

6. プッシュダウン・オートマトン:

6.1. プッシュダウン・オートマトン(PDA)の定義

6.2. PDAの言語

6.3. PDAとCFGの等価性

(6.4. 決定性プッシュダウン・オートマトン)

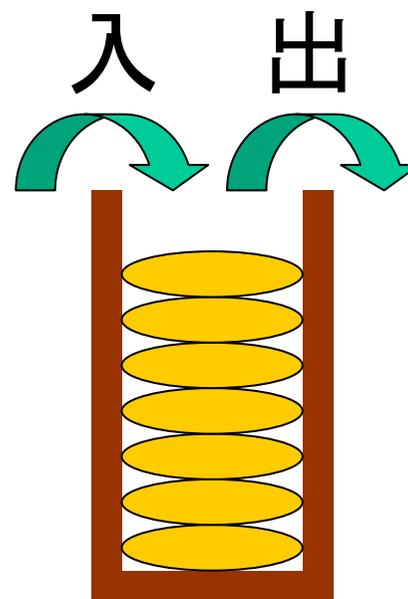
6.1. プッシュダウン・オートマトン (PDA)の定義

プッシュダウン・オートマトン(PDA) =
オートマトン + プッシュダウンスタック

プッシュダウンスタック...

- 食堂のお盆
- (最近見ない)コインケース
- いわゆる stack

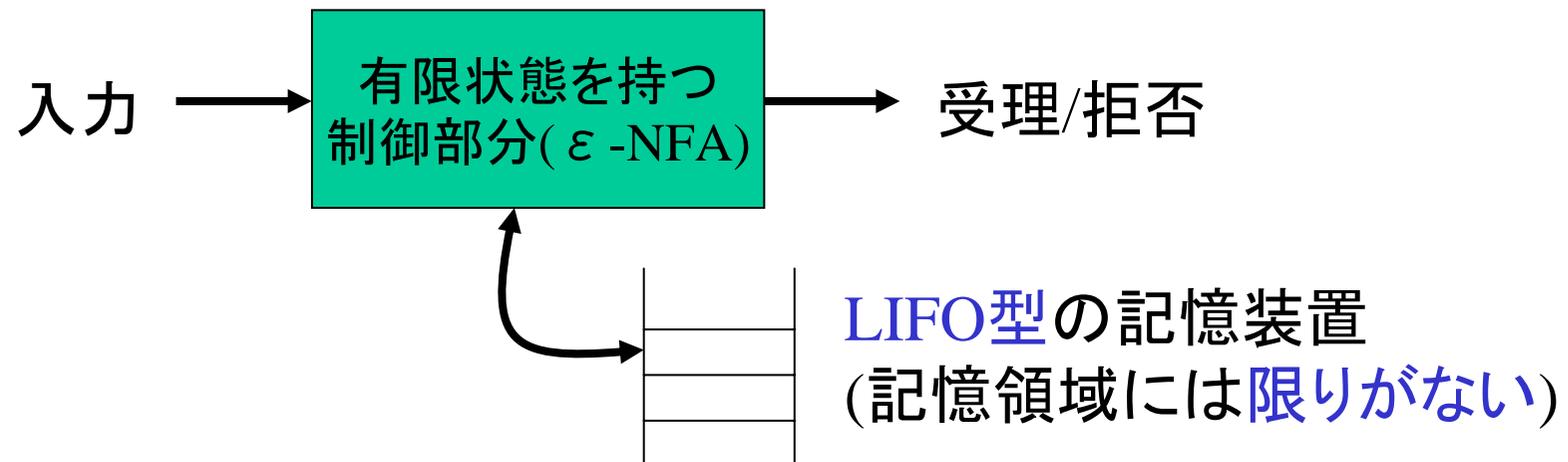
LIFO: Last In First Out



6.1. プッシュダウン・オートマトン (PDA)の定義

6.1.1. 直感的な説明

PDA とは ϵ -NFA が stack を一つ持った機械モデル



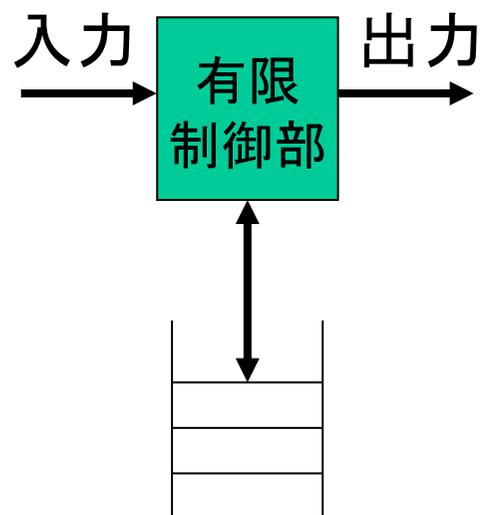
6.1. プッシュダウン・オートマトン (PDA)の定義

受理条件:

1. 受理状態
2. スタックが空

6.1.1. 直感的な説明

PDA とは ϵ -NFA が stack を一つ持った機械モデル



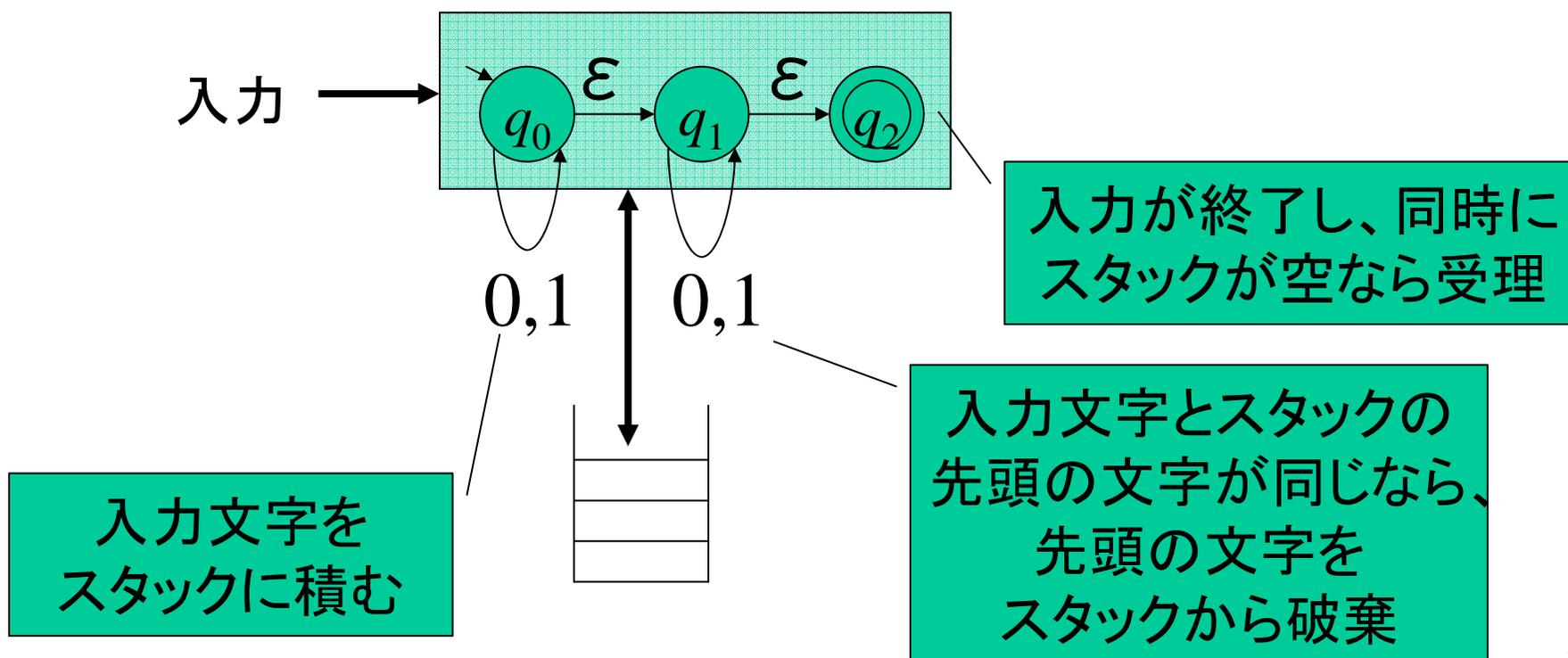
動作プロセス:

1. 入力を1文字読む
 ϵ -動作のときは0文字
2. 入力 x と スタックの一番上の文字 y に応じて状態遷移する
3. スタックを操作する:
 1. 一番上の文字を取り出して捨てる
 2. 一番上の文字を書き換える
 3. 2に加えて、いくつかの文字を記憶
4. 入力が終わって 受理条件 を満たしていたら受理。入力が残っているならステップ1へ

6.1. プッシュダウン・オートマトン (PDA)の定義

6.1.1. 直感的な説明

例) $L = \{ ww^R \mid w \in \{0,1\}^* \}$ を受理する PDA

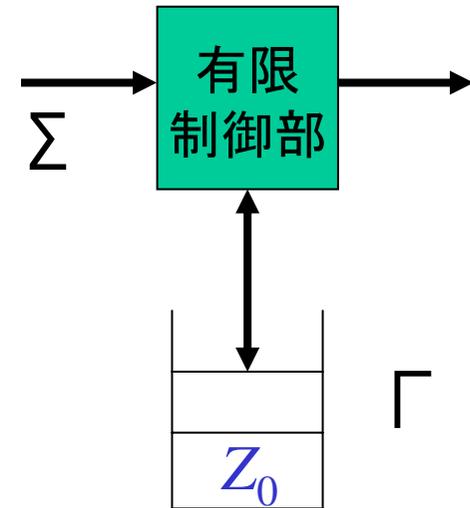


6.1. プッシュダウン・オートマトン (PDA)の定義

6.1.2. PDA の形式的定義

PDA $P=(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

- Q : 状態の有限集合
- Σ : 入力アルファベット
- q_0 : $q_0 \in Q$ を満たす初期状態
- F : $F \subseteq Q$ を満たす受理状態
- Γ : スタックアルファベット; スタックに記憶する文字集合
- Z_0 : $Z_0 \in \Gamma$ を満たす開始記号; スタックには最初にこの文字が1つ入っているとす。 (スタックの[底]を判定するための便宜上の文字)



