Informatik III

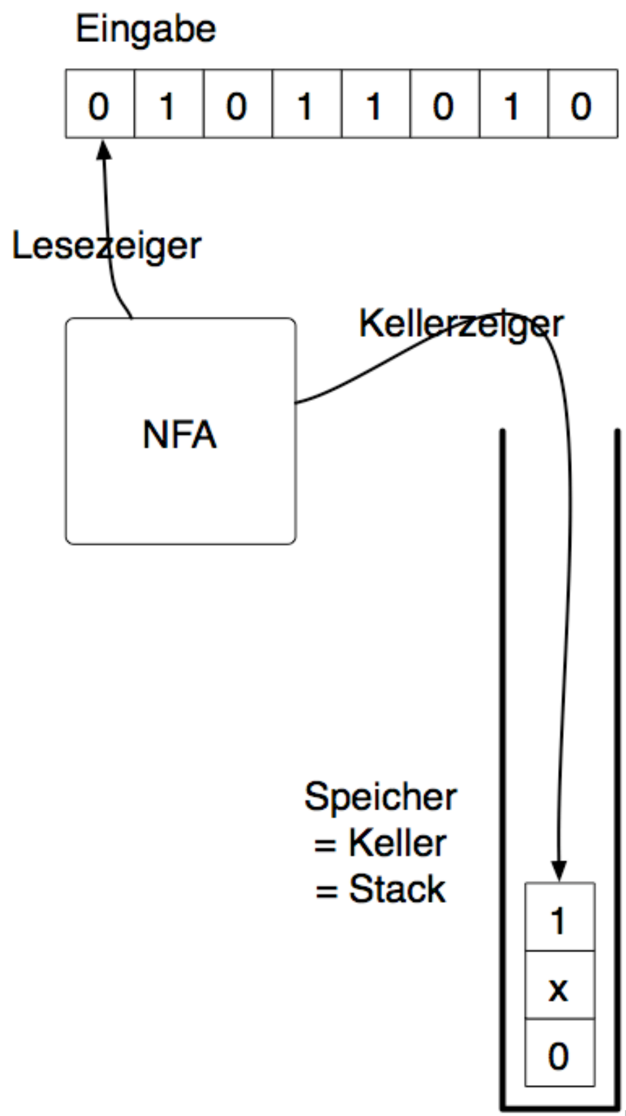
Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Christian Schindelhauer
Wintersemester 2006/07
8. Vorlesung
17.11.2006
schindel@informatik.uni-freiburg.de

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Prinzip des Kellerautomats Push-Down-Automaton (PDA)

- **Ein Kellerautomat vereinigt einen**
  - NFA mit einem
  - Keller (Stack)
- **Der Keller kann potenziell beliebig viel Zeichen speichern**
- **Zugriff ist eingeschränkt:**
  - Pop: Auslesen des obersten Zeichens (inklusive Entfernen)
  - Push: Hineinlegen eines Zeichens
- **Auf den Übergängen des NFA wird der Zugriff auf den Keller festgelegt**
  - zusätzlich zum aktuellen Zeichen der Eingabe,
  - die weiterhin von links nach rechts gelesen wird.



Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Keller-Automat: Formale Definition

- **Definition**
  - Ein Kellerautomat (pushdown automaton - PDA) wird durch ein Sechser-Tupel $(Q, \Sigma, \Gamma, \delta, q_0, F)$, wobei Q, Σ, Γ, F endliche Mengen sind und
  1. Q ist die Menge der Zustände
  2. Σ ist das Eingabealphabet
  3. Γ ist das Kelleralphabet
  4. $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow \mathbf{P}(Q \times \Gamma_\varepsilon)$ ist die Übergangsfunktion
  5. q_0 ist der Startzustand
  6. $F \subseteq Q$ ist die Menge der akzeptierenden Zustände

- **Ein PDA akzeptiert die Eingabe w,**
  - wenn es eine Darstellung $w= w_1 w_2 \dots w_m$ mit $w_i \in \Sigma_\varepsilon$ gibt
  - wenn es eine Zustandsfolge $q = r_0 r_2 \dots r_m$ mit $s_i \in Q$ gibt
  - wenn es Zeichenketten $s_0, s_1, \dots, s_m \in \Gamma_\varepsilon^*$ gibt, so dass

