Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Automatentheorie und Formale Sprachen

– Pushdown-Automaten –

Prof. Dr. Michael Neitzke

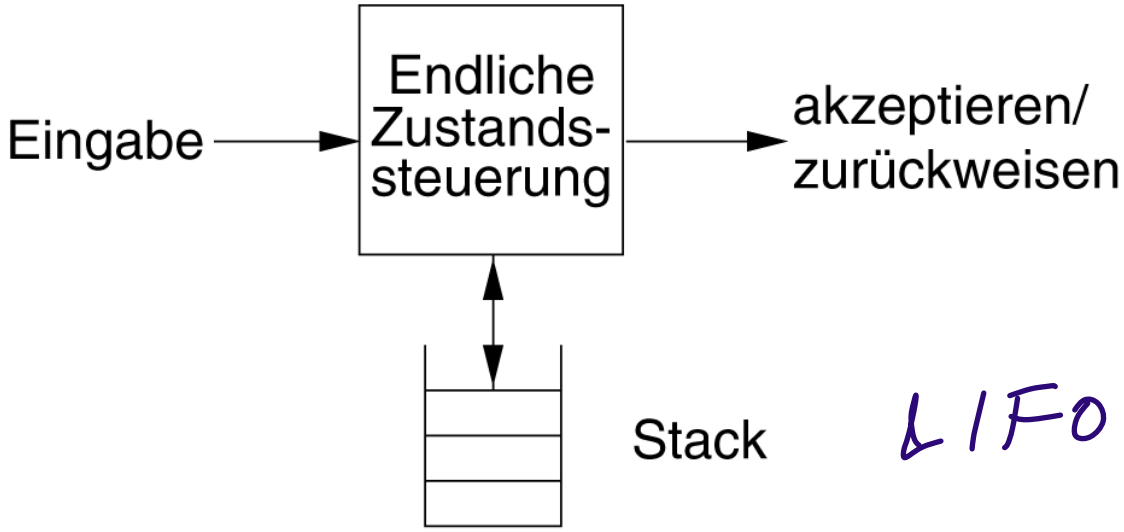
PDA1: Pushdown-Automaten

Diskussion

Denken Sie an die Einschränkungen von regulären Sprachen. Welche Möglichkeit fehlt den endlichen Automaten?

Pushdown-Automaten

- Automaten für kontextfreie Sprachen: Pushdown-Automaten
- Deutsche Bezeichnung: "Kellerautomat"
- ε-NEA mit Stack
  - Kein wahlfreier Zugriff auf Speicher!



Diskussion

Wie müsste ein Pushdown-Automat formal beschrieben werden?

Pushdown-Automaten: Definition

Ein PDA wird durch ein 7-Tupel beschrieben:

- **$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$**
  - Q Endliche Menge von Zuständen
  - Σ Endliches Eingabe-Alphabet
  - Γ Endliches Stack-Alphabet
  - δ Übergangsfunktion $\delta : Q \times (\Sigma \cup \varepsilon) \times \Gamma \rightarrow P(Q \times \Gamma^*)$
  - $q_0 \in Q$ Startzustand
  - $Z_0 \in \Gamma$ Startsymbol des Stacks
  - $F \subseteq Q$ Menge von Endzuständen

