FSK
Zentralübung 5

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik

Stand: 14. Juni 2023
Folien ursprünglich von PD Dr. David Sabel

Quiz: CYK-Algorithmus

Der CYK-Algorithmus

A ist benannt nach Noam Chomsky, Nobuo Yoneda und Stephen Cole Kleene

B löst das Wortproblem für Typ 1-Sprachen

C verwendet das Prinzip der dynamischen Programmierung

D hat exponentielle Laufzeit (in der Größe des Worts und Grammatik fester Größe)

E hat lineare Laufzeit (in der Größe des Worts und Grammatik fester Größe)

F erwartet als Eingabe eine Grammatik in Chomsky-Normalform

Quiz: CYK-Algorithmus

Der CYK-Algorithmus

A ist benannt nach Noam Chomsky, Nobuo Yoneda und Stephen Cole Kleene
nein, ist benannt nach John Cocke, Daniel Younger, Tadao Kasami

B löst das Wortproblem für Typ 1-Sprachen
nein

C verwendet das Prinzip der dynamischen Programmierung
ja

D hat exponentielle Laufzeit (in der Größe des Worts und Grammatik fester Größe)
nein, hat polynomielle Laufzeit

E hat lineare Laufzeit (in der Größe des Worts und Grammatik fester Größe)
nein, kubische Laufzeit

F erwartet als Eingabe eine Grammatik in Chomsky-Normalform
ja

Quiz: Chomsky-Normalform

Welche der folgenden Grammatiken ist in Chomsky-Normalform?

A $(\{S, A, B\}, \{a, b\}, \{S \to \varepsilon \mid AB,\ A \to a,\ B \to b\})$

B $(\{S, A, B\}, \{a, b\}, \{S \to AB \mid BA,\ A \to a \mid AA,\ B \to b \mid BB\})$

C $(\{S, A, B\}, \{a, b\}, \{S \to AB \mid BA \mid SB \mid b,\ A \to a \mid AA,\ B \to BB\})$

D $(\{S, A, B\}, \{a, b\}, \{S \to aS \mid AA \mid BB, A \to a, B \to b\})$

E $(\{S, A, B\}, \{a, b\}, \{S \to A \mid AB,\ A \to a \mid BSB,\ B \to b \mid bb\})$

Quiz: Chomsky-Normalform

Welche der folgenden Grammatiken ist in Chomsky-Normalform?

A $(\{S, A, B\}, \{a, b\}, \{S \to \varepsilon \mid AB,\ A \to a,\ B \to b\})$
nein, aufgrund von $S \to \varepsilon$

B $(\{S, A, B\}, \{a, b\}, \{S \to AB \mid BA,\ A \to a \mid AA,\ B \to b \mid BB\})$
ja

C $(\{S, A, B\}, \{a, b\}, \{S \to AB \mid BA \mid SB \mid b,\ A \to a \mid AA,\ B \to BB\})$
ja

D $(\{S, A, B\}, \{a, b\}, \{S \to aS \mid AA \mid BB, A \to a, B \to b\})$
nein, aufgrund von $S \to aS$

E $(\{S, A, B\}, \{a, b\}, \{S \to A \mid AB,\ A \to a \mid BSB,\ B \to b \mid bb\})$
nein, aufgrund von $S \to A$, $A \to BSB$ und $B \to bb\}$

Wiederholung: Das Pumping-Lemma für CFLs

Lemma (Pumping-Lemma für CFLs)

Sei L eine kontextfreie Sprache. Dann gibt es eine Zahl $n \in \mathbb{N}_{>0}$, sodass jedes Wort $z \in L$, das Mindestlänge n hat (d.h. $|z| \geq n$), als $z = uvwxy$ geschrieben werden kann, sodass gilt:

- $|vx| \geq 1$
- $|vwx| \leq n$
- für alle $i \geq 0$: $uv^i wx^i y \in L$.

Kontextfreiheit Widerlegen mit Pumping-Lemma als Spiel

Sei L die formale Sprache.

1. Der **Gegner** wählt die Zahl $n \in \mathbb{N}_{>0}$.
2. **Wir** wählen das Wort $z \in L$ mit $|z| \geq n$.
3. Der **Gegner** wählt die Zerlegung $z = uvwxy$ mit $|vx| \geq 1$ und $|vwx| \leq n$.
4. **Wir** gewinnen das Spiel, wenn wir ein $i \geq 0$ angeben können, sodass $uv^i wx^i w \notin L$.