1. $r_0 = q_0$ und $s_0 = \varepsilon$
  - Startzustand mit leeren Keller

2. Für i = 0, ..., m-1 gilt:
  - $(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$, wobei
  - $s_i = at$ und $s_{i+1} = bt$
    für passende $a,b \in \Gamma_\varepsilon$, $t \in \Gamma_\varepsilon^*$
  - Übergang mit Kellerverhalten:
    - Lies a, Schreibe b

3. $r_m \in F$
  - Ein akzeptierender Zustand erscheint als Endzustand

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Letzte Vorlesung

- **Kontextfreie Grammatiken:**
  - „Wortersetzer“
- **CYK-Algorithmus:**
  - Löst das „Wortproblem“
- **Kellerautomaten:**
  - NFA mit einem Keller
- **Theorem 6.1:**
  - Eine Sprache ist genau dann kontextfrei, wenn ein Kellerautomat sie erkennt
- **Lemma 6.1:**
  - Ist eine Sprache kontextfrei, dann erkennt Sie ein Kellerautomat.
- **Es fehlt noch:**
  - Lemma 6.2:
    - Die Sprache, die von einem Kellerautomat erkannt wird, ist kontextfrei.

PDAs erkennen nur kontextfreie Sprachen

- **Lemma 6.2**
  - Die Sprache, die von einem Kellerautomat erkannt wird, ist kontextfrei
- **Vorbereitung:**
  - Ein PDA kann so modifiziert werden ohne seine Sprache zu verändern, dass die folgenden Eigenschaften gelten:
    1. Es gibt nur einen einzigen akzeptierenden Zustand q_{akz}
    2. Bevor der PDA akzeptiert, leert der PDA seinen Keller
    3. In jedem Übergang wird
       - entweder ein Zeichen vom Keller geholt oder
       - ein Zeichen abgelegt.
       - aber nicht beides zugleich
  - Die Variable A[rs] beschreibt alle Worte, die der PDA abarbeiten kann, wenn er mit einem leeren Keller bei Zustand r beginnt und bei Zustand s wieder mit einem leeren Keller endet

PDAs erkennen nur kontextfreie Sprachen

- **Lemma 6.2**
  - Die Sprache, die von einem Kellerautomat erkannt wird, ist kontextfrei
- **Beweis**
  - Für $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{akz}\})$ konstruieren wir eine kontextfreie Grammatik G
  - Die Variablen von G sind: $\{A[pq] \mid p,q \in Q\}$
  - Die Startvariable ist $A[q_0 q_{akz}]$
  - Die Ersetzungsregeln sind:
    - Für jedes $p,q,r,s \in Q$, $t \in \Gamma$ und $a,b \in \Sigma$
      - falls $\delta(p,a,\varepsilon)$ das Element (r,t) enthält und
      - falls $\delta(s,b,t)$ das Element (q, ε),
        - füge $A[pq] \rightarrow a\, A[rs]\, b$ zu G hinzu
    - Für jedes $p,q,r \in Q$ füge
      - $A[pq] \rightarrow A[pr]\, A[rq]$ zu G hinzu
    - Für jedes $p \in Q$ füge die Regel
      - $A[pp] \rightarrow \varepsilon$ zu G hinzu

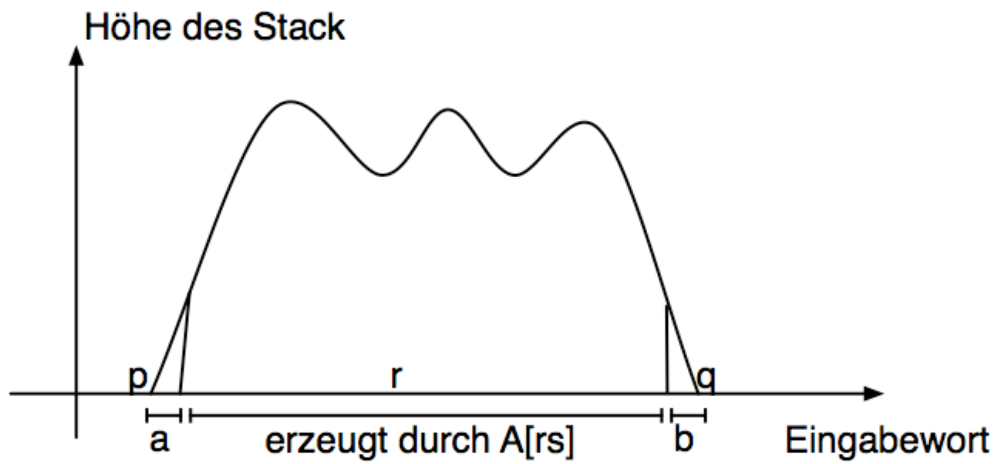
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