Nicht-Deterministisch hier

ε-Übergänge bedeuten Nichtdeterminismus

Arbeitsweise Pushdown-Automat

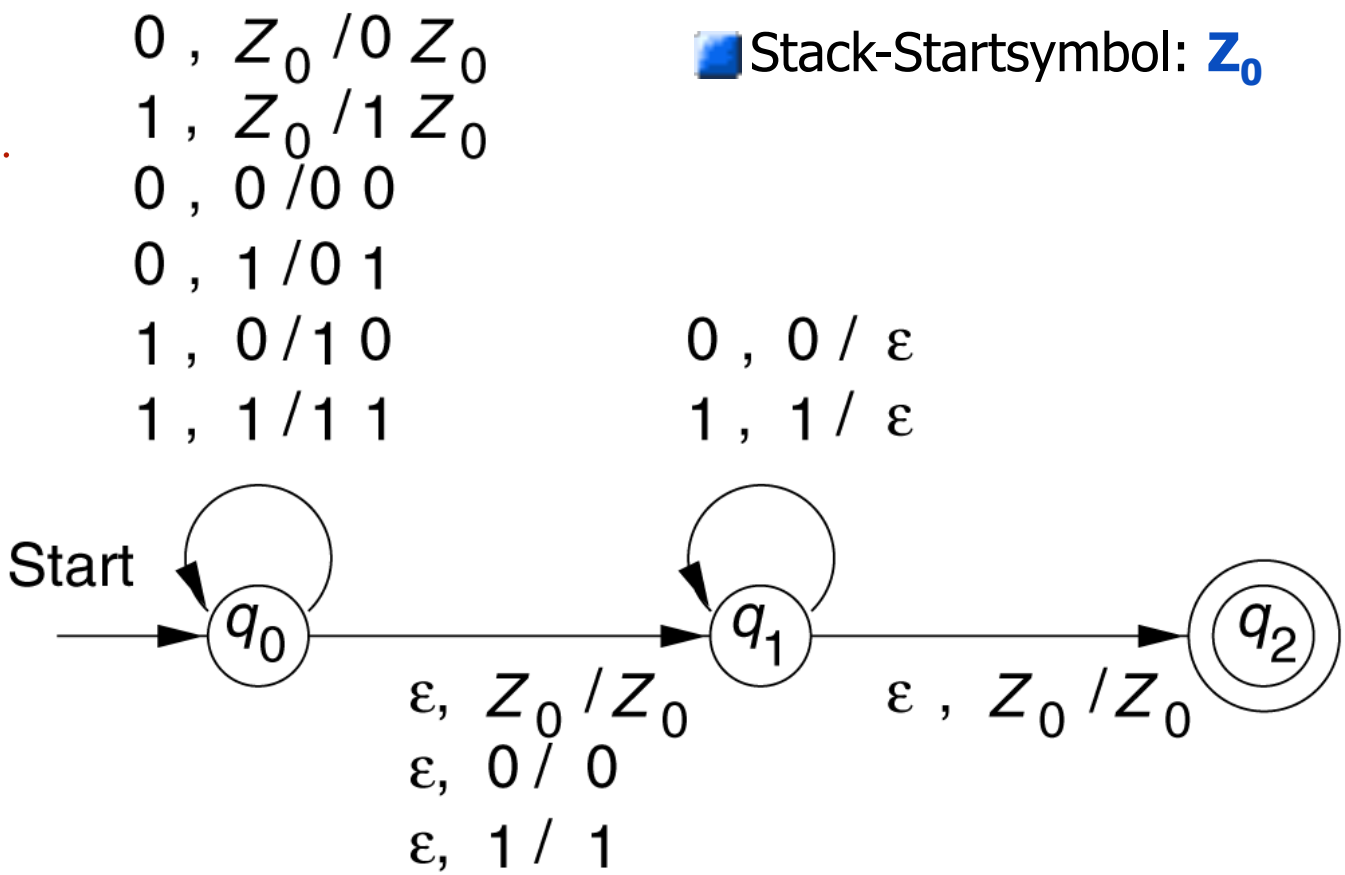
- PDA befindet sich immer in einer Menge von Zuständen
- Übergang hängt ab von
  - Zustand
  - Eingabesymbol
  - Oberstem Symbol des Stacks
- Übergangsfunktion legt fest:
  - Nachfolge-Zustand (oder mehrere oder Keiner)
  - Zeichenreihe aus Stacksymbolen, die oberstes Stacksymbol ersetzen
- Akzeptanz
  - entweder ✓ Durch Endzustand
  - Oder: durch leeren Stack
  - Beide Automatentypen sind äquivalent!

