2	3	4	5
$\downarrow j$	1	T	U	T	U	T
	2					
	3					
	4					
	5					

3-fache Schleife:

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit

$P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	T	U	T	U	T	
2	S,T					
3						
4						
5						

3-fache Schleife: $j = 2, i = 1, k = 1: V(1, 2) := V(1, 2) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 1), \\ C \in V(2, 1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit

$P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	T	U	T	U	T	
2	S, T	S	S, T			
3						
4						
5						

3-fache Schleife: $j = 2, i = 3, k = 1: V(3, 2) := V(3, 2) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(3, 1), \\ C \in V(4, 1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	T	U	T	U	T	
2	S, T	S	S, T	S		
3						
4						
5						

3-fache Schleife: $j = 2, i = 4, k = 1: V(4, 2) := V(4, 2) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(4, 1), \\ C \in V(5, 1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	T	U	T	U	T	
2	S, T	S	S, T	S		
3						
4						
5						

3-fache Schleife: $j = 3, i = 1, k = 1: V(1, 3) := V(1, 3) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 1), \\ C \in V(2, 2) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	T	U	T	U	T	
2	S, T	S	S, T	S		
3	T	S				
4						
5						

3-fache Schleife: $j = 3, i = 2, k = 1: V(2, 3) := V(2, 3) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(2, 1), \\ C \in V(3, 2) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	T	U	T	U	T	
2	S,T	S	S,T	S		
3	T	S	T			
4						
5						

3-fache Schleife: $j = 4, i = 1, k = 1: V(1, 4) := V(1, 4) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 1), \\ C \in V(2, 3) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	T	U	T	U	T	
2	S, T	S	S, T	S		
3	T	S	T			
4	T					
5						

3-fache Schleife: $j = 4, i = 1, k = 2: V(1, 4) := V(1, 4) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 2), \\ C \in V(3, 2) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit

$P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	T	U	T	U	T	
2	S, T	S	S, T	S		
3	T	S	T			
4	S, T					
5						

3-fache Schleife: $j = 4, i = 1, k = 3: V(1, 4) := V(1, 4) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 3), \\ C \in V(4, 1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	T	U	T	U	T	
2	S, T	S	S, T	S		
3	T	S	T			
4	S, T	S				
5						

3-fache Schleife: $j = 4, i = 2, k = 1: V(2, 4) := V(2, 4) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(2, 1), \\ C \in V(3, 3) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	T	U	T	U	T	
2	S, T	S	S, T	S		
3	T	S	T			
4	S, T	S				
5						

3-fache Schleife: $j = 4, i = 2, k = 2: V(2, 4) := V(2, 4) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(2, 2), \\ C \in V(4, 2) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$\xrightarrow{\quad i \quad}$				
$V(i, j)$		1	2	3	4	5
1	T	U	T	U	T	
2	S, T	S	S, T	S		
3	T	S	T			
4	S, T	S				
5						

3-fache Schleife: $j = 4, i = 2, k = 3: V(2, 4) := V(2, 4) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(2, 3), \\ C \in V(5, 1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$\xrightarrow{\quad i \quad}$				
$V(i,j)$		1	2	3	4	5
1	T	U	T	U	T	
2	S,T	S	S,T	S		
3	T	S	T			
4	S,T	S				
5						

3-fache Schleife: $j = 5, i = 1, k = 1: V(1, 5) := V(1, 5) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 1), \\ C \in V(2, 4) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit
 $P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

		a	b	a	b	a
		$i \rightarrow$				
$V(i,j)$		1	2	3	4	5
1	T	U	T	U	T	
2	S,T	S	S,T	S		
3	T	S	T			
4	S,T	S				
5	T					

3-fache Schleife: $j = 5, i = 1, k = 3: V(1, 5) := V(1, 5) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 3), \\ C \in V(4, 2) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

Seien $w = ababa$ und $G = (\{S, T, U\}, \{a, b, c\}, P, S)$ mit

$P = \{S \rightarrow TU, S \rightarrow UT, T \rightarrow TT, T \rightarrow TU, T \rightarrow a, U \rightarrow UU, U \rightarrow b\}$

$a \quad b \quad a \quad b \quad a$
 $\xrightarrow{\quad i \quad}$

$V(i,j)$	1	2	3	4	5
1	T	U	T	U	T
2	S, T	S	S, T	S	
3	T	S	T		
4	S, T	S			
5	T				

$\downarrow j$

3-fache Schleife: $j = 5, i = 1, k = 4: V(1, 5) := V(1, 5) \cup \left\{ A \mid \begin{array}{l} A \rightarrow BC, \\ B \in V(1, 4), \\ C \in V(5, 1) \end{array} \right\}$

