6b
Der CYK-Algorithmus

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 29. Mai 2024
Basierend auf Folien von PD Dr. David Sabel

Wiederholung: Entscheidbarkeit

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 $A \to w \in P$ 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 $A \to w \in P$ 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 $A \to w \in P$ 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.

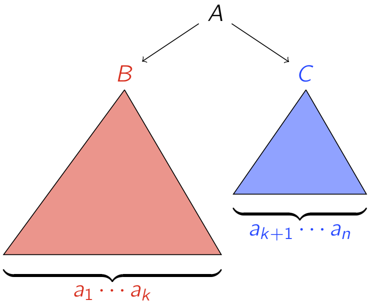
Grober Ansatz:

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 $A \to a_1 \in P$.
- Anderenfalls $(n > 1)$ kann u nur erzeugt werden, wenn
  - es eine Produktion $A \to BC \in P$ 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 $A \to BC \in P$ 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

$$\begin{aligned} P = \{S &\to AB \mid BA, \\ A &\to AA \mid AB \mid a, \\ B &\to BB \mid b \end{aligned}$$

und das Wort *bbbaab*.

Beispiel für naives rekursives Suchen

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

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

und das Wort *bbbaab*.

S erzeugt $bbbaab$, denn $S \to BA \in P$ und

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

Beispiel für naives rekursives Suchen

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

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

und das Wort $bbbaab$.

S erzeugt $bbbaab$, denn $S \to BA \in P$ und

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

Beispiel für naives rekursives Suchen

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

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

und das Wort $bbbaab$.

S erzeugt $bbbaab$, denn $S \to BA \in P$ und

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

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

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

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

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

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

3. Prüfe, ob $S \in V(1,n)$.

Beispiel für den CYK-Algorithmus

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

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

1. Schritt: Füllen der $V(i,1)$-Einträge

Beispiel für den CYK-Algorithmus

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

b b d d c

i

j

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

1. Schritt: Füllen der $V(i, 1)$-Einträge

Beispiel für den CYK-Algorithmus

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

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

1. Schritt: Füllen der $V(i, 1)$-Einträge

Beispiel für den CYK-Algorithmus

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

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

1. Schritt: Füllen der $V(i, 1)$-Einträge

Beispiel für den CYK-Algorithmus

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

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

1. Schritt: Füllen der $V(i, 1)$-Einträge

Beispiel für den CYK-Algorithmus

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

$i \longrightarrow$, $j \downarrow$

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

1. Schritt: Füllen der $V(i, 1)$-Einträge

Beispiel für den CYK-Algorithmus

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

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

2. Schritt:

Beispiel für den CYK-Algorithmus

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

i

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

j

2. Schritt: $j = 2,\ i = 1,\ k = 1$: $V(1,2) := V(1,2) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,1), \\ C \in V(2,1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 2,\ i = 2,\ k = 1$: $V(2,2) := V(2,2) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(2,1), \\ C \in V(3,1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

Seien $w = bbddc$ 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
$V(i,j)$	1	2	3	4	5
1	B	B	D	D	C
2		A			
3					
4					
5					

i →, j ↓

2. Schritt: $j = 2,\ i = 3,\ k = 1$: $V(3,2) := V(3,2) \cup \left\{ A \;\middle|\; \begin{array}{l} A \rightarrow BC, \\ B \in V(3,1), \\ C \in V(4,1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 2,\ i = 4,\ k = 1$: $V(4,2) := V(4,2) \cup \left\{ A \,\middle|\, \begin{array}{l} A \to BC, \\ B \in V(4,1), \\ C \in V(5,1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 3,\ i = 1,\ k = 1$: $V(1,3) := V(1,3) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,1), \\ C \in V(2,2) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 3,\ i = 1,\ k = 2$: $V(1,3) := V(1,3) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,2), \\ C \in V(3,1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

$$2.\ \text{Schritt: } j = 3,\ i = 2,\ k = 1\text{: } V(2,3) := V(2,3) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(2,1), \\ C \in V(3,2) \end{array} \right\}$$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 3,\ i = 2,\ k = 2$: $V(2,3) := V(2,3) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(2,2), \\ C \in V(4,1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 3,\ i = 3,\ k = 1$: $V(3,3) := V(3,3) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(3,1), \\ C \in V(4,2) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 3,\ i = 3,\ k = 2$: $V(3,3) := V(3,3) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(3,2), \\ C \in V(5,1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 4,\ i = 1,\ k = 1$: $V(1,4) := V(1,4) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,1), \\ C \in V(2,3) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 4,\ i = 1,\ k = 2$: $$V(1,4) := V(1,4) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,2), \\ C \in V(3,2) \end{array} \right\}$$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 4,\ i = 1,\ k = 3$: $V(1,4) := V(1,4) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,3), \\ C \in V(4,1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 4,\ i = 2,\ k = 1$: $V(2,4) := V(2,4) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(2,1), \\ C \in V(3,3) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 4,\ i = 2,\ k = 2$: $V(2,4) := V(2,4) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(2,2), \\ C \in V(4,2) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 4,\ i = 2,\ k = 3$: $V(2,4) := V(2,4) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(2,3), \\ C \in V(5,1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 5,\ i = 1,\ k = 1$: $V(1,5) := V(1,5) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,1), \\ C \in V(2,4) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 5,\ i = 1,\ k = 2$: $V(1,5) := V(1,5) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,2), \\ C \in V(3,3) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 5,\ i = 1,\ k = 3$: $V(1,5) := V(1,5) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,3), \\ C \in V(4,2) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 5,\ i = 1,\ k = 4$: $V(1,5) := V(1,5) \cup \left\{ A \,\middle|\, \begin{array}{l} A \to BC, \\ B \in V(1,4), \\ C \in V(5,1) \end{array} \right\}$

Beispiel für den CYK-Algorithmus

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

	b	b	d	d	c
	i				
$V(i,j)$	1	2	3	4	5
j = 1	B	B	D	D	C
2		A			
3		E			
4	A				
5	S				

3. Schritt: 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
$P = \{S \to TU,\ S \to UT,\ T \to TT,\ T \to TU,\ T \to a,\ U \to UU,\ U \to b\}$.