Typ B

Typ A (s. Folie 9)

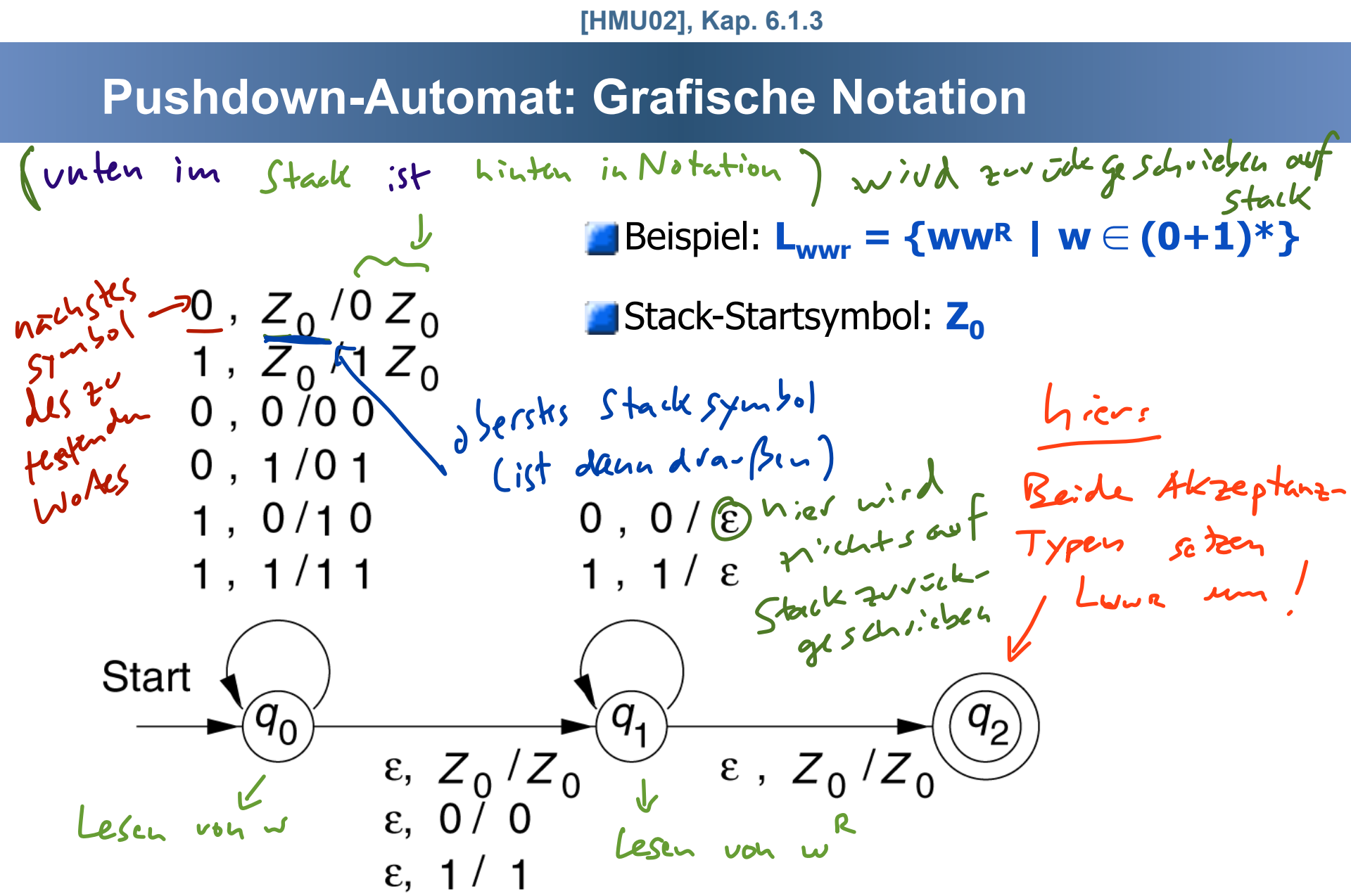
Pushdown-Automat: Grafische Notation

- Beispiel: $L_{wwr} = \{ww^R \mid w \in (0+1)^*\}$
- Stack-Startsymbol: Z_0