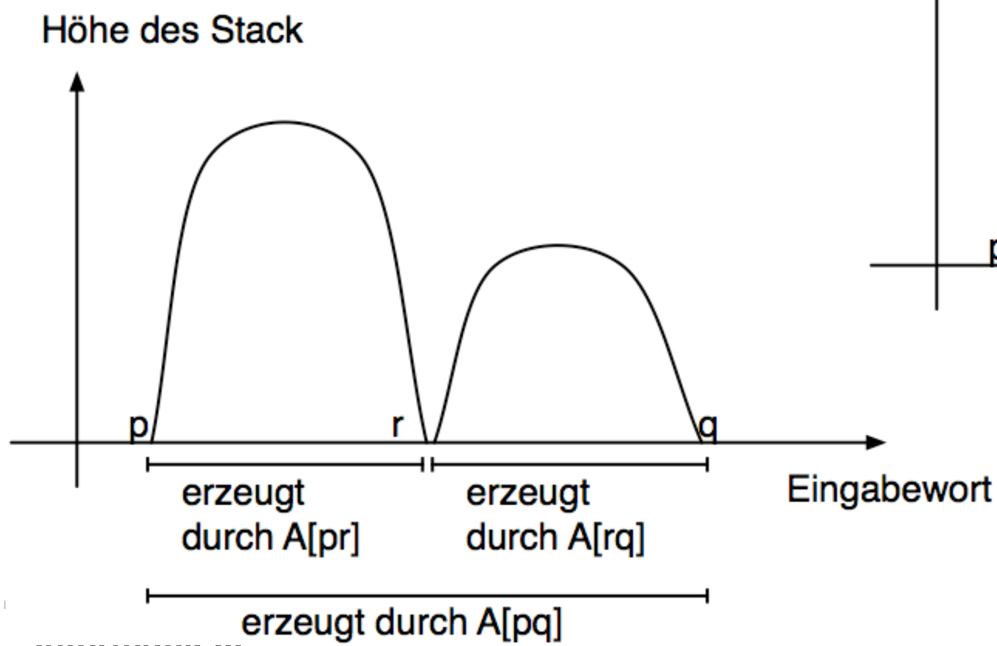
PDAs erkennen nur kontextfreie Sprachen Intuition zur Konstruktion

Die Ersetzungsregeln sind:

– Für jedes p,q,r,s $\in$ Q, t $\in \Gamma$ und a,b $\in \Sigma$
  - falls δ(p,a,ε) das Element (r,t) enthält und
  - falls δ(s,b,t) das Element (q, ε),
  - füge A[pq] $\rightarrow$ aA[rs]b zu G hinzu

– Für jedes p,q,r $\in$ Q füge
  - A[pq] $\rightarrow$ A[pr] A[rq] zu G hinzu





PDAs erkennen nur kontextfreie Sprachen - Korrektheit

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Ergebnis der Konstruktion:**
  - Wenn A[pq] ein Wort x erzeugt, so kann x den PDA P vom Zustand p in den Zustand q überführen, wobei am Anfang und Ende der Stapel jeweils leer ist
    - Beweis per Induktion über die Anzahl der Ableitungsschritte, die zur Ableitung von x aus der Variablen A[pq] benötigt werden
  - Kann x den PDA P vom Zustand p in den Zustand q überführen, wobei am Anfang und Ende der Stapel jeweils leer ist, so kann x aus A[pq] abgeleitet werden
    - Beweis per Induktion über die Anzahl der Schritte in der Rechnung, die p in q überführen
  - Aus der Startvariable können genau diejenigen Worte $x \in \Sigma^*$ abgeleitet werden, die P aus dem Zustand q_0 in den Zustand q_{accept} überführen, wobei am Anfang und am Ende der Stapel jeweils leer ist.
- **Dies sind genau die Worte, die von P akzeptiert werden. Somit ist L = L(G) = L(P)**