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

1. Schritt: 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
$P = \{S \to TU,\ S \to UT,\ T \to TT,\ T \to TU,\ T \to a,\ U \to UU,\ U \to b\}$.

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

1. Schritt: 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
$P = \{S \to TU,\ S \to UT,\ T \to TT,\ T \to TU,\ T \to a,\ U \to UU,\ U \to b\}$.

i

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

1. Schritt: 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
$P = \{S \to TU,\ S \to UT,\ T \to TT,\ T \to TU,\ T \to a,\ U \to UU,\ U \to b\}$.

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

j

1. Schritt: 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
$V(i,j)$	1	2	3	4	5
1	T	U	T	U	
2					
3					
4					
5					

1. Schritt: 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
$P = \{S \to TU,\ S \to UT,\ T \to TT,\ T \to TU,\ T \to a,\ U \to UU,\ U \to b\}$.

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

1. Schritt: 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
$P = \{S \to TU,\ S \to UT,\ T \to TT,\ T \to TU,\ T \to a,\ U \to UU,\ U \to b\}$.

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

j

2. Schritt:

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

2. Schritt: $j = 2,\ i = 1,\ k = 1$: $V(1,2) := V(1,2) \cup \left\{ A \;\middle|\; \begin{array}{l} A \rightarrow BC, \\ B \in V(1,1), \\ C \in V(2,1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

$$2.\ \text{Schritt: } j = 2,\ i = 2,\ k = 1:\ V(2,2) := V(2,2) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(2,1), \\ C \in V(3,1) \end{array} \right\}$$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 2,\ i = 3,\ k = 1$: $V(3,2) := V(3,2) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(3,1), \\ C \in V(4,1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 2,\ i = 4,\ k = 1$: $V(4, 2) := V(4, 2) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(4, 1), \\ C \in V(5, 1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 3$, $i = 1$, $k = 1$: $V(1,3) := V(1,3) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,1), \\ C \in V(2,2) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 3$, $i = 1$, $k = 2$: $V(1,3) := V(1,3) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,2), \\ C \in V(3,1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 3,\ i = 2,\ k = 1$: $V(2,3) := V(2,3) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(2,1), \\ C \in V(3,2) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 3,\ i = 2,\ k = 2$: $V(2,3) := V(2,3) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(2,2), \\ C \in V(4,1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2\. Schritt: $j = 3,\ i = 3,\ k = 1$: $V(3,3) := V(3,3) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(3,1), \\ C \in V(4,2) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 3,\ i = 3,\ k = 2$: $V(3,3) := V(3,3) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(3,2), \\ C \in V(5,1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

$i \longrightarrow$

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

$j \downarrow$

2. Schritt: $j = 4,\ i = 1,\ k = 1$: $V(1,4) := V(1,4) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,1), \\ C \in V(2,3) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 4,\ i = 1,\ k = 2$: $V(1,4) := V(1,4) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,2), \\ C \in V(3,2) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 4,\ i = 1,\ k = 3$: $V(1,4) := V(1,4) \cup \left\{ A \,\middle|\, \begin{array}{l} A \to BC, \\ B \in V(1,3), \\ C \in V(4,1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

	a	b	a	b	a
$V(i,j)$	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					

2. Schritt: $j = 4,\ i = 2,\ k = 1$: $V(2,4) := V(2,4) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(2,1), \\ C \in V(3,3) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 4,\ i = 2,\ k = 2$: $V(2,4) := V(2,4) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(2,2), \\ C \in V(4,2) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

	a	b	a	b	a
$V(i,j)$	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					

2. Schritt: $j = 4,\ i = 2,\ k = 3$: $V(2,4) := V(2,4) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(2,3), \\ C \in V(5,1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 5,\ i = 1,\ k = 1$: $V(1,5) := V(1,5) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,1), \\ C \in V(2,4) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 5,\ i = 1,\ k = 2$: $V(1,5) := V(1,5) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,2), \\ C \in V(3,3) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

	a	b	a	b	a
$V(i,j)$	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				

2. Schritt: $j = 5,\ i = 1,\ k = 3$: $V(1,5) := V(1,5) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,3), \\ C \in V(4,2) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

2. Schritt: $j = 5,\ i = 1,\ k = 4$: $V(1,5) := V(1,5) \cup \left\{ A \;\middle|\; \begin{array}{l} A \to BC, \\ B \in V(1,4), \\ C \in V(5,1) \end{array} \right\}$

Weiteres Beispiel für den CYK-Algorithmus

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

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

3. Schritt: Da $S \notin V(1,5)$, gilt $w \notin L(G)$.

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 ∈ V | A → a_i ∈ P}
für j = 2 bis n tue
    für i = 1 bis n + 1 − j tue
        V(i,j) := ∅;
        für k = 1 bis j − 1 tue
            V(i,j) := V(i,j) ∪ {A ∈ V | A → BC ∈ P, B ∈ V(i,k), C ∈ V(i+k, j−k)}
wenn S ∈ V(1,n) dann
    return Ja
sonst
    return Nein
```

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

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.

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.

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. $\square$

Online-Tool

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 Siehe Skript (nur FSK). $\square$

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. Berechne die Zahl n.

Entscheiden des Endlichkeitsproblems

Satz

Das Endlichkeitsproblem für kontextfreie Grammatiken ist entscheidbar.

Beweis Entscheidungsprozedur:

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