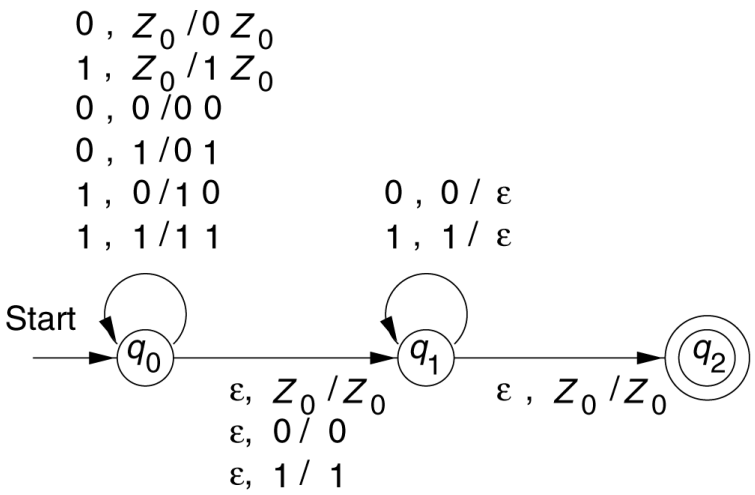
Pushdown-Automat: Grafische Notation

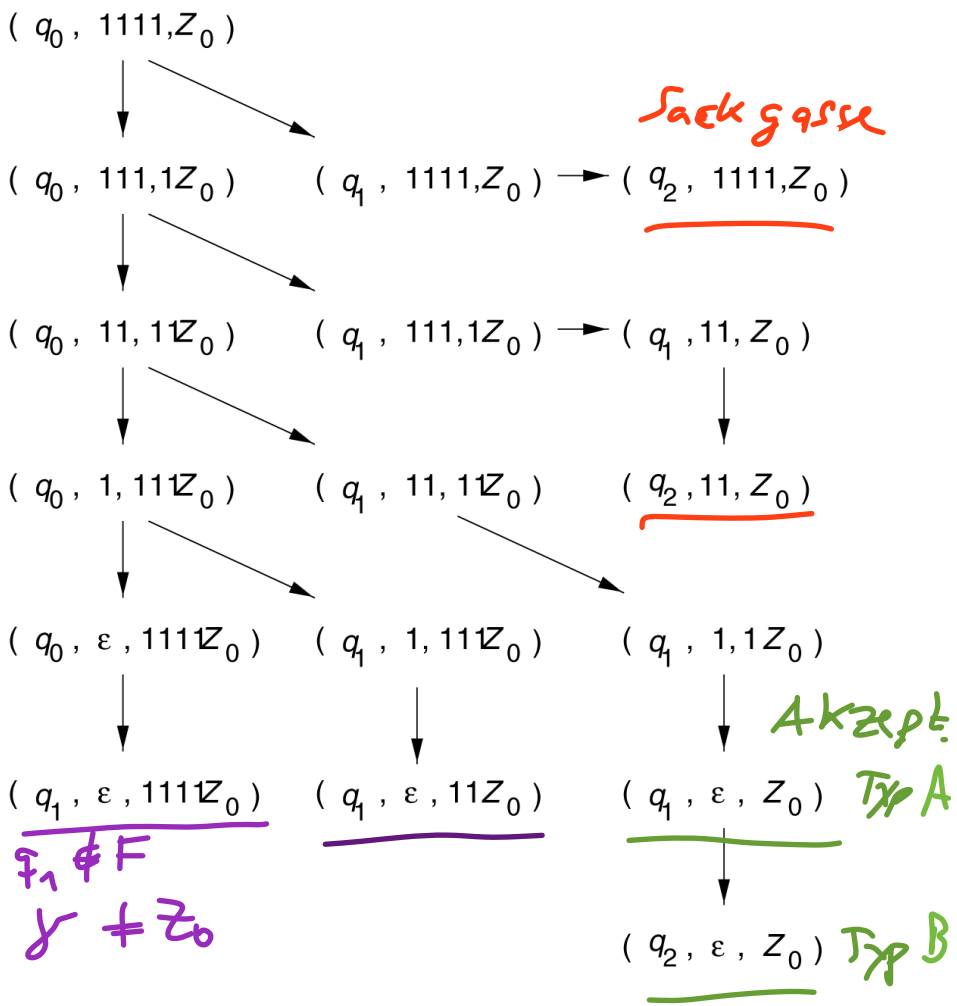
- Beispiel: $L_{wwr} = \{ww^R \mid w \in (0+1)^*\}$
- Stack-Startsymbol: Z_0