Überblick: Kontextfreie Sprachen

- **Formale Grammatik**
  - Einführung, Beispiele
  - Formale Definition
  - Das Wort-Problem
  - Chomsky Normalform
  - Cocke-Younger-Kasami-Algorithmus
- **Kellerautomaten**
  - Formale Definition
  - Beispiele
  - Äquivalenz zu kontextfreien Sprachen
- **Nichtkontextfreie Sprachen**
  - Pumping-Lemma für kontextfreie Sprachen

Reguläre Sprachen vs. kontextfreie Sprachen

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Korollar 6.3:**
  - Jede reguläre Sprache L ist auch kontextfrei.
- **Beweis:**
  - Jede reguläre Sprache L besitzt mindestens einen DFA A, der diese Sprache akzeptiert
  - Jeder DFA ist auch ein NFA
  - Jeder NFA ist auch ein PDA
  - Folglich gibt es einen PDA (und zwar A), der L akzeptiert
  - Damit ist L auch kontextfrei

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Nicht-kontextfreie Sprachen

Das Pumping-Lemma für reguläre Sprachen

- **Pumping-Lemma für reguläre Sprachen**
  - Sei A eine reguläre Sprache.
    - Dann gibt es eine Zahl $p>0$
    - so dass für jedes Wort s mit $|s|\geq p$
    - s in drei Teile geteilt werden kann: $s = xyz$, wobei gilt
      - für alle $i\geq 0$: $xy^iz \in A$
      - $|y| > 0$
      - $|xy| \leq p$.

Das Pumping-Lemma für kontextfreie Sprachen

- **Pumping-Lemma für kontextfreie Sprachen:**
  - Sei L eine kontextfreie Sprache
    - Dann existiert eine Zahl $p>0$
    - So dass für jedes Wort s mit $|s| \geq p$
    - s in fünf Teile geteilt werden kann $s = uvxyz$, wobei gilt
      - für alle $i \geq 0$: $uv^i xy^i z \in L$
      - $|vy| \geq 1$
      - $|vxy| \leq p$

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Vergleich der beiden Pumping Lemmata

- **Pumping-Lemma für reguläre Sprachen**
  - Sei A eine reguläre Sprache.
    - Dann gibt es eine Zahl p>0
    - so dass für jedes Wort s mit $|s| \geq p$
    - s in drei Teile geteilt werden kann: s = xyz, wobei gilt
      - für alle $i \geq 0$: $xy^iz \in A$
      - $|y| > 0$
      - $|xy| \leq p$.