Algorithmus 8: CYK-Algorithmus

Eingabe: CFG $G = (V, \Sigma, P, S)$ in Chomsky-Normalform und Wort $w = a_1 \cdots a_n \in \Sigma^+$

Ausgabe: Ja, wenn $w \in L(G)$ und Nein, wenn $w \notin L(G)$

Beginn

für $i = 1$ bis n **tue**

└ $V(i, 1) := \{A \in V \mid A \rightarrow a_i \in P\}$

für $j = 2$ bis n **tue**

└ **für** $i = 1$ bis $n + 1 - j$ **tue**

└ $V(i, j) := \emptyset;$

└ **für** $k = 1$ bis $j - 1$ **tue**

└ $V(i, j) := V(i, j) \cup \left\{ A \in V \mid \begin{array}{l} A \rightarrow BC \in P, \\ B \in V(i, k), \\ C \in V(i + k, j - k) \end{array} \right\}$

wenn $S \in V(1, n)$ **dann**

└ return Ja

sonst

└ return Nein

Theorem

Das Wortproblem für Typ 2-Grammatiken ist entscheidbar:

Es gibt einen Algorithmus, der bei Eingabe von Typ 2-Grammatik G und Wort w nach endlicher Zeit entscheidet, ob $w \in L(G)$ gilt oder nicht. Zudem entscheidet er das Wortproblem in Polynomialzeit.

Theorem

Das Wortproblem für Typ 2-Grammatiken ist entscheidbar:

Es gibt einen Algorithmus, der bei Eingabe von Typ 2-Grammatik G und Wort w nach endlicher Zeit entscheidet, ob $w \in L(G)$ gilt oder nicht. Zudem entscheidet er das Wortproblem in Polynomialzeit.

Beweis Algorithmus 8 ist eine Entscheidungsprozedur.

Laufzeit des CYK-Algorithmus

Theorem

Das Wortproblem für Typ 2-Grammatiken ist entscheidbar:

Es gibt einen Algorithmus, der bei Eingabe von Typ 2-Grammatik G und Wort w nach endlicher Zeit entscheidet, ob $w \in L(G)$ gilt oder nicht. Zudem entscheidet er das Wortproblem in Polynomialzeit.

Beweis Algorithmus 8 ist eine Entscheidungsprozedur.

Er besteht aus drei geschachtelte für-Schleifen. Im Inneren wird noch über alle Produktionen aus P iteriert. Die Laufzeitkomplexität kann daher mit $O(n^3 \cdot |P|)$ abgeschätzt werden. □

Tobias Lindebar hat ein Web-Tool zum Üben und Anschauen entwickelt:

www.cip.ifi.lmu.de/~lindebar/

Das Endlichkeitsproblem

Das **Endlichkeitsproblem** für Typ i -Grammatiken ist die Frage, ob für eine gegebene Typ i -Grammatik G die Ungleichheit $|L(G)| < \infty$ gilt.

Das Endlichkeitsproblem

Das **Endlichkeitsproblem** für Typ i -Grammatiken ist die Frage, ob für eine gegebene Typ i -Grammatik G die Ungleichheit $|L(G)| < \infty$ gilt.

Satz

Das Endlichkeitsproblem für kontextfreie Grammatiken ist entscheidbar.

Das Endlichkeitsproblem

Das **Endlichkeitsproblem** für Typ i -Grammatiken ist die Frage, ob für eine gegebene Typ i -Grammatik G die Ungleichheit $|L(G)| < \infty$ gilt.

Satz

Das Endlichkeitsproblem für kontextfreie Grammatiken ist entscheidbar.

Wir brauchen zuerst ein Lemma.

Entscheiden des Endlichkeitsproblems

Lemma

Sei G eine CFG in Chomsky-Normalform.

Sei n die Zahl aus dem Pumping-Lemma für kontextfreie Sprachen.

Es gilt $|L(G)| = \infty$ g.d.w. es ein Wort $z \in L(G)$ mit $n \leq |z| < 2n$ gibt.

Entscheiden des Endlichkeitsproblems

Lemma

Sei G eine CFG in Chomsky-Normalform.

Sei n die Zahl aus dem Pumping-Lemma für kontextfreie Sprachen.

Es gilt $|L(G)| = \infty$ g.d.w. es ein Wort $z \in L(G)$ mit $n \leq |z| < 2n$ gibt.

Beweis

← Sei $z \in L(G)$ mit $n \leq |z| < 2n$. Mit dem Pumping-Lemma gilt $uv^iwx^iy \in L(G)$ für alle $i \in \mathbb{N}$. Also $|L(G)| = \infty$.

