演習問題:

$\Sigma = \{a, b\}$ **とする。**

次の言語を認識する有限オートマトンを構成し、状態遷移図で表せ。

(1) $A = \{a^{2n}b^{2m+1} | n, m \geq 0\}$
(a **が偶数個 (** 0 **個も可) 続いた後に、**
b **が奇数個続く)**

(2) $B = \{vabbaaw | v, w \in \Sigma^*\}$
(部分列として $abbaa$ **を含む)**

ちょっとしたコツ (tips):

「後に続く文字列が何だったら受理か」

が全く同じ状態は一つの状態にまとめられる。

これが違う状態はまとめられない。
(違う状態として用意する必要あり)

有限オートマトンでの計算可能性問題

- 言語 $A \subset \Sigma^*$ に対し、
  A を認識する有限オートマトン M
  が存在するか？

- 有限オートマトンによって
  認識可能な言語はどのようなものか？

$\longrightarrow$ 正規言語・正規表現

語の演算

語 $v = a_1 \cdots a_k, w = b_1 \cdots b_l \in \Sigma^*$ **に対し**

$$vw := a_1 \cdots a_k b_1 \cdots b_l$$

：連結・連接 (concatnation)

連接演算により Σ^* **は単位的自由半群を成す**

$$S = (S, \cdot) \text{：半群}$$

$$\Updownarrow$$

$$\cdot : S \times S \longrightarrow S \text{：二項演算で結合律を満たす}$$

正規演算

言語 $A, B \subset \Sigma^*$ **に対し、**

- $A \cup B := \{w | w \in A$ **または** $w \in B\}$
  ：和集合演算
- $AB = A \circ B := \{vw | v \in A, w \in B\}$
  ：連結 (連接) 演算
- $A^* := \{w_1 w_2 \cdots w_n | n \geq 0, w_i \in A\}$
  ：star 演算

(言語全体の集合 $\mathcal{P}(\Sigma^*)$ **上の演算)**

正規表現

- 各 alphabet $a \in \Sigma$ は正規表現
- 空列 ε は正規表現
- 空集合 $\emptyset$ は正規表現
- 正規表現 R, S に対し
  $(R \cup S)$ は正規表現 ($(R|S)$ とも書く)
- 正規表現 R, S に対し
  $(R \circ S)$ は正規表現 ((RS) とも書く)
- 正規表現 R に対し R^* は正規表現
- 以上のものだけが正規表現

$\cdots$ 帰納的導出による定義

正規表現 R に対し、言語 $L(R)$ を次で定める。

- $L(a) = \{a\}$
- $L(\varepsilon) = \{\varepsilon\}$
- $L(\emptyset) = \emptyset$
- $L(R \cup S) = L(R) \cup L(S)$
- $L(R \circ S) = L(R) \circ L(S)$
- $L(R^*) = L(R)^*$

正規表現で表される言語 $\cdots$ 正規言語

定理:

L : **正規言語**

$\Updownarrow$

L **が或る有限オートマトンで認識される**

このような一般論を考えるには、
有限オートマトンの概念を
少し一般化する方が良い。

. . . 非決定性有限オートマトン

非決定性 ··· あてずっぽう有り

状態の遷移先を一意に決めない
(幾つかあって分岐していく)

$\longrightarrow$ どれかが受理すれば OK!!

どの道を辿れば良いか知っていて、
受理が検証出来れば良い、

と考えることも出来る。

非決定性有限オートマトンの形式的定義

$$M = (Q, \Sigma, \delta, s, F)$$

ここに、

- Q：有限集合 $\cdots$ 状態の集合
- Σ：有限集合 $\cdots$ alphabet
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$：遷移関数
  $\cdots$ 可能な遷移先全体の集合
- $s \in Q$ $\cdots$ 初期状態
- $F \subset Q$ $\cdots$ 受理状態の集合