Wenn wir **für jede Wahl des Gegners** das Spiel gewinnen können, dann haben wir gezeigt, dass L nicht kontextfrei ist.

Aufgabe

Aufgabe

Zeige, dass $L = \{a^m b a^m b a^m \mid m \in \mathbb{N}\}$ nicht kontextfrei ist.

Aufgabe

Aufgabe

Zeige, dass $L = \{a^m b a^m b a^m \mid m \in \mathbb{N}\}$ nicht kontextfrei ist.

Beweis mit Pumping-Lemma:

Aufgabe

Aufgabe

Zeige, dass $L = \{a^m b a^m b a^m \mid m \in \mathbb{N}\}$ nicht kontextfrei ist.

Beweis mit Pumping-Lemma:

- Sei n beliebig (vom Gegner gewählt).

Aufgabe

Aufgabe

Zeige, dass $L = \{a^m b a^m b a^m \mid m \in \mathbb{N}\}$ nicht kontextfrei ist.

Beweis mit Pumping-Lemma:

- Sei n beliebig (vom Gegner gewählt).
- Wir wählen $z = a^n b a^n b a^n$.

Aufgabe

Aufgabe

Zeige, dass $L = \{a^m b a^m b a^m \mid m \in \mathbb{N}\}$ nicht kontextfrei ist.

Beweis mit Pumping-Lemma:

- Sei n beliebig (vom Gegner gewählt).
- Wir wählen $z = a^n b a^n b a^n$.
- Sei $z = uvwxy$ mit $|vwx| \leq n$ und $|vx| \geq 1$ (vom Gegner zerlegt).

Aufgabe

Aufgabe

Zeige, dass $L = \{a^m b a^m b a^m \mid m \in \mathbb{N}\}$ nicht kontextfrei ist.

Beweis mit Pumping-Lemma:

- Sei n beliebig (vom Gegner gewählt).
- Wir wählen $z = a^n b a^n b a^n$.
- Sei $z = uvwxy$ mit $|vwx| \leq n$ und $|vx| \geq 1$ (vom Gegner zerlegt).
- Dann kann vwx nicht zwei $b's$ enthalten.

Aufgabe

Aufgabe

Zeige, dass $L = \{a^m b a^m b a^m \mid m \in \mathbb{N}\}$ nicht kontextfrei ist.

Beweis mit Pumping-Lemma:

- Sei n beliebig (vom Gegner gewählt).
- Wir wählen $z = a^n b a^n b a^n$.
- Sei $z = uvwxy$ mit $|vwx| \leq n$ und $|vx| \geq 1$ (vom Gegner zerlegt).
- Dann kann vwx nicht zwei $b's$ enthalten.
- Fall: vx enthält ein b. Dann kann uv^0wx^0y nicht in L liegen, da das b entfernt wurde.

Aufgabe

Aufgabe

Zeige, dass $L = \{a^m b a^m b a^m \mid m \in \mathbb{N}\}$ nicht kontextfrei ist.

Beweis mit Pumping-Lemma:

- Sei n beliebig (vom Gegner gewählt).
- Wir wählen $z = a^n b a^n b a^n$.
- Sei $z = uvwxy$ mit $|vwx| \leq n$ und $|vx| \geq 1$ (vom Gegner zerlegt).
- Dann kann vwx nicht zwei $b's$ enthalten.
- Fall: vx enthält ein b. Dann kann uv^0wx^0y nicht in L liegen, da das b entfernt wurde.
- Fall vx enthält kein b. Dann $uv^2wx^2y \notin L$, da maximal zwei a-Folgen aufgepumpt wurden, die dritte a-Folge aber noch aus n vielen a's besteht (und die Trennung durch b noch vorhanden ist).

Kellerautomaten (PDAs)

Definition (Kellerautomat, PDA)

Ein (nichtdeterministischer) Kellerautomat (PDA, pushdown automaton) ist ein Tupel $M = (Z, \Sigma, \Gamma, \delta, z_0, \#)$, wobei

- Z ist eine endliche Menge von Zuständen,
- Σ ist das (endliche) Eingabealphabet,
- Γ ist das (endliche) Kelleralphabet,
- $\delta : (Z \times (\Sigma \cup \{\varepsilon\}) \times \Gamma) \to \mathcal{P}_e(Z \times \Gamma^*)$ ist die Zustandsüberführungsfunktion (oder nur Überführungsfunktion)
- $z_0 \in Z$ ist der Startzustand und
- $\# \in \Gamma$ ist das Startsymbol im Keller.