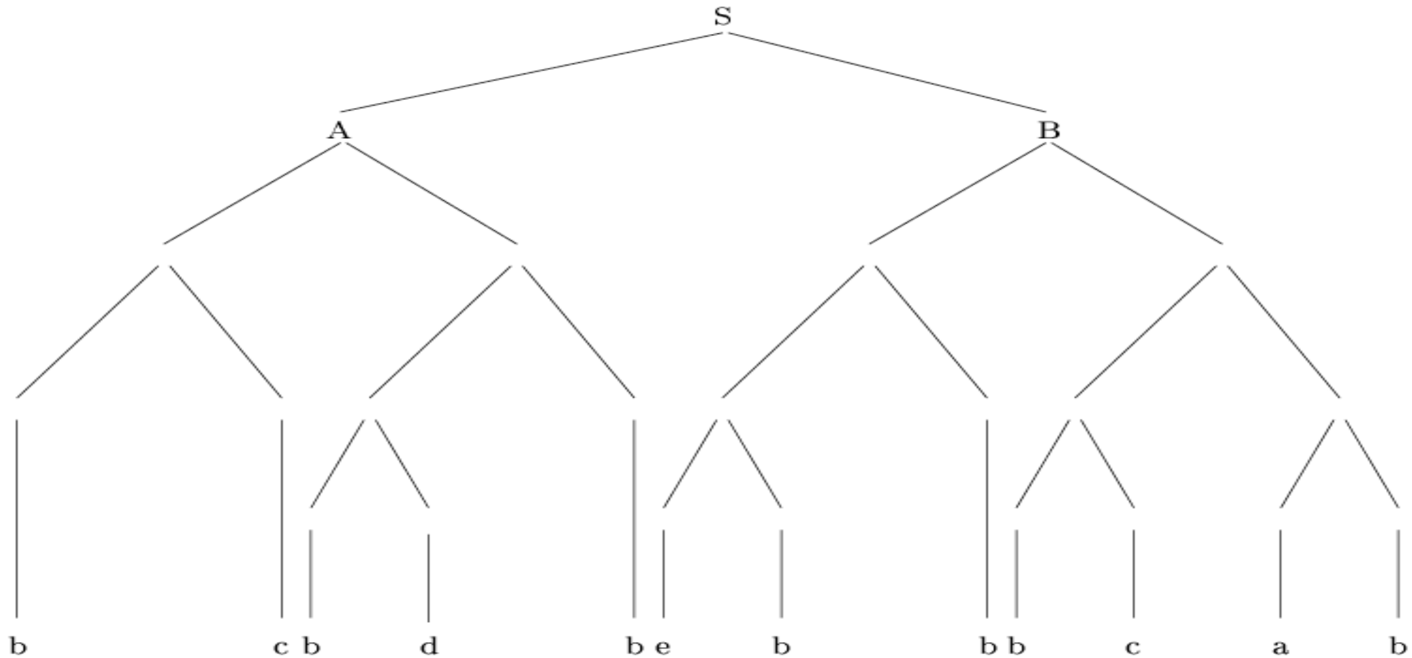
- **Pumping-Lemma für kontextfreie Sprachen:**
  - Sei L eine kontextfreie Sprache
    - Dann existiert eine Zahl p>0
    - so dass für jedes Wort s mit $|s| \geq p$
    - s in fünf Teile geteilt werden kann: s = uvxyz, wobei gilt
      - für alle $i \geq 0$: uv^ixy^iz \in L
      - $|vy| \geq 1$
      - $|vxy| \leq p$

Beweis des Pumping-Lemmas für kontextfreie Sprachen - Beweisidee

- **Verwende Chomsky-Normalform als Grammatik für L**
- **Betrachte Ableitungsbaum als Binärbaum**
- **Betrachte ein Wort w mit $|w| \geq p = 2^{|V|}$**
- **Auf dem Weg von einem Blatt zur Wurzel liegen $|V|+1$ Variablen, also kommt eine Variable doppelt vor**
- **Pumpe diesen Teilbaum auf**

Beweis des Pumping-Lemmas für kontextfreie Sprachen

- **Sei G eine Grammatik in Chomsky-Normalform, die L erzeugt**
  - Solch eine Grammatik existiert immer
- **Betrachte Ableitungsbaum eines Wortes**



