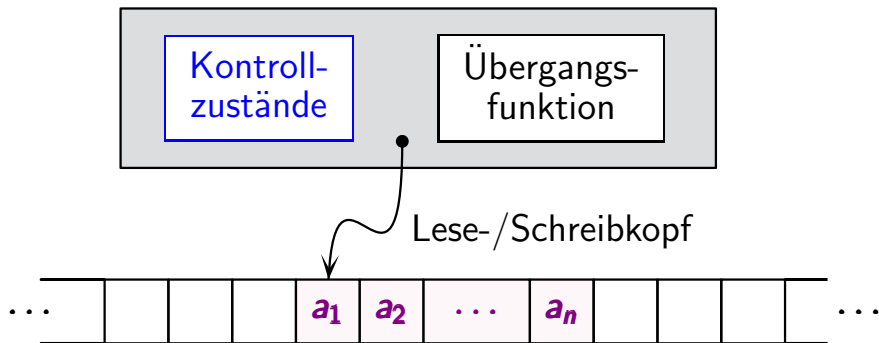


endliche Kontrolle mit unendlichem Band



- sequentieller Zugriff auf Bandzellen mittels Lese/Schreibkopf
- Band fungiert als Eingabeband und Arbeitsspeicher

gegeben: Grammatik $G = (V, \Sigma, \mathcal{P}, S)$ vom Typ 0
Arbeitsweise einer TM \mathcal{T}_G für die Sprache $\mathcal{L}(G)$
bei Eingabe w :

WHILE $w \neq S$ DO

 wähle nichtdeterministisch eine Regel $u \rightarrow v$

 IF v ist ein Teilwort von w

 THEN

 ersetze ein Vorkommen von v in w durch u

 FI

OD

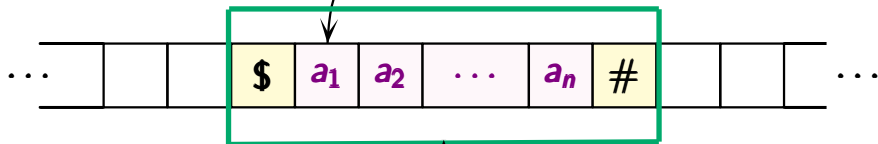
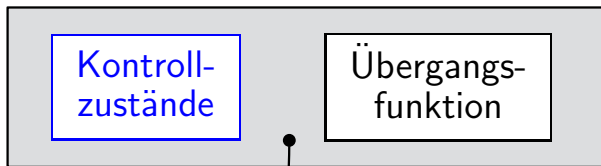
akzeptiere

$$S \Rightarrow^* xuy \Rightarrow xvy \Rightarrow^* w$$

Linear beschränkter Automat (LBA)

51

Turingmaschine mit zwei Begrenzungssymbolen #, \$



darf nicht
betreten werden

darf nicht
betreten werden

sämtliche Rechnungen finden **hier** statt

LBA für die Sprache

$$L = \left\{ w \in \{a, b, c\}^* : \begin{array}{l} \text{Anzahl } a\text{'s} = \text{Anzahl } b\text{'s} = \text{Anzahl } c\text{'s} \end{array} \right\}$$

z.B. $aabbcc$, $caabcb$, $cacbba \in L$

$aaabbb$, $abcabcc$, $aabbc \notin L$

Idee: LBA gestartet mit einem Wort $w \in \{a, b, c\}^*$

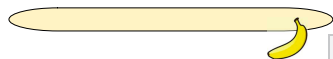
- streicht für jedes a in w ein b und ein c
- Terminierung: sobald alle a 's gelöscht sind

WHILE w enthält mindestens ein a DO
 ersetze ein Vorkommen von a durch \checkmark ;
IF w enthält mindestens ein b
 THEN ersetze ein Vorkommen von b durch \checkmark
 ELSE verwerfe FI
IF w enthält mindestens ein c
 THEN ersetze ein Vorkommen von c durch \checkmark
 ELSE verwerfe FI
OD
IF w besteht nur noch aus \checkmark 's
 THEN akzeptiere
 ELSE verwerfe FI

	Sprachklasse	Automatenmodell
Typ 0	beliebige Grammatik semientscheidbare Sprachen	Turingmaschinen NTM oder DTM
Typ 1	kontextsensitive Sprachen	linear beschränkte Automaten LBA oder ? DLBA ?
Typ 2	kontextfreie Sprachen	Kellerautomaten NKA
	deterministisch- kontextfreie Sprachen	deterministische KA DKA
Typ 3	reguläre Sprachen	endliche Automaten NFA oder DFA