非決定性有限オートマトン

$$M = (Q, \Sigma, \delta, s, F)$$

が、語 w を受理する

$$\Updownarrow$$

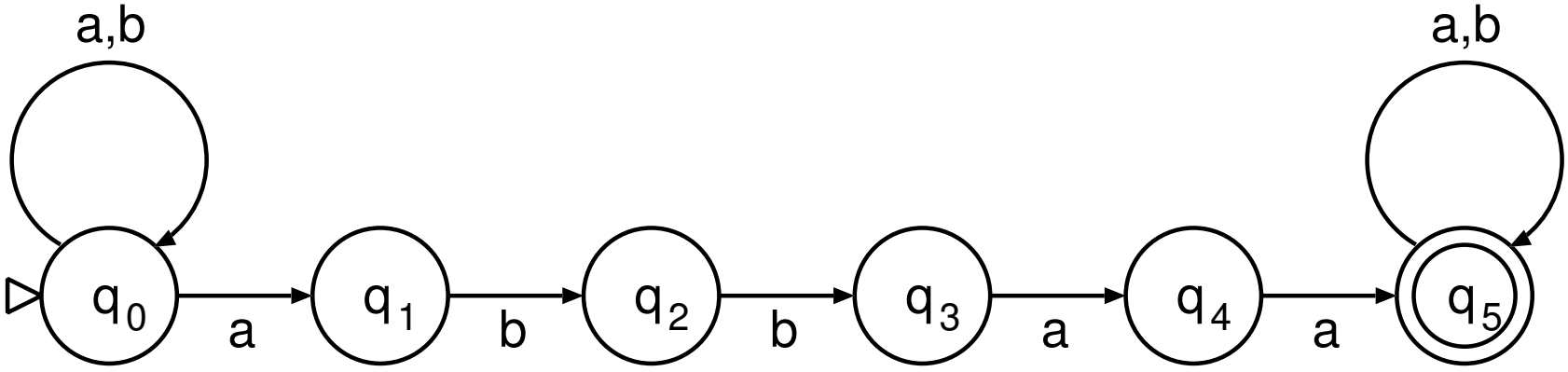
$\exists a_1, a_2, \cdots, a_n \in \Sigma \cup \{\varepsilon\} : w = a_1 a_2 \cdots a_n$
$\exists r_0, r_1, \ldots, r_n \in Q :$

- $r_0 = s$
- $r_i \in \delta(r_{i-1}, a_i)$ $(i = 1, \ldots, n)$
- $r_n \in F$

$L(M)$：M **が受理する語の全体**

$\cdots$ M **が認識する言語**

非決定性有限オートマトンの例
(状態遷移図による表示)



定理:

L **が或る (決定性) 有限オートマトンで認識される**

$\Updownarrow$

L **が或る非決定性有限オートマトンで認識される**

非決定性有限オートマトン M **に対し、** $L(M)$ **を認識する決定性有限オートマトン** M' **が構成できる**

定理:

L : **正規言語**

$\Updownarrow$

L **が或る有限オートマトンで認識される**

$\Updownarrow$

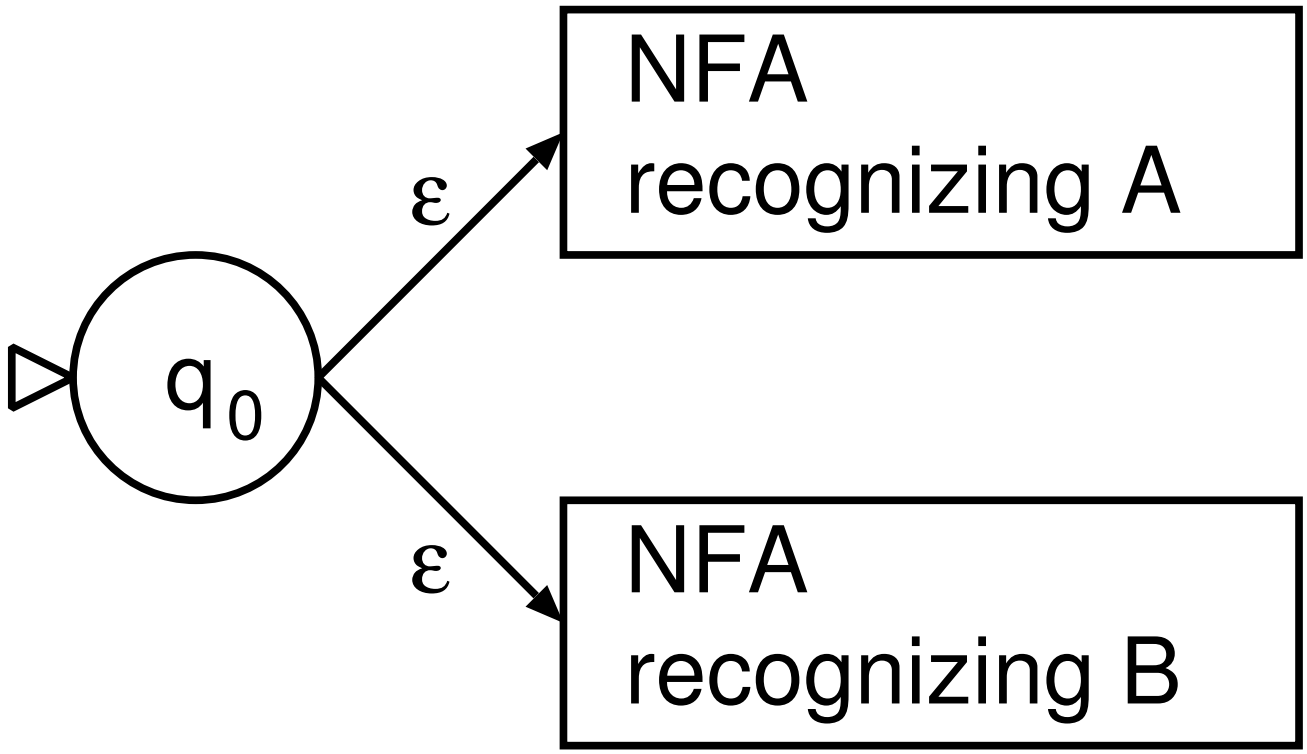
L **が或る非決定性有限オートマトンで認識される**

正規言語:

- $L(a) = \{a\}$
- $L(\varepsilon) = \{\varepsilon\}$
- $L(\emptyset) = \emptyset$
- $L(R \cup S) = L(R) \cup L(S)$
- $L(R \circ S) = L(R) \circ L(S)$
- $L(R^*) = L(R)^*$

言語 A, B を認識する
非決定性有限オートマトンから、
言語 $A \cup B, A \circ B, A^*$ を認識する
非決定性有限オートマトンが
構成できれば良い。(それは比較的容易)

$A \cup B$ を認識する
非決定性有限オートマトンの構成



非決定性有限オートマトンで認識できない言語が存在する!!

($\Longleftrightarrow$ 正規でない言語が存在する)

例: $A = \{a^n b^n \mid n \geq 0\}$

(a と b との個数が同じ)

証明は部屋割り論法

(の一種の pumping lemma)

による

より強力な計算モデルが必要

- **プッシュダウンオートマトン**
- **チューリングマシン**

例: “文法的に正しい” 数式とは どのようなものか?

- 単独の文字 (変数名) は式
- 式と式とを演算子で繋いだものは式
- 式を括弧で括ったものは式
- それだけ

$\longrightarrow$ これは式を作り出す規則とも考えられる

“文法的に正しい” 数式

初期記号 (開始変数) E から出発して、
次の規則のいづれかを
“非決定的に” 適用して得られるもの のみ

- $E \to A$
- $E \to EBE$
- $E \to (E)$
- $A \to$ 変数名のどれか
- $B \to$ 演算子のどれか
- 変数名・演算子・$(\ \cdot\ )$ は
それ以上書換えない (終端記号)

$\longrightarrow$ 生成規則 (書換規則)

生成規則を与えることでも
言語を定めることが出来る

$\longrightarrow$ **生成文法**

生成規則による “正しい” 語の生成

- **初期変数を書く**
- **今ある文字列中の或る変数を**
  生成規則のどれかで書換える
- **変数がなくなったら終わり**

例: $\{a^{2n}b^{2m+1}|n, m \geq 0\}$

(a が偶数個 (0 個も可) 続いた後に、b が奇数個続く)

正規表現で表すと、 $(aa)^*b(bb)^*$

- $S \to aaS$
- $S \to bB$
- $B \to bbB$
- $B \to \varepsilon$

まとめて次のようにも書く

- $S \to aaS|bB$
- $B \to bbB|\varepsilon$

実際の (自然言語を含めた) “文法” では、
或る特定の状況で現われた場合だけ
適用できる規則もあるだろう。

そのような生成規則は例えば次の形:

- $uAv \longrightarrow uwv$

$u, v \in \Sigma^*$ **: 文脈 (context)**

変数 A **が** uAv **の形で現われたら、**
語 $w \in \Sigma^*$ **で書換えることが出来る。**

生成文法の形式的定義

- V：有限集合 (**変数**の集合)
- Σ：有限集合 (**終端記号**の集合)
  ここに $V \cap \Sigma = \emptyset$
- R：有限集合 $\subset (V \cup \Sigma)^* \times (V \cup \Sigma)^*$
  (**規則**の集合)
- $S \in V$：**開始変数**

$(v, w) \in R$ が生成規則 $v \to w$ を表す。

文脈自由文法: 文脈が全て空列 ε

即ち、規則が全て $A \to w$ $(A \in V)$ の形

文脈自由文法の形式的定義

- V : 有限集合 (**変数**の集合)
- Σ : 有限集合 (**終端記号**の集合)
  ここに $V \cap \Sigma = \emptyset$
- R : 有限集合 $\subset V \times (V \cup \Sigma)^*$
  (**規則**の集合)
- $S \in V$: **開始変数**

$(A, w) \in R$ が生成規則 $A \to w$ を表す。

例: 言語 $A = \{a^n b^n | n \geq 0\}$ は
正規言語ではないが文脈自由言語である。

- $S \to aSb|\varepsilon$

従って、

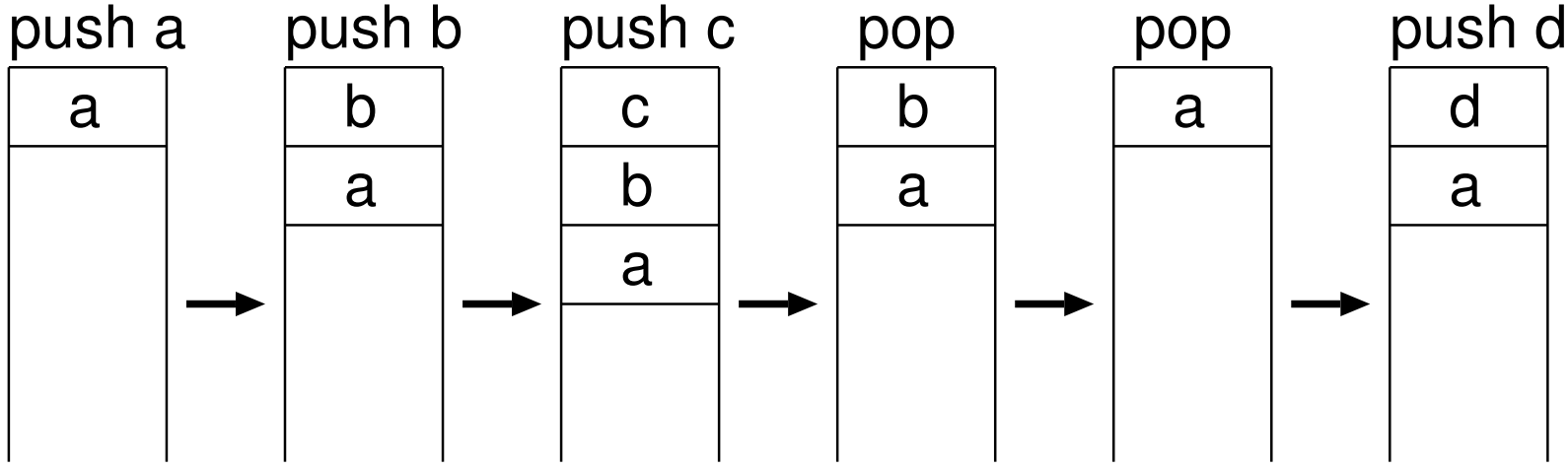
文脈自由言語は正規言語より真に広い!!

さて、正規言語を計算するモデルが
有限オートマトンであった。

文脈自由言語を計算するモデル
$\cdots$ プッシュダウンオートマトン

プッシュダウンオートマトン

(非決定性) 有限オートマトンに
プッシュダウンスタックを取り付けたもの



無限 (非有界) の情報を保持できるが、
読み書きは先頭だけ
· · · LIFO (Last In First Out)