6.1. プッシュダウン・オートマトン (PDA)の定義

6.1.2. PDA の形式的定義

$$\text{PDA } P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

$$\text{関数 } \delta : Q \times \Sigma \times \Gamma \rightarrow 2^Q \times \Gamma^*$$

– δ は「現在の状態」「入力1文字」「スタックの
トップの文字 X 」が与えられ、

– 「次の状態」「 X を置き換える文字列 Y 」を返す
関数

• 「入力1文字」は ε も可

• Y は右が可:

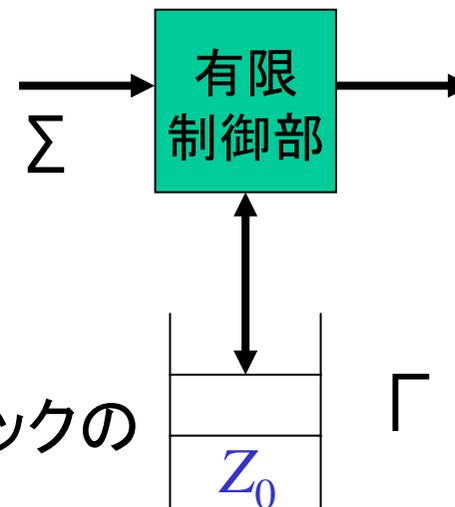
• $Y = \varepsilon$; X の破棄

• Y が1文字; X の置換

• Y が2文字以上; 置換 + 追加

– δ は非決定的なので、

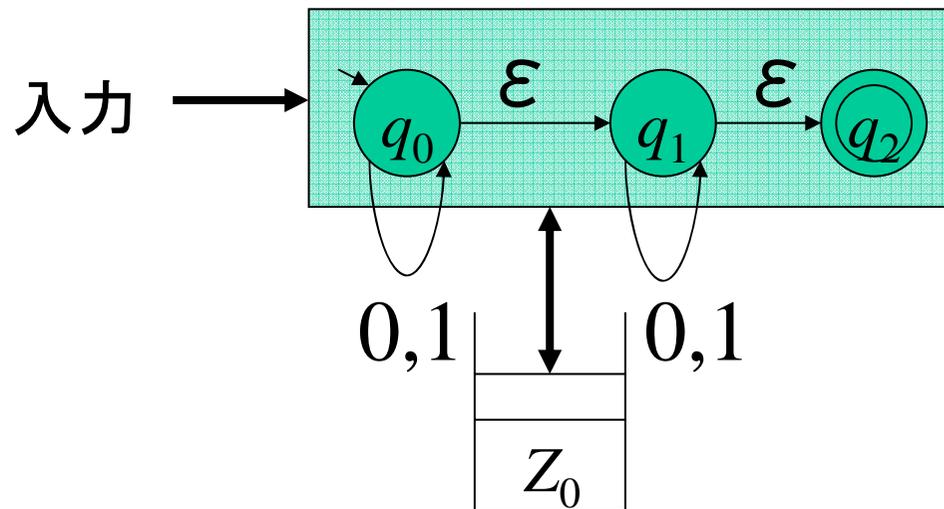
同じ入力に対する行き先が複数ありえる



6.1. プッシュダウン・オートマトン (PDA)の定義

6.1.2. PDAの形式的な定義

例) $L = \{ ww^R \mid w \in \{0,1\}^* \}$ を受理する PDA P

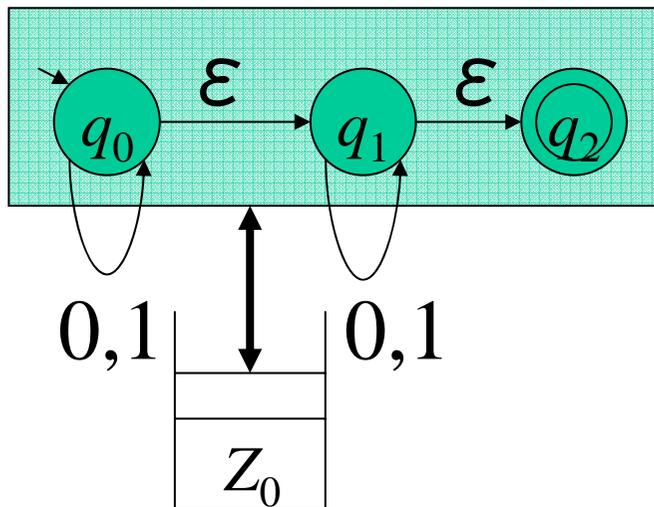


$$P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, \delta, q_0, Z_0, \{q_2\})$$