Quizfragen (1)

Eine Konfiguration eines Kellerautomaten ist ein Tupel (z, w, W).
Was sind z, w und W?

Quizfragen (1)

Eine Konfiguration eines Kellerautomaten ist ein Tupel (z, w, W).
Was sind z, w und W?

- z ist der aktuelle Zustand
- w ist die Resteingabe
- W ist der aktuelle Kellerinhalt

Quizfragen (2)

Sei $(z', B_1 \cdots B_k) \in \delta(z, a, A)$. Dann gilt

$$(z, aw, AW) \vdash ???$$

Was ist für ??? einzusetzen?

Quizfragen (2)

Sei $(z', B_1 \cdots B_k) \in \delta(z, a, A)$. Dann gilt

$$(z, aw, AW) \vdash ???$$

Was ist für ??? einzusetzen?

$$(z', w, B_1 \cdots B_k W)$$

Quizfragen (2)

Sei $(z', B_1 \cdots B_k) \in \delta(z, a, A)$. Dann gilt

$$(z, aw, AW) \vdash ???$$

Was ist für ??? einzusetzen?

$$(z', w, B_1 \cdots B_k W)$$

Sei $(z', B_1 \cdots B_k) \in \delta(z, \varepsilon, A)$. Dann gilt

$$???_1 \vdash ???_2$$

Was ist für $???_1$ und $???_2$ einzusetzen?

Quizfragen (2)

Sei $(z', B_1 \cdots B_k) \in \delta(z, a, A)$. Dann gilt

$$(z, aw, AW) \vdash ???$$

Was ist für ??? einzusetzen?

$$(z', w, B_1 \cdots B_k W)$$

Sei $(z', B_1 \cdots B_k) \in \delta(z, \varepsilon, A)$. Dann gilt

$$???_1 \vdash ???_2$$

Was ist für $???_1$ und $???_2$ einzusetzen?

$$(z, w, AW) \vdash (z', w, B_1 \cdots B_k W)$$

Quizfragen (3)

Vervollständige: $w \in L(M)$ g.d.w. es existiert z, sodass $(z_0, w, \#) \vdash_M^* ???$ für Kellerautomaten.

Quizfragen (3)

Vervollständige: $w \in L(M)$ g.d.w. es existiert z, sodass $(z_0, w, \#) \vdash_M^* ???$ für Kellerautomaten.

$$(z, \varepsilon, \varepsilon)$$

Quizfragen (3)

Vervollständige: $w \in L(M)$ g.d.w. es existiert z, sodass $(z_0, w, \#) \vdash_M^*$??? für Kellerautomaten.

$$(z, \varepsilon, \varepsilon)$$

Vervollständige: $w \in L(M)$ g.d.w. es existiert z, sodass $(z_0, w, \#) \vdash_M^*$??? für Kellerautomaten mit Endzuständen E.

Quizfragen (3)

Vervollständige: $w \in L(M)$ g.d.w. es existiert z, sodass $(z_0, w, \#) \vdash_M^*$??? für Kellerautomaten.

$$(z, \varepsilon, \varepsilon)$$

Vervollständige: $w \in L(M)$ g.d.w. es existiert z, sodass $(z_0, w, \#) \vdash_M^*$??? für Kellerautomaten mit Endzuständen E.

$$(z, \varepsilon, W), \text{ wo } z \in E$$

PDA entwerfen

Aufgabe

Entwerfen Sie einen PDA, der die Sprache $L = \{a^{2n}b^n \mid n \in \mathbb{N}_{>0}\}$ akzeptiert.

PDA entwerfen

Aufgabe

Entwerfen Sie einen PDA, der die Sprache $L = \{a^{2n}b^n \mid n \in \mathbb{N}_{>0}\}$ akzeptiert.

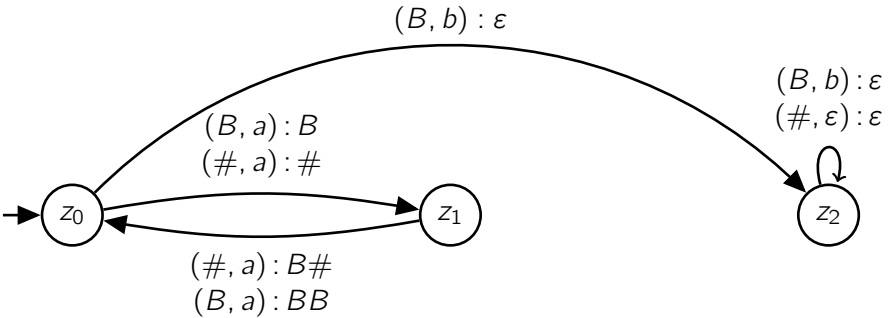
- Keller $= \underbrace{B \cdots B}_{i\text{-mal}}\#$: noch i mal b lesen

PDA entwerfen

Aufgabe

Entwerfen Sie einen PDA, der die Sprache $L = \{a^{2n}b^n \mid n \in \mathbb{N}_{>0}\}$ akzeptiert.