	Typ 0	Typ 1	Typ 2	DCFL	Typ 3
Durchschnitt	✓	✓	nein	nein	✓
Vereinigung	✓	✓	✓	nein	✓
Komplement	nein	✓	nein	✓	✓
Konkatenation	✓	✓	✓	nein	✓
Kleeneabschluss	✓	✓	✓	nein	✓

DCFL deterministic contextfree languages



verwende vier atomare Formeln:

down

below

up

banana

- (1) Wenn der Affe nicht auf dem Stuhl steht, dann kann er den Stuhl unter die Banane schieben.
- (2) Der Affe kann auf den Stuhl klettern.
- (3) Wenn der Affe auf dem Stuhl steht und der Stuhl unter der Banane steht, dann kann der Affe die Banane erreichen.
- (4) Der Affe steht zunächst nicht auf dem Stuhl.

(1) *down* \rightarrow *below*

(2) *down* \rightarrow *up*

(3) *up* \wedge *below* \rightarrow *banana*

(4) *down*

} logische
Folgerung:
banana

- (1) Wenn der Affe nicht auf dem Stuhl steht, dann kann er den Stuhl unter die Banane schieben.
- (2) Der Affe kann auf den Stuhl klettern.
- (3) Wenn der Affe auf dem Stuhl steht und der Stuhl unter der Banane steht, dann kann der Affe die Banane erreichen.
- (4) Der Affe steht zunächst nicht auf dem Stuhl.

Aussagenlogische Formeln werden gebildet aus

- Aussagensymbolen $x, y, z, \dots \in AP$
- den Konstanten *true*, *false*
- logischen Verknüpfern wie

Negation	\neg	“nicht”	} Basisoperatoren
Konjunktion	\wedge	“und”	
Disjunktion	\vee	“oder”	} abgeleitet
	\vdots		

wobei AP = Menge von **Aussagensymbolen**

- * englische Abkürzung für “**atomic propositions**”
- * Boolesche Variablen, Atome, atomare Formeln

1. **true** ist eine aussagenlogische Formel.
2. Jedes Aussagensymbol $x \in AP$ ist eine aussagenlogische Formel.
3. Mit α und β sind auch $(\neg\alpha)$ und $(\alpha \wedge \beta)$ aussagenlogische Formeln.
4. Nichts sonst ist eine aussagenlogische Formel.

saloppe Kurzschreibweise (“abstrakte Syntax”):

$$\alpha ::= \mathbf{true} \mid x \mid \neg\alpha \mid \alpha \wedge \beta$$

abstrahiert von nicht-logischen Symbolen (Klammern)

Sei AP eine Menge von Aussagensymbolen.

$$\alpha ::= \mathit{true} \mid x \mid \neg\alpha \mid \alpha \wedge \beta$$

wobei $x \in AP$, $\neg \hat{=}$ Negation, $\wedge \hat{=}$ Konjunktion

abgeleitete Operatoren:

$$\mathit{false} \stackrel{\text{def}}{=} \neg\mathit{true}$$

$$\alpha \vee \beta \stackrel{\text{def}}{=} \neg(\neg\alpha \wedge \neg\beta)$$

$$\alpha \rightarrow \beta \stackrel{\text{def}}{=} \neg\alpha \vee \beta$$

$$\alpha \leftrightarrow \beta \stackrel{\text{def}}{=} (\alpha \wedge \beta) \vee (\neg\alpha \wedge \neg\beta)$$

\vdots

Disjunktion
("oder")

Implikation

Äquivalenz

Negation	\neg	hat höchste Priorität, gefolgt von
Konjunktion	\wedge	“und”
Disjunktion	\vee	“oder”
Implikation	\rightarrow	“wenn ..., dann ...”
Äquivalenz	\leftrightarrow	“genau dann, wenn”

Beispiel:

$$\neg x \wedge y \vee z = (((\neg x) \wedge y) \vee z)$$

$$x \wedge y \leftrightarrow \neg y \vee z = (x \wedge y) \leftrightarrow ((\neg y) \vee z)$$

aussagenlogische Formel α über AP
+ Belegung I für AP

Wahrheitswert
 $\alpha^I \in \{0, 1\}$

$0 \hat{=} \text{“falsch”}$ $1 \hat{=} \text{“wahr”}$

Belegung I für AP , auch **Interpretation** genannt

Abbildung $I : AP \rightarrow \{0, 1\}$, die jedem
Atom einen Wahrheitswert zuordnet