Konfigurationen eines Pushdown-Automaten

- **Konfiguration** ist Tripel (q, w, γ)
  - q : Zustand
  - w : restliche Eingabe
  - γ : Inhalt des Stack
- Beispiel:

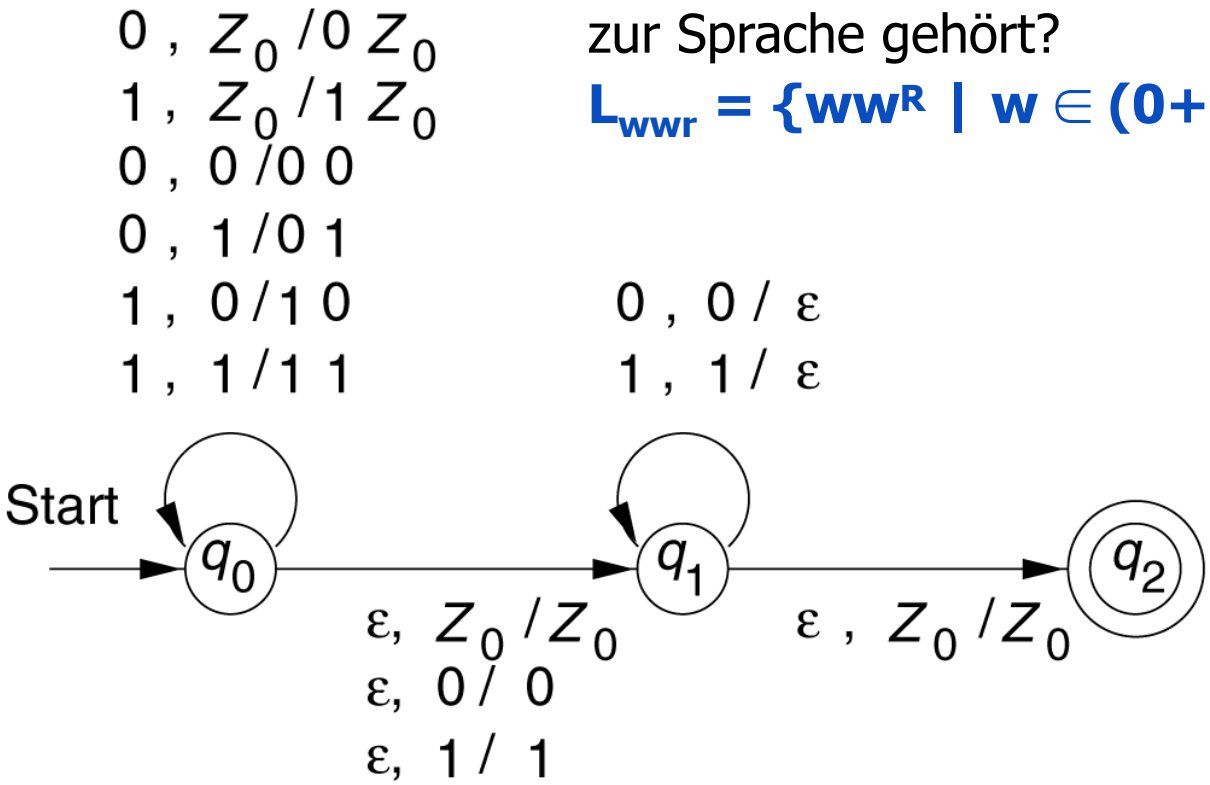




Diskussion

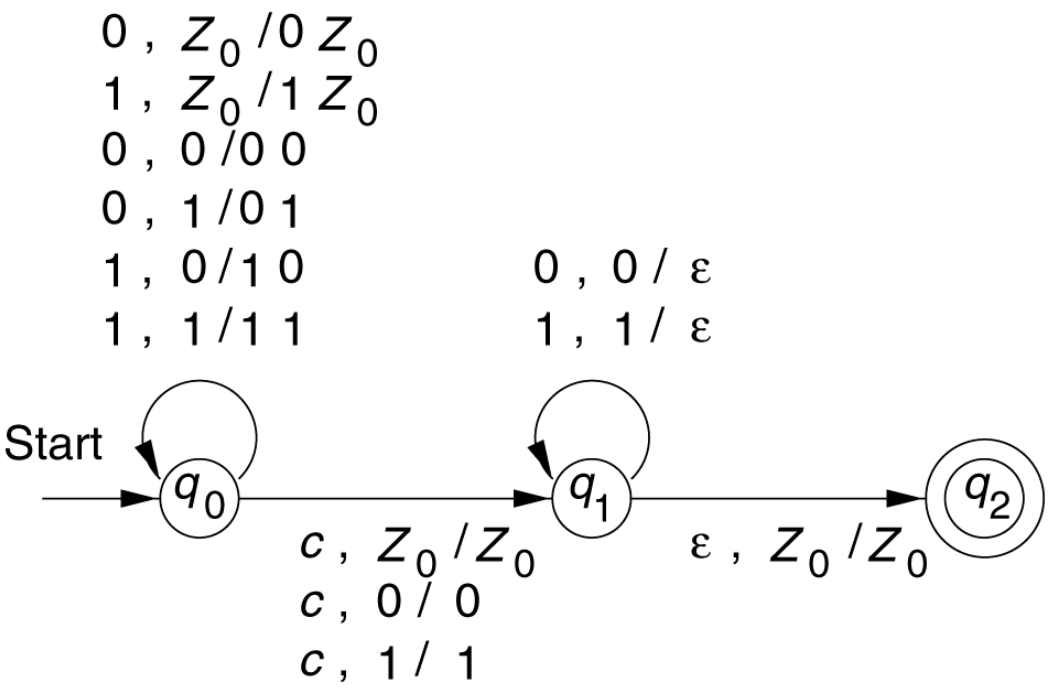
Machen Sie sich noch einmal klar, wie der PDA für die Sprache L_{wwr} arbeitet. Wie könnte ein deterministischer PDA feststellen, ob ein Wort zur Sprache gehört?

$L_{wwr} = \{ww^R \mid w \in (0+1)^*\}$



Deterministische Pushdown-Automaten

- Deterministisch heißt: Keine Situation mit mehreren Möglichkeiten
- Schwächer als nichtdeterministische PDA!
- Beispiel: Es existiert kein deterministischer PDA für L_{wwr}
  - Nicht klar, wann w^R beginnt
  - Aber: L_{wcwr} kann erkannt werden (spezielles Symbol als Mittelmarkierung)



Diskussion: Sprachklassen

- Wie verhalten sich die Sprachklassen determistischer Pushdown-Automaten (**DPDA**) und nicht-deterministischer Pushdown-Automaten (**NPDA**) zueinander?
- Welche Beziehung gibt es zu den regulären Sprachen?