6.1. プッシュダウン・オートマトン (PDA)の定義

6.1.2. PDAの形式的な定義

例) $L = \{ ww^R \mid w \in \{0,1\}^* \}$ を受理する PDA P



q_0 に関する δ

- 入力をスタックに積む

$$\delta(q_0, 0, Z_0) = \{(q_0, 0Z_0)\}, \delta(q_0, 1, Z_0) = \{(q_0, 1Z_0)\}$$

$$\delta(q_0, 0, 0) = \{(q_0, 00)\}, \delta(q_0, 1, 0) = \{(q_0, 10)\}$$

$$\delta(q_0, 0, 1) = \{(q_0, 01)\}, \delta(q_0, 1, 1) = \{(q_0, 11)\}$$

- ϵ 動作で q_1 に遷移

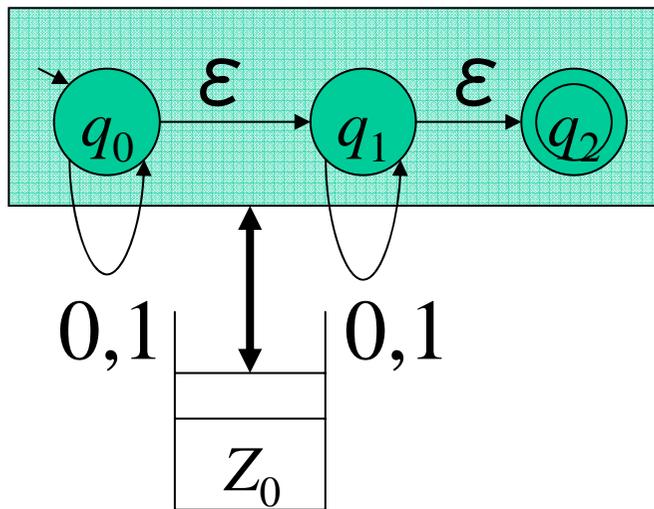
$$\delta(q_0, \epsilon, Z_0) = \{(q_1, Z_0)\}, \delta(q_0, \epsilon, 0) = \{(q_1, 0)\}, \delta(q_0, \epsilon, 1) = \{(q_1, 1)\}$$

$$P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, \delta, q_0, Z_0, \{q_2\})$$

6.1. プッシュダウン・オートマトン (PDA)の定義

6.1.2. PDAの形式的な定義

例) $L = \{ ww^R \mid w \in \{0,1\}^* \}$ を受理する PDA P



q_1 に関する δ

- 入力とスタックのトップを比較

$$\delta(q_1, 0, 0) = \{(q_1, \epsilon)\}$$

$$\delta(q_1, 1, 1) = \{(q_1, \epsilon)\}$$

- 入力とスタックのトップがどちらも空になったら ϵ 動作で q_2 に遷移

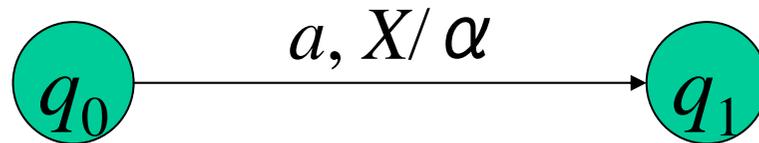
$$\delta(q_1, \epsilon, Z_0) = \{(q_2, Z_0)\}$$

$$P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, \delta, q_0, Z_0, \{q_2\})$$