- Keller $= \underbrace{B \cdots B}_{i\text{-mal}}\#$: noch i mal b lesen



Quiz: Deterministische Kellerautomaten

Betrachte einen deterministischen Kellerautomaten (mit Endzuständen). Welche der folgenden Bedingungen gelten stets?

A Es gibt keine ε-Übergänge.

B Wenn es einen ε-Übergang mit Kellersymbol A und Zustand z gibt, dann gibt es keinen anderen Übergang für A und z

C Wenn $(z', W) \in \delta(z, a, A)$ gilt, dann gibt es keinen ε-Übergang für Zustand z und Kellersymbol A.

D Für alle z, a, A: $\delta(z, a, A) \neq \emptyset$.

E Wenn $(z', W) \in \delta(z, a, A)$ gilt, dann gibt es keinen Zustand und Kellersymbole $(z'', W') \in \delta(z, a, A)$ mit $z' \neq z''$.

F Wenn $(z', W) \in \delta(z, a, A)$ gilt, dann gibt es keinen Zustand und Kellersymbole $(z'', W') \in \delta(z, a, A')$ für alle $A' \neq A$.

Quiz: Deterministische Kellerautomaten

Betrachte einen deterministischen Kellerautomaten (mit Endzuständen). Welche der folgenden Bedingungen gelten stets?

A Es gibt keine ε-Übergänge. falsch

B Wenn es einen ε-Übergang mit Kellersymbol A und Zustand z gibt, dann gibt es keinen anderen Übergang für A und z richtig

C Wenn $(z', W) \in \delta(z, a, A)$ gilt, dann gibt es keinen ε-Übergang für Zustand z und Kellersymbol A. richtig

D Für alle z, a, A: $\delta(z, a, A) \neq \emptyset$. falsch

E Wenn $(z', W) \in \delta(z, a, A)$ gilt, dann gibt es keinen Zustand und Kellersymbole $(z'', W') \in \delta(z, a, A)$ mit $z' \neq z''$. richtig

F Wenn $(z', W) \in \delta(z, a, A)$ gilt, dann gibt es keinen Zustand und Kellersymbole $(z'', W') \in \delta(z, a, A')$ für alle $A' \neq A$. falsch

Allgemein ist die Bedingung bei DPDAs:

$$|\delta(z, a, A)| + |\delta(z, \varepsilon, A)| \leq 1$$

Definition der Turingmaschine

Definition (Turingmaschine)

Eine Turingmaschine (TM) ist ein 7-Tupel $M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$ mit

- Z ist eine endliche Menge von Zuständen,
- Σ ist das (endliche) Eingabealphabet,
- $\Gamma \supset \Sigma$ ist das (endliche) Bandalphabet,
- δ ist die Zustandsüberführungsfunktion
  - deterministische TM (DTM): $\delta : Z \times \Gamma \to Z \times \Gamma \times \{L, R, N\}$,
  - nichtdeterministische TM (NTM): $\delta : Z \times \Gamma \to \mathcal{P}(Z \times \Gamma \times \{L, R, N\})$
- $z_0 \in Z$ ist der Startzustand,
- $\square \in \Gamma \setminus \Sigma$ ist das Blank-Symbol und
- $E \subseteq Z$ ist die Menge der Endzustände.

Konfigurationen

Konfiguration ist ein Wort wzw', sodass:

- die TM ist im Zustand z,
- auf dem Band steht $\cdots \square\square ww'\square\square \cdots$ und
- der Schreib-Lesekopf steht auf dem ersten Symbol von w'.

Startkonfiguration: $z_0 w$

Quiz: Sei $\delta(z_0, a) = (z_1, A, R), \delta(z_0, b) = (z_1, B, L), \delta(z_0, c) = (z_1, C, N)$.

Was ist die Nachfolgekonfiguration von abz_0cabc?