- **Betrachte Ableitungsbaum als Binärbaum**
- **Setze $p=2^{|V|}$**

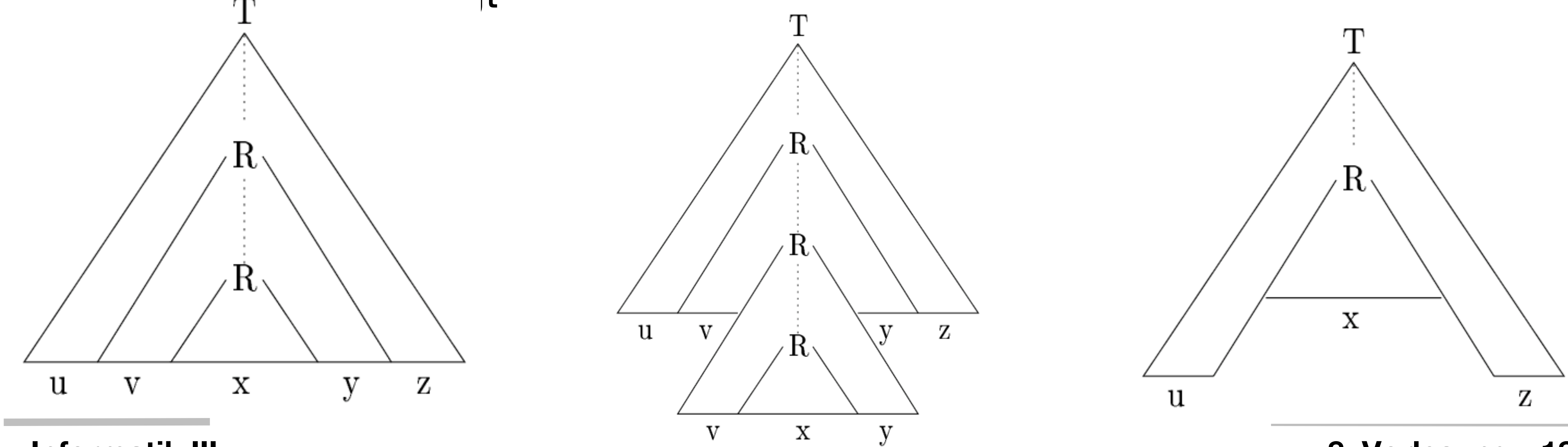
Beweis des Pumping-Lemmas für kontextfreie Sprachen

- **Betrachte Wort w der Länge p:**
- **Ein beliebiger Ableitungsbaum T für w muss nun genau |w|≥p Blätter besitzen (Binärbaum zu T muss ≥p Blätter besitzen)**
  - Binärbaum muss Tiefe mindestens |V| haben, d.h. es muss einen Pfad von der Wurzel zu einem Blatt geben, auf dem mindestens |V| Kanten liegen
  - Wähle einen Pfad der Länge mindestens |V| in dem Binärbaum des Ableitungsbaums T aus
- **Auf dem Pfad liegen mindestens |V|+1 viele Knoten mit Variablen aus V**
  - Auf dem Weg vom Blatt zur Wurzel treffen wir eine Variable R doppelt an (Schubfachprinzip)
- **Betrachte nun Teilbaum, der R als Wurzel hat**
  - Ordne dem Teilbaum das Teilwort des Wortes zu, das sich aus den Terminalen an den Blättern des Teilbaums ergibt

Beweis des Pumping-Lemmas für kontextfreie Sprachen

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Somit haben wir nun w=uvxyz**
- **Diese erfüllt**
  - $|vy| \geq 1$ (da Chomsky-Normalform)
  - $|vxy| \leq p$ (Der Teilbaum mit Wurzel R hat höchstens Tiefe $|V|$)
  - für alle $i \geq 0$: $uv^i xy^i z \in A$
    - i>1: Platziere an Stelle R den Teilbaum mit Wurzel R und erhalte wieder korrekten Ableitungsbaum
    - i=0: Ersetze den Teilbaum, der am ersten mit R bezeichneten Teilbaum hängt, durch den Teilbaum, der am zweiten mit R bezeichneten t



Beispiel: Pumping Lemma

- **Beispiel:**
  - Die Sprache $L=\{a^n b^n c^n \mid n \geq 1\}$ ist nicht kontextfrei.
- *Beweis*:
  - Sei $p>0$ beliebig
  - Wähle $w= a^p b^p c^p \in L$
  - Sei $w=uvxyz$ eine Aufteilung von w in fünf Teile mit
    - $|vy|\geq 1$
    - $|vxy|\leq p$
    - Aus $|vxy|\leq p$ folgt, daß nur zwei der drei Symbole enthalten sein können
    - Führe Fallunterscheidung basierend auf fehlendem Buchstaben durch

Fallunterscheidung

- **a‘s fehlen:**
  - Betrachte das Wort $w=uv^0xy^0z$
  - da $|vy|\geq 1$ fehlen nun einige der Symbole b,c, die in w auftauchen
  - vy enthält kein a, daher fehlt kein a
  - damit enthält $w=uv^0xy^0z$ mehr a‘s als b‘s oder c‘s und liegt nicht in L