Eigenschaften eines DPDA

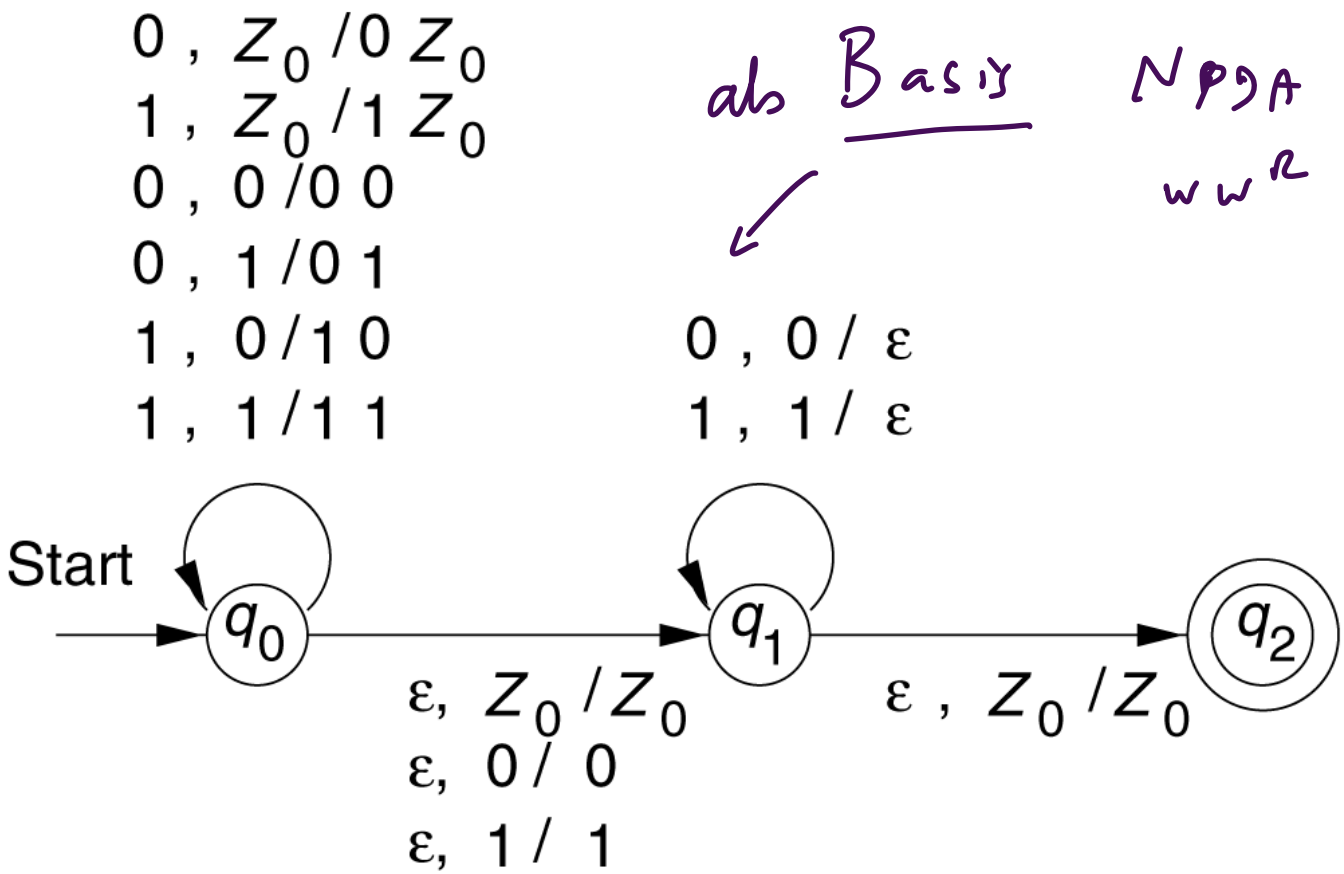
- Sprachen der DPDAs liegen zwischen kontextfreien und regulären Sprachen
  - Echte Teilmenge der kontextfreien Sprachen
  - Echte Obermenge der regulären Sprachen
- Die Sprache eines jeden DPDAs verfügt über eine eindeutige Grammatik.
- Aber: Nicht für jede **eindeutige kontextfreie Sprache** gibt es einen DPDA
  - Beispiel $\mathbf{L_{wwr}}$ **: S → 0S0 | 1S1 |** ε

informell:

$$L(DEA) \subsetneq L(DPDA) \subsetneq L(eKFG) \subsetneq L(KFG) = L(NPDA)$$

Übung

Entwerfen Sie einen PDA für die Sprache $\{0^n1^n \mid n \geq 1\}$.