Entscheiden des Endlichkeitsproblems

\implies Durch Widerspruch. Wir nehmen an, es gibt kein Wort $z \in L(G)$ für $n \leq |z| < 2n$, aber trotzdem gilt $|L(G)| = \infty$.

Entscheiden des Endlichkeitsproblems

⇒ Durch Widerspruch. Wir nehmen an, es gibt kein Wort $z \in L(G)$ für $n \leq |z| < 2n$, aber trotzdem gilt $|L(G)| = \infty$.

Sei z das kürzeste Wort $\in L(G)$ mit $|z| \geq n$. Für dieses Wort gilt $|z| \geq 2n$.

Entscheiden des Endlichkeitsproblems

⇒ Durch Widerspruch. Wir nehmen an, es gibt kein Wort $z \in L(G)$ für $n \leq |z| < 2n$, aber trotzdem gilt $|L(G)| = \infty$.

Sei z das kürzeste Wort $\in L(G)$ mit $|z| \geq n$. Für dieses Wort gilt $|z| \geq 2n$.

Durch das Pumping-Lemma gibt es u, v, w, x, y mit $z = uvwxy$, $|vx| > 0$ und $|vwx| \leq n$, sodass insbesondere $uv^0wx^0y = uwy \in L(G)$ gilt.

Entscheiden des Endlichkeitsproblems

⇒ Durch Widerspruch. Wir nehmen an, es gibt kein Wort $z \in L(G)$ für $n \leq |z| < 2n$, aber trotzdem gilt $|L(G)| = \infty$.

Sei z das kürzeste Wort $\in L(G)$ mit $|z| \geq n$. Für dieses Wort gilt $|z| \geq 2n$.

Durch das Pumping-Lemma gibt es u, v, w, x, y mit $z = uvwxy$, $|vx| > 0$ und $|vwx| \leq n$, sodass insbesondere $uv^0wx^0y = uwy \in L(G)$ gilt.

Beobachte auch, dass $|uvwxy| \geq n$ und $|vwx| \leq n$ dazu führen, dass $|uwy| \geq |uy| \geq n$.

Entscheiden des Endlichkeitsproblems

⇒ Durch Widerspruch. Wir nehmen an, es gibt kein Wort $z \in L(G)$ für $n \leq |z| < 2n$, aber trotzdem gilt $|L(G)| = \infty$.

Sei z das kürzeste Wort $\in L(G)$ mit $|z| \geq n$. Für dieses Wort gilt $|z| \geq 2n$.

Durch das Pumping-Lemma gibt es u, v, w, x, y mit $z = uvwxy$, $|vx| > 0$ und $|vwx| \leq n$, sodass insbesondere $uv^0wx^0y = uwy \in L(G)$ gilt.

Beobachte auch, dass $|uvwxy| \geq n$ und $|vwx| \leq n$ dazu führen, dass $|uwy| \geq |uy| \geq n$.

Da $|uwy| < |uvwxy| = |z|$ und $|uwy| \geq n$ gelten, war z nicht minimal gewählt. Widerspruch. □

Entscheiden des Endlichkeitsproblems

Satz

Das Endlichkeitsproblem für kontextfreie Grammatiken ist entscheidbar.

Entscheiden des Endlichkeitsproblems

Satz

Das Endlichkeitsproblem für kontextfreie Grammatiken ist entscheidbar.

Beweis Entscheidungsprozedur:

Entscheiden des Endlichkeitsproblems

Satz

Das Endlichkeitsproblem für kontextfreie Grammatiken ist entscheidbar.

Beweis Entscheidungsprozedur:

1. Teste mit dem CYK-Algorithmus für alle Wörter $w \in \Sigma^+$ der Länge $n \leq |w| < 2n$, ob $w \in L(G)$ gilt.

Entscheiden des Endlichkeitsproblems

Satz

Das Endlichkeitsproblem für kontextfreie Grammatiken ist entscheidbar.

Beweis Entscheidungsprozedur:

1. Teste mit dem CYK-Algorithmus für alle Wörter $w \in \Sigma^+$ der Länge $n \leq |w| < 2n$, ob $w \in L(G)$ gilt.
2. Wenn $w \in L(G)$ für eines der Wörter gilt, dann $|L(G)| = \infty$.

Entscheiden des Endlichkeitsproblems

Satz

Das Endlichkeitsproblem für kontextfreie Grammatiken ist entscheidbar.

Beweis Entscheidungsprozedur:

1. Teste mit dem CYK-Algorithmus für alle Wörter $w \in \Sigma^+$ der Länge $n \leq |w| < 2n$, ob $w \in L(G)$ gilt.
2. Wenn $w \in L(G)$ für eines der Wörter gilt, dann $|L(G)| = \infty$.
3. Sonst $|L(G)| < \infty$. □