6.1. プッシュダウン・オートマトン (PDA)の定義

6.1.3. PDAの図による表現

- 関数 δ を辺のラベルで表現する

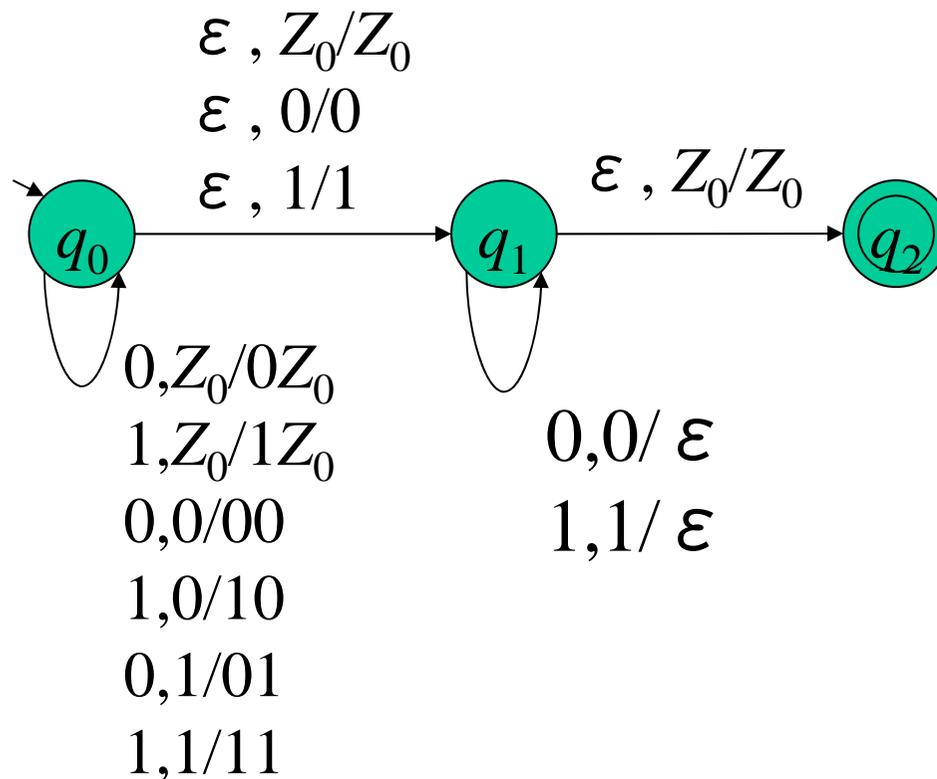


$\delta (q_0, a, X) = \{ \dots, (q_1, \alpha), \dots \}$ の図示

6.1. プッシュダウン・オートマトン (PDA)の定義

6.1.3. PDAの図による表現

例) $L = \{ ww^R \mid w \in \{0,1\}^* \}$ を受理する PDA P の δ



6.1. プッシュダウン・オートマトン (PDA)の定義

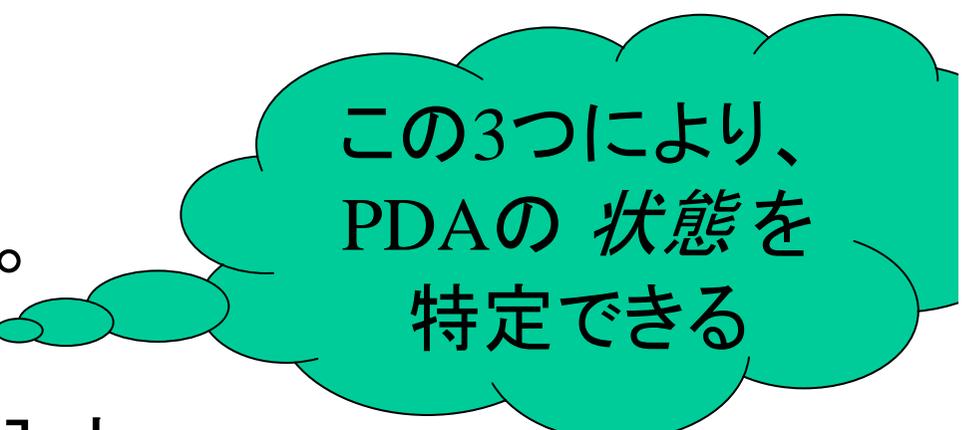
6.1.4. PDAの状況の関係

- オートマトンは「状態」だけで特定できた
- PDAでは「状態」+「スタックの文字列」でないと状態が特定できない

⇒ $\hat{\delta}$ 記法は適切でない

PDAの**状況**とは (q, w, γ) 。

- ただし
- $q \in Q$: 状態
 - $w \in \Sigma^*$: 残りの入力
 - $\gamma \in \Gamma^*$: スタックの文字列



この3つにより、
PDAの**状態**を
特定できる

状況は**時点表示**, ID (Instantaneous Description)とも言う

6.1. プッシュダウン・オートマトン (PDA)の定義

6.1.4. PDAの状況の関係

- PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ において
 $\delta(q, a, X)$ が (p, α) を含むなら、 $w \in \Sigma^*$, $\beta \in \Gamma^*$
に対して

$$(q, aw, X\beta) \vdash_P (p, w, \alpha\beta)$$

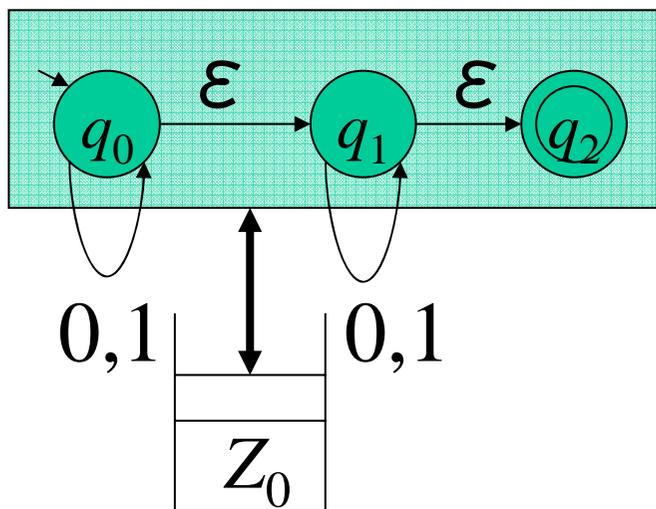
PDAの1ス
テップを表現

- と書く(P がわかっているなら \vdash と書く)。
- 0ステップ以上の一般の動作を表現するときは \vdash^* を使う:
 1. 状況 I に対し $I \vdash^* I$
 2. 状況 I, J, K に対し $I \vdash J$ & $J \vdash^* K$ なら $I \vdash^* K$