- **b‘s fehlen:**
  - Betrachte das Wort $w=uv^2xy^2z$
  - Da $|vy|\geq 1$ taucht das Symbol a oder das Symbol c in vy auf
  - damit enthält $w=uv^2xy^2z$ mehr a‘s oder c‘s als b‘s und liegt nicht in L

- **c‘s fehlen:**
  - Betrachte das Wort $w=uv^2xy^2z$
  - Da $|vy|\geq 1$ taucht das Symbol a oder das Symbol b in vy auf
  - damit enthält $w=uv^2xy^2z$ mehr a‘s oder b‘s als c‘s und liegt nicht in L

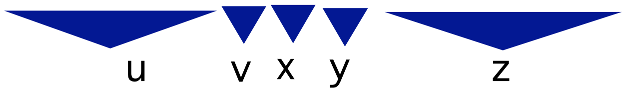
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Fehlversuch: Pumping Lemma

- **Beispiel:**
  - –Die Sprache L={ w w| w ∈ {0,1}*} ist nicht kontextfrei.
- *Pum*:
  - –Sei L eine kontextfreie Sprache
    - Dann existiert eine Zahl $p>0$
    - so dass für jedes Wort s mit $|s|\geq p$
    - s in fünf Teile geteilt werden kann: $s = uvxyz$, wobei gilt
      - für alle $i\geq 0$: uv^ixy^iz \in L
      - $|vy|\geq 1$
      - $|vxy|\leq p$

- **Betrachte:**
- **w = 0......0 1 0......0 1**
  - – also $w = 0^p10^p1$
- **Jetzt führt die Wahl:**

0000......0 0 1 0 0000......001



- **zu keinem Widerspruch!**
- **So geht das nicht.**

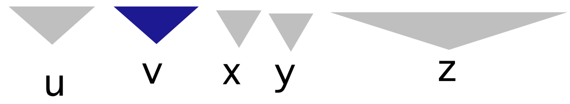
Beispiel
Pumping Lemma

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Beispiel:**
  - Die Sprache L={ w w| w ∈ {0,1}*} ist nicht kontextfrei.
- *Pum*:
  - Sei L eine kontextfreie Sprache
    - Dann existiert eine Zahl p>0
    - so dass für jedes Wort s mit $|s| \geq p$
    - s in fünf Teile geteilt werden kann: s = uvxyz, wobei gilt
      - für alle $i \geq 0$: uv^ixy^iz \in L
      - $|vy| \geq 1$
      - $|vxy| \leq p$

- **Betrachte:**
- **w = 0...0 1...1 0...0 1...1**
  - also $w = 0^p1^p0^p1^p$

00000 11111 00000 11111

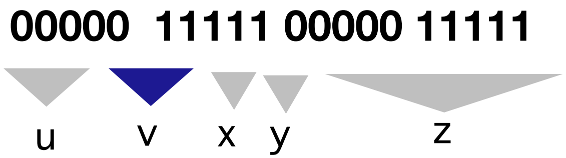


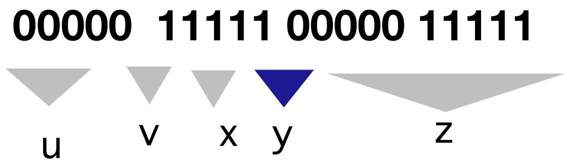
00000 11111 00000 11111

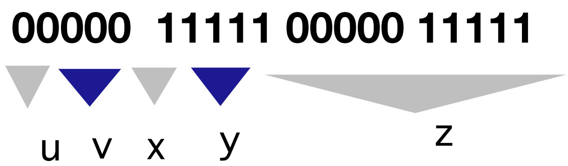
Beispiel Pumping Lemma

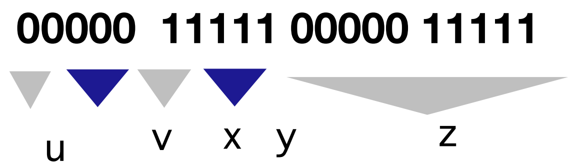
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **L={ w w| w ∈ {0,1}*}**
- **w = 0…0 1…1 0…0 1…1**
  - also $w = 0^p1^p0^p1^p$









Ende der 8. Vorlesung

Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Informatik III
Christian Schindelhauer
schindel@informatik.uni-freiburg.de