Induktive Definition der Wahrheitswerte $\alpha^I \in \{0, 1\}$
für gegebene Belegung $I : AP \rightarrow \{0, 1\}$

$$\mathit{true}^I \stackrel{\text{def}}{=} 1$$

$$x^I \stackrel{\text{def}}{=} I(x) \quad \text{für } x \in AP$$

$$(\neg \alpha)^I \stackrel{\text{def}}{=} 1 - \alpha^I = \begin{cases} 1 & : \text{ falls } \alpha^I = 0 \\ 0 & : \text{ falls } \alpha^I = 1 \end{cases}$$

$$(\alpha \wedge \beta)^I \stackrel{\text{def}}{=} \min\{\alpha^I, \beta^I\} = \begin{cases} 1 & : \text{ falls } \alpha^I = \beta^I = 1 \\ 0 & : \text{ sonst} \end{cases}$$

$$(\alpha \vee \beta)^I = \max\{\alpha^I, \beta^I\} = \begin{cases} 1 & : \text{falls } \alpha^I = 1 \\ & \text{oder } \beta^I = 1 \\ 0 & : \text{sonst} \end{cases}$$

$$(\alpha \rightarrow \beta)^I = \begin{cases} 1 & : \text{falls } \alpha^I \leq \beta^I \\ 0 & : \text{sonst} \end{cases}$$

$$(\alpha \leftrightarrow \beta)^I = \begin{cases} 1 & : \text{falls } \alpha^I = \beta^I \\ 0 & : \text{sonst} \end{cases}$$

$$\text{true}^I \stackrel{\text{def}}{=} 1 \quad x^I \stackrel{\text{def}}{=} I(x) \quad (\neg\alpha)^I \stackrel{\text{def}}{=} 1 - \alpha^I$$

$$(\alpha \wedge \beta)^I \stackrel{\text{def}}{=} \min\{\alpha^I, \beta^I\} = \begin{cases} 1 & : \text{falls } \alpha^I = \beta^I = 1 \\ 0 & : \text{sonst} \end{cases}$$

Formel α heißt **erfüllbar**,

falls es eine erfüllende Belegung \mathbf{I} für α gibt, d.h.,
falls $\alpha^{\mathbf{I}} = 1$ für wenigstens eine Belegung \mathbf{I} .

Formel α heißt **gültig** (oder **Tautologie**),

falls alle Belegungen erfüllend für α sind, d.h.,
falls $\alpha^{\mathbf{I}} = 1$ für alle Belegungen \mathbf{I} .

$y \wedge x \wedge (\neg y \vee \neg x)$ unerfüllbar, d.h., nicht erfüllbar

$(\neg y \rightarrow x) \wedge \neg y$ erfüllbar \leftarrow vgl. $y^{\mathbf{I}} = 0, x^{\mathbf{I}} = 1$

nicht gültig \leftarrow vgl. $y^{\mathbf{I}} = 1, x^{\mathbf{I}} = \dots$

$y \vee x \vee (\neg y \wedge \neg x)$ gültig

Beispiel: n -Damen Problem

135A

1	×				
2				×	
3		×			
4	×				
5			×		
	1	2	3	4	5

Aufgabe: platziere n Damen auf dem $n \times n$ Schachbrett so, dass sie sich nicht gegenseitig bedrohen.

Ziel: konstruiere aussagenlogische Formel α_n , so dass

erfüllende Belegungen für α_n

\cong Lösungen des n -Damen Problems

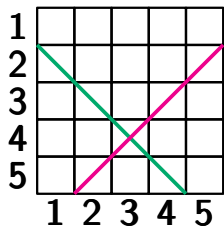
Beispiel: n -Damen Problem

139A

Die Lösungen (d.h., zulässige Plazierungen der n Damen) entsprechen den erfüllenden Belegungen der Formel

$$\bigwedge_{1 \leq i \leq n} \bigvee_{1 \leq j \leq n} x_{ij} \wedge \bigwedge_{1 \leq i, j \leq n} (x_{ij} \rightarrow \bigwedge_{(k, \ell) \in H(i, j)} \neg x_{k\ell})$$

$(k, \ell) \in H(i, j)$ gdw $(k, \ell) \neq (i, j)$ und



$$i+j = 7$$

$$i-j = 1$$

entweder $k = i$

oder $\ell = j$

oder $k - \ell = i - j$

oder $k + \ell = i + j$