6.1. プッシュダウン・オートマトン (PDA)の定義

6.1.4. PDA の状況の関係

例) $L = \{ ww^R \mid w \in \{0,1\}^* \}$ を受理する PDA



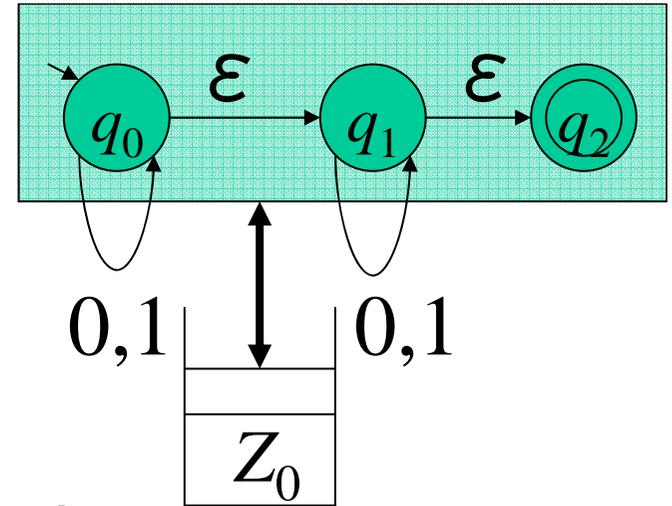
入力101101に対する状況の遷移:

$$(q_0, 101101, Z_0) \vdash (q_0, 01101, 1Z_0)$$

$$(q_0, 101101, Z_0) \vdash (q_1, 101101, Z_0)$$

$$(q_0, 101101, Z_0) \begin{cases} \rightarrow (q_0, 01101, 1Z_0) \\ \rightarrow (q_1, 101101, Z_0) \end{cases}$$

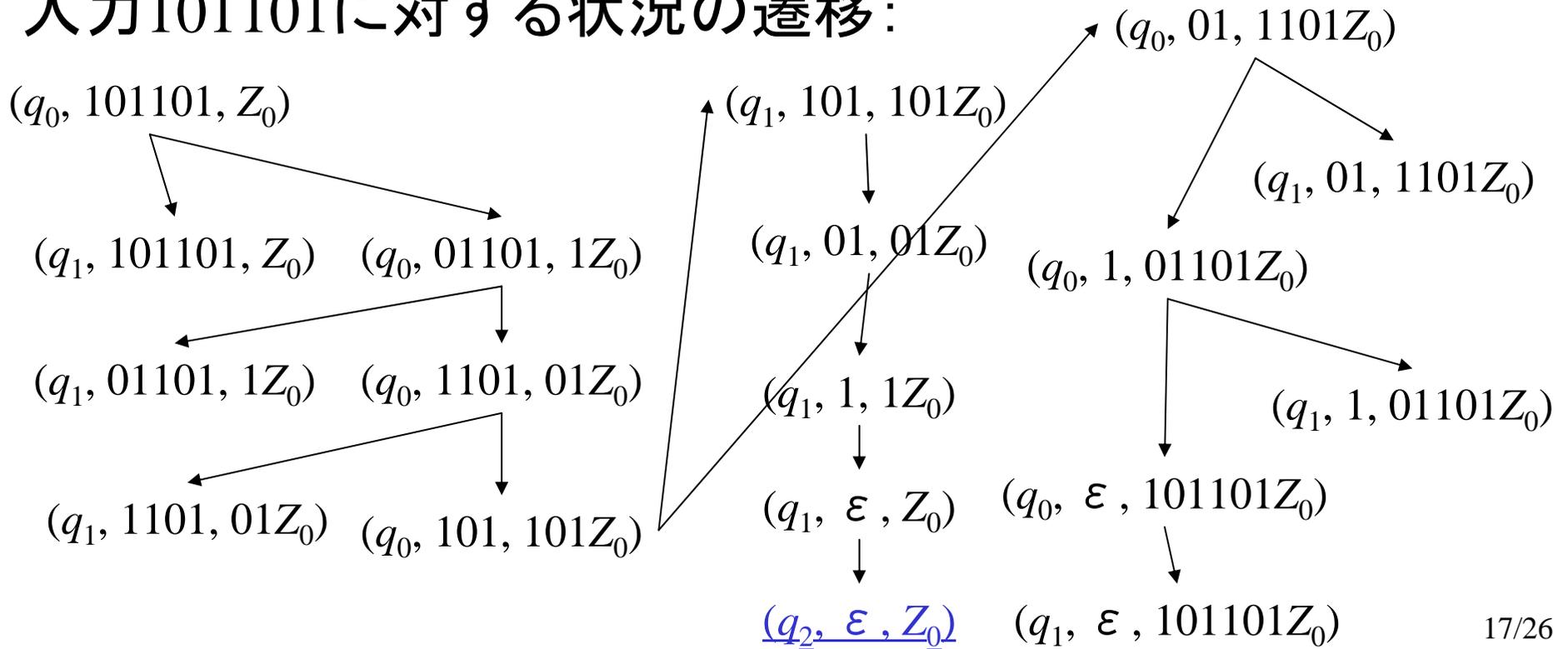
6.1. プッシュダウン・オートマトン (PDA) の定義



6.1.4. PDA の状況の関係

例) $L = \{ ww^R \mid w \in \{0,1\}^* \}$ を受理する PDA

入力 101101 に対する状況の遷移:



6.1. プッシュダウン・オートマトン (PDA)の定義

6.1.4. PDA の状況の関係

[定理] PDA P の計算プロセスを考えると、

$$(q, x, \alpha) \vdash^* (p, y, \beta) \text{ なら } (q, xw, \alpha \gamma) \vdash^* (p, yw, \beta \gamma)$$