A $abCz_0abcC$
B abz_1Cabc
C az_1Bcabc
D $z_1AbcAbc$

Konfigurationen

Konfiguration ist ein Wort wzw', sodass:

- die TM ist im Zustand z,
- auf dem Band steht $\cdots \square\square ww'\square\square \cdots$ und
- der Schreib-Lesekopf steht auf dem ersten Symbol von w'.

Startkonfiguration: $z_0 w$

Quiz: Sei $\delta(z_0, a) = (z_1, A, R), \delta(z_0, b) = (z_1, B, L), \delta(z_0, c) = (z_1, C, N)$.

Was ist die Nachfolgekonfiguration von abz_0cabc?

A $abCz_0abcC \rightarrow$ Nein
B $abz_1Cabc \rightarrow$ Ja
C $az_1Bcabc \rightarrow$ Nein
D $z_1AbcAbc \rightarrow$ Nein

Konfigurationen

Konfiguration ist ein Wort wzw', sodass:

- die TM ist im Zustand z,
- auf dem Band steht $\cdots \square\square ww'\square\square \cdots$ und
- der Schreib-Lesekopf steht auf dem ersten Symbol von w'.

Startkonfiguration: $z_0 w$

Quiz: Sei $\delta(z_0, a) = (z_1, A, R), \delta(z_0, b) = (z_1, B, L), \delta(z_0, c) = (z_1, C, N)$.

Was ist die Nachfolgekonfiguration von abz_0cabc? Was ist die Nachfolgekonfiguration von z_0bcbca?

A $abCz_0abcC \rightarrow$ Nein
B $abz_1Cabc \rightarrow$ Ja
C $az_1Bcabc \rightarrow$ Nein
D $z_1AbcAbc \rightarrow$ Nein

A Bz_1cbca
B z_1Bcbca
C $Bcbcaz_1$
D $z_1\square Bcbca$

Konfigurationen

Konfiguration ist ein Wort wzw', sodass:

- die TM ist im Zustand z,
- auf dem Band steht $\cdots \square\square ww'\square\square \cdots$ und
- der Schreib-Lesekopf steht auf dem ersten Symbol von w'.

Startkonfiguration: $z_0 w$

Quiz: Sei $\delta(z_0, a) = (z_1, A, R), \delta(z_0, b) = (z_1, B, L), \delta(z_0, c) = (z_1, C, N)$.

Was ist die Nachfolgekonfiguration von abz_0cabc? Was ist die Nachfolgekonfiguration von z_0bcbca?

A $abCz_0abcC \rightarrow$ Nein
B $abz_1Cabc \rightarrow$ Ja
C $az_1Bcabc \rightarrow$ Nein
D $z_1AbcAbc \rightarrow$ Nein

A $Bz_1cbca \rightarrow$ Nein
B $z_1Bcbca \rightarrow$ Nein
C $Bcbcaz_1 \rightarrow$ Nein
D $z_1\square Bcbca \rightarrow$ Ja

Definition (Akzeptierte Sprache einer TM)

Sei $M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$ eine TM.
Die von M akzeptierte Sprache $L(M)$ ist definiert als

$$L(M) := \{w \in \Sigma^* \mid \exists u, v \in \Gamma^*, z \in E : z_0 w \vdash_M^* uzv\}$$

Weitere Begriffe:

- TM akzeptiert: TM erreicht Endzustand
- TM verwirft: TM erreicht keinen Endzustand

Aufgabe

Aufgabe

Geben Sie eine Turingmaschine an, die genau bei Eingabe a^{2^n} mit $n \in \mathbb{N}$ akzeptiert.

Vorgehen:

- Überlege Idee zum Erkennen von L.
- „Programmiere“ dies erstmal grob, phasenweise.
- Modelliere es genauer mit Zuständen und Übergängen.

Aufgabe

Aufgabe

Geben Sie eine Turingmaschine an, die genau bei Eingabe a^{2^n} mit $n \in \mathbb{N}$ akzeptiert.

Idee zum Erkennen von L:

- Auf dem Band steht zu Beginn $\cdots \square a \cdots a \square \cdots$
- Erkenne, ob Anzahl an a's eine Zahl der Form 2^n ist.
- Idee: $2^n = 2 \cdot 2^{n-1} \implies$ streicht man die Hälfte aller a's, so muss der Rest immer noch von der Form a^{2^m} sein.
- $a^{2^n} \to a^{2^{n-1}} \to a^{2^{n-2}} \to \cdots$
- Wenn man keine gleichen Hälften hat, dann stand dort a^n mit n ungerade. $\implies$ verwirf
- Ausnahmefall: $n = 0$: Auf den Band steht **ein** a.