Übung

- Beispiel: $\{0^n1^n \mid n \geq 1\}$
- Stack-Startsymbol: Z_0

0 , Z_0 /0 Z_0

~~1 , Z_0 /1 Z_0~~

0 , 0 /0 0

~~0 , 1 /0 1~~

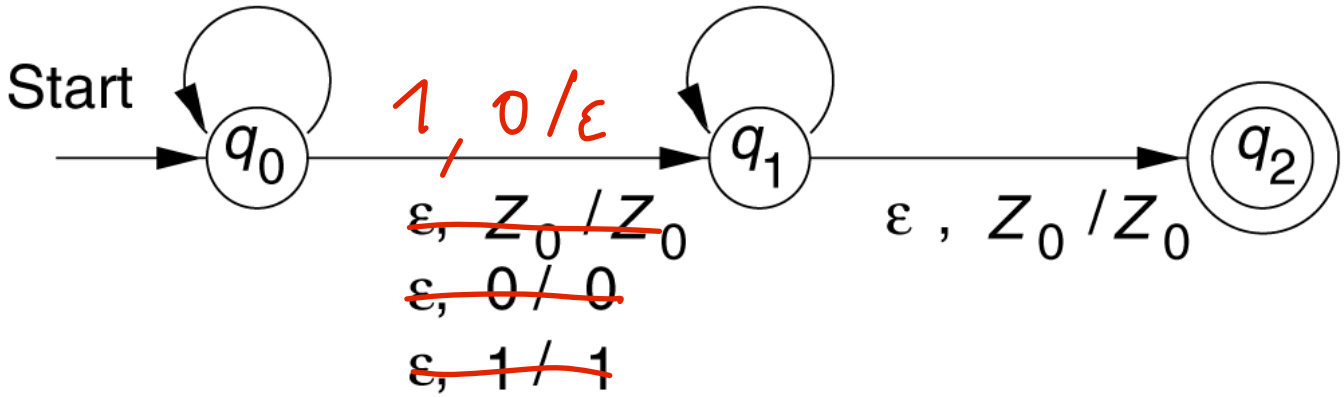
~~1 , 0 /1 0~~

~~1 , 1 /1 1~~

1, 0 /ε

~~0 , 0 / ε~~

~~1 , 1 / ε~~



Transformation KFG → PDA

- Jede KFG kann in einen PDA transformiert werden, und zwar in einen NPDA
- Grammatik muss zunächst in die Greibach Normalform gebracht werden

Transformation GNF → NPDA

- Folgende Grammatik ist in Greibach-Normalform:

$$S \to aB \mid bA$$
$$A \to a \mid aS \mid bAA$$
$$B \to b \mid bS \mid aBB$$

- NPDA benötigt nur einen wesentlichen Zustand
  - Übergänge des NPDA:

a, S/B
b, S/A
a, A/ε
a, A/S
b, A/AA
b, B/ε
b, B/S
a, B/BB

Ende V9

- **Akzeptiert der NPDA durch Endzustand oder leeren Stack?**
- **Welche Sprache akzeptiert er?**

PDA1: Rückblick auf die Ziele

	PDA1
Kenntnis	- Definitionen der rot hervorgehobenen Begriffe kennen
Verständnis	- PDAs formal beschreiben können und vorgegebene Modelle erklären können. - PDAs (und zwar DPDAs und NPDAs) simulieren können. - Die Beziehungen zwischen den Sprachklassen regulärer Sprachen, kontextfreier Sprachen, eindeutiger Sprachen, der Sprachen von DPDAs und NPDAs erläutern und begründen können
Anwendung	- Eine gegebene KFG in einen NPDA transformieren können
Analyse Synthese	- PDAs auf Basis einer formalen Beschreibung oder einer Anwendungssituation entwickeln können
Beurteilung	