[定理] PDA P の計算プロセスを考えると、

$$(q, xw, \alpha) \vdash^* (p, yw, \beta) \text{ なら } (q, x, \alpha \gamma) \vdash^* (p, y, \beta \gamma)$$

★『 $(q, x, \alpha \gamma) \vdash^* (p, y, \beta \gamma)$ なら $(q, x, \alpha) \vdash^* (p, y, \beta)$ 』で
はない

6.2. PDAの言語

PDAの受理状態:

1. 入力を読み終わったときに**受理状態**にある
2. 入力を読み終わったときに**スタックが空**

与えられたPDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ に対して、

$$L(P) = \{ w \mid \text{ある } q_f \in F \text{ に対し } (q_0, w, Z_0) \stackrel{*}{\vdash} (q_f, \varepsilon, \alpha) \}$$

$$N(P) = \{ w \mid \text{ある } q \in Q \text{ に対し } (q_0, w, Z_0) \stackrel{*}{\vdash} (q, \varepsilon, \varepsilon) \}$$

- $N(P)$ を考えるときは F は関係ないので、6つ組 $(Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ で記述することもある
- 日本語テキストの $N(P)$ の定義では $q \in F$ となっているが間違い

(258ページ)

6.2. PDAの言語

PDA P によって定義される2種類の言語:

1. $L(P)$: 入力を読み終わったときに**受理状態**
2. $N(P)$: 入力を読み終わったときに**スタックが空**

➤ P を固定すると一般に $L(P) \neq N(P)$

例) スタックに最後に Z_0 がいつでも残る P に対しては $N(P) = \emptyset$

[定理]

PDA Q に対して $N(Q) = L(P)$ を満たす PDA P が存在。

PDA P に対して $L(P) = N(Q)$ を満たす PDA Q が存在。

6.2. PDAの言語

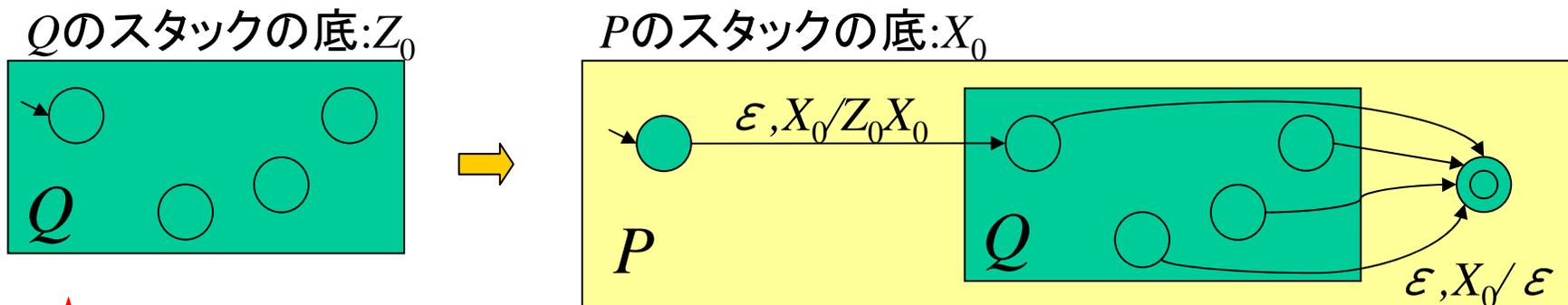
[定理]

1. PDA Q に対して $N(Q) = L(P)$ を満たす PDA P が存在。
2. PDA P に対して $L(P) = N(Q)$ を満たす PDA Q が存在。

[略証]与えられた Q から条件を満たす P を以下の通り構成

アイデア: 最初にスタックの底に特別な記号 X_0 をつむ...

「 Q の計算でスタックが空」=「 P の計算でスタックのトップが X_0 」



★ Q の入力が終了かつスタックが空のときのみ
 P が受理状態でかつ入力が終了

6.2. PDAの言語

[定理]

1. PDA Q に対して $N(Q) = L(P)$ を満たす PDA P が存在。
2. PDA P に対して $L(P) = N(Q)$ を満たす PDA Q が存在。

[略証] 与えられた P から条件を満たす Q を以下の通り構成

アイデア1: 最初にスタックの底に特別な記号 X_0 をつむ

... P の計算の模倣中に、スタックが空になることはない

アイデア2: P の受理状態から Q の受理状態に ε 動作で遷移する

... 入力を消費しないことに注意

アイデア3: Q の受理状態ではスタックの中身をすべて破棄

6.2. PDAの言語

[定理]