Aufgabe

Aufgabe

Geben Sie eine Turingmaschine an, die genau bei Eingabe a^{2^n} mit $n \in \mathbb{N}$ akzeptiert.

„Programmiere“ dies erstmal grob, phasenweise

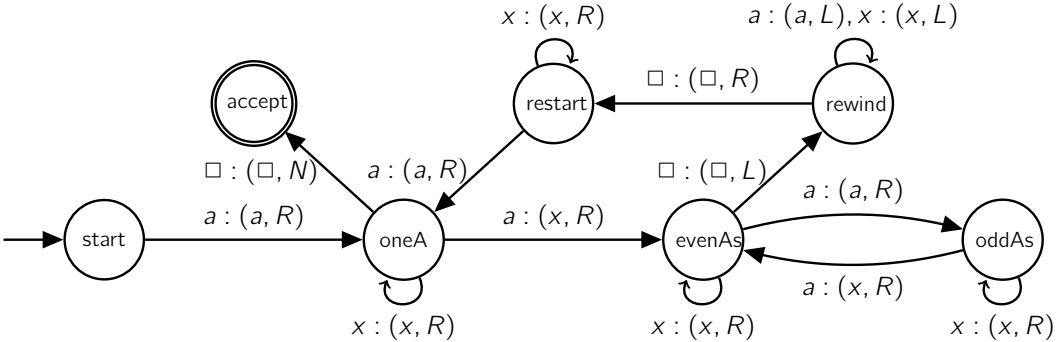
- Wenn nur ein a, dann akzeptiere.
- Hälfte streichen: Ersetze jedes **zweite** a durch x.
- Streichen: Ersetze a durch Symbol x.
- Wenn ungerade viele a's und mehr als ein a gestrichen wurden, dann verwirf.
- Starte von Neuem mit nächster Phase.

Aufgabe

Aufgabe

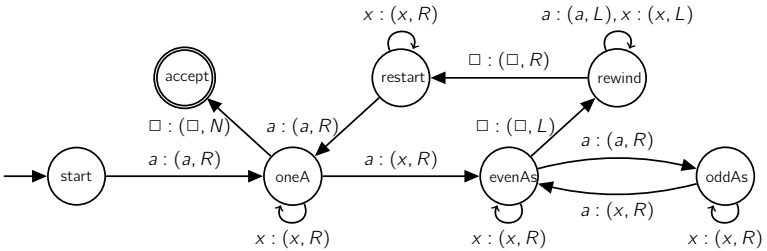
Geben Sie eine Turingmaschine an, die genau bei Eingabe a^{2^n} mit $n \in \mathbb{N}$ akzeptiert.

Modelliere es genauer mit Zuständen und Übergängen



Für DTM: Alle anderen Übergänge in einen zusätzlichen Müllzustand

Lauf der TM auf aaaa



start *aaaa*	⊢ *a* oneA *aaa*	⊢ *ax* evenAs *aa*	⊢ *axa* oddAs *a*
⊢ *axax* evenAs □	⊢ *axa* rewind *x*	⊢ *ax* rewind *ax*	⊢ *a* rewind *xax*
⊢ rewind *axax*	⊢ rewind □*axax*	⊢ restart *axax*	⊢ *a* oneA *xax*
⊢ *ax* oneA *ax*	⊢ *axx* evenAs *x*	⊢ *axxx* evenAs □	⊢ *axx* rewind *x*
⊢ *ax* rewind *xx*	⊢ *a* rewind *xxx*	⊢ rewind *axxx*	⊢ rewind □*axxx*
⊢ restart *axxx*	⊢ *a* oneA *xxx*	⊢ *ax* oneA *xx*	⊢ *axx* oneA *x*
⊢ *axxx* oneA □	⊢ *axxx* accept □		

Input für http://turingmachinesimulator.com

http://turingmachinesimulator.com/shared/fsxyubfqam

```
name: a^2^n
init: start
accept: accept
// delta:
start,a
oneA,a,>

oneA,a
evenAs,x,>

oneA,x
oneA,x,>

oneA,_
accept,_,-

evenAs,a
oddAs,a,>

evenAs,x
evenAs,x,>

oddAs,a
evenAs,x,>

oddAs,x
oddAs,x,>

evenAs,_
rewind,_,<

rewind,a
rewind,a,<

rewind,x
rewind,x,<

rewind,_
restart,_,>

restart,x
restart,x,>

restart,_
accept,_,-

restart,a
oneA,a,>
```