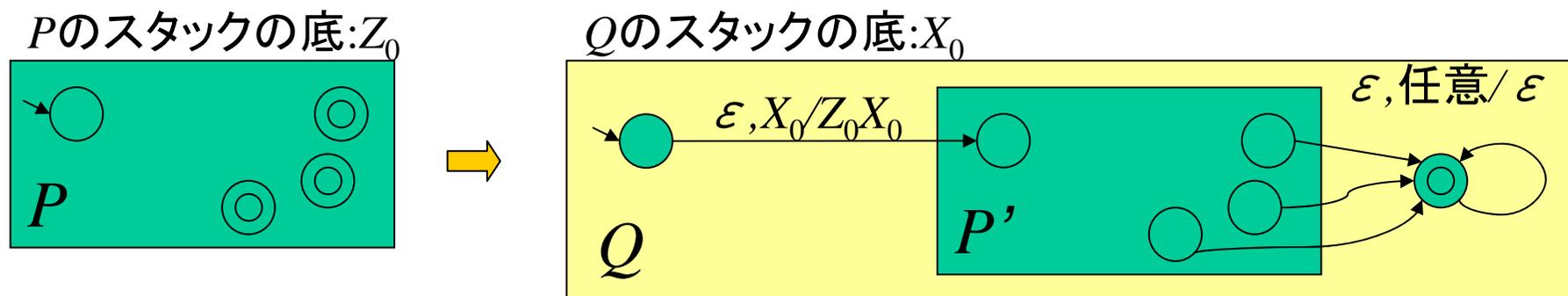
1. PDA Q に対して $N(Q) = L(P)$ を満たす PDA P が存在。
2. PDA P に対して $L(P) = N(Q)$ を満たす PDA Q が存在。

[略証] 与えられた P から条件を満たす Q を以下の通り構成

アイデア1: 最初にスタックの底に特別な記号 X_0 をつむ
... P の計算の模倣中に、スタックが空になることはない

アイデア2: P の受理状態から Q の受理状態に ϵ 動作で遷移する
... 入力を消費しないことに注意

アイデア3: Q の受理状態ではスタックの中身をすべて破棄



★ P が受理状態であつ入力が終了のときのみ
 Q の入力が終了かつスタックが空

6.3. PDA とCFGの等価性

[結論]

PDAで受理できる言語=CFGで生成できる言語

定理

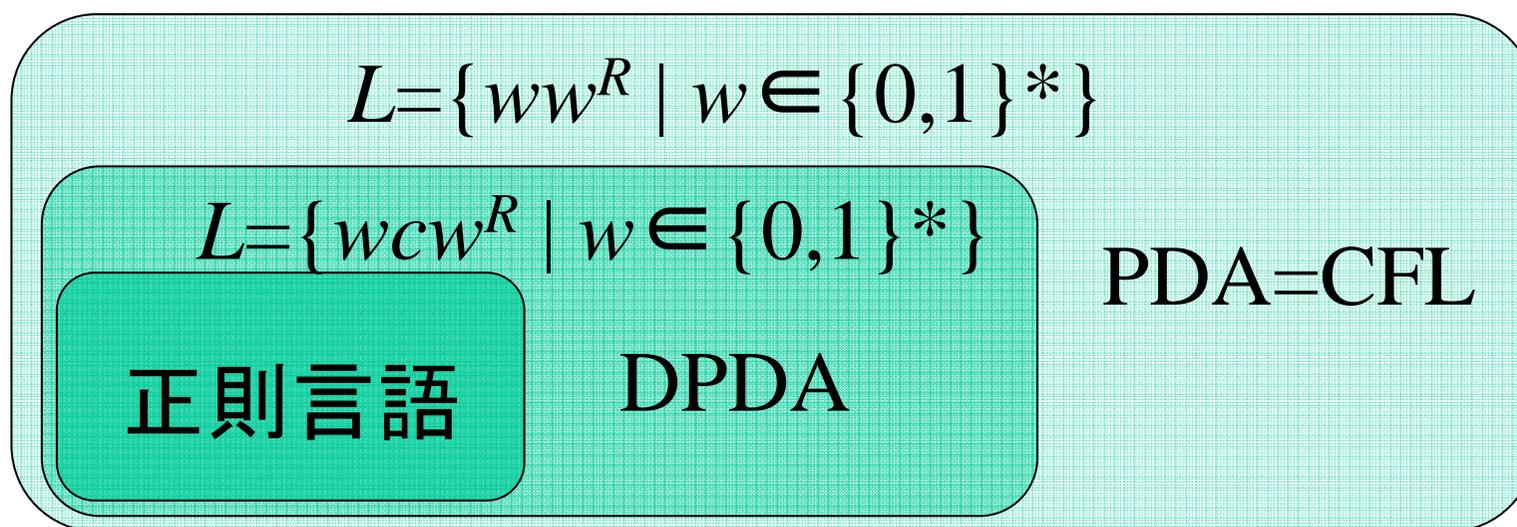
1. 任意のCFG G に対し、 $L(G)=N(P)$ を満たす PDA P が存在する。
2. 任意の PDA P に対し、 $N(P)=L(G)$ を満たす CFG G が存在する。

証明は省略
(難しくはないが、複雑)

6.4. 決定性PDA(概要)

決定性PDA: PDAにおいて、非決定性を取り除いたもの。「次の状態」が一意的に決まる。

正則言語、DPDAの言語、PDAの言語の関係:



★DPDAのクラスは実用的クラス(例;LR(k),YACC)

6. プッシュダウン・オートマトン: 演習問題(8)

[問1] 正則言語は文脈自由言語に真に含まれることを証明せよ。(集合 A が集合 B に真に含まれるとは、 $A \subseteq B$ かつ $A \neq B$ が成立するときを言う。)