計算理論　第8回
第5章：文脈自由文法と言語(1/2)

基礎工学部情報科学科
中川　博之

計算理論(後半)の進め方，概要

- 担当：中川 博之
  - TAは前半と同じ 吉田 征樹 君 (M1)
- テキスト，ミニレポートの形態も基本的に同じ
  - ミニレポートの回答期限：翌週月曜の23:59
- 主に扱う内容：文脈自由文法とその応用，
  その先に見えるもの
  - 最後の数回はテキストの範囲を越えます

講義の予定(中川担当分)

- 第8回　文脈自由文法と構文木
- 第9回　文脈自由文法の応用
- 第10回　プッシュダウンオートマトン
- 第11回　文脈自由言語の標準形
- 第12回　文脈自由言語の反復補題
- 第13回　文脈自由言語の閉包性と決定問題
- 第14回　文脈依存言語
- 第15回　チューリングマシンと決定可能性
- 第16回　期末試験(第8～15回講義分)

本日の概要

- 第5章：文脈自由文法と言語 (の前半)
  - テキスト：p.192～
  - 5.1 文脈自由文法
  - 5.2 構文木
- 重要概念
  - 文脈自由文法，文脈自由言語，導出，構文木

5.1 文脉自由文法

5.1.1 直観的な例: 回文

- 回文 (palindrome):
  - 前から読んでも後ろから読んでも同じ文字列
- 例
  - トマト
  - たけやぶやけた
  - 悪い鉄柵が腐っているわ
  - wasitacatisaw
- 【回文の定義】文字列 w が回文 $\Leftrightarrow w = w^R$
  - w^R : 文字列 w の前後を反転したもの

回文言語 L_{pal}

- ここではアルファベット{0, 1}に限定した回文を考える
- 回文言語 L_{pal}を集合として考えると
  - 属する文字列の例: 010, 0110, 101101, 1, 0, ε
  - 属さない文字列の例: 10, 110, 0101
- 形式的な定義
  - $L_{pal} = \{w \in \{0, 1\}^* \mid w = w^R\}$
  - L_{pal}は正則言語ではない

回文の再帰的な定義

- 基礎: ε, 0, 1は回文である
- 再帰:もしwが回文なら, 0w0と1w1も回文

回文はこれらの規則で構成できるものに限る

- この言語の定義は再帰的構造を有している

→文脈自由文法はこのような再帰的定義を形式的に記述する記法

文脈自由文法

- 再帰的定義を形式的に記述する記法のひとつ
  – いくつかの規則で構成

- 回文を定義する規則
  1. P → ε
  2. P → 0
  3. P → 1
  4. P → 0P0
  5. P → 1P1

・1～3が基礎，4, 5が再帰に該当
・Pは回文のクラスを表す変数
規則4のイメージ：
「Pが回文のクラスに属せば，0P0も回文クラスに属する」

5.1.2 文脈自由文法の定義

- 文脈自由文法 (Context-Free Grammar: CFG) Gは以下の4つ組で定義される

$$G = (V, T, P, S)$$

  - V: 変数 (variable) の集合
    - 非終端記号 (nonterminal symbol) とも呼ぶ
  - T: 終端記号 (terminal) の集合
  - P: 生成規則 (production rule)の集合
  - S: 出発記号 (start variable, start symbol)

回文文法 G_{pal}

- 回文文法G_{pal}を$G = (V, T, P, S)$の形式に当てはめると…
  - $V = \{P\}$
  - $T = \{0, 1\}$
  - $P = \{P \to \varepsilon, P \to 0, P \to 1, P \to 0P0, P \to 1P1\}$
  - $S = P$

よって, $G_{pal} = (\{P\}, \{0, 1\}, \{P \to \varepsilon, P \to 0, P \to 1, P \to 0P0, P \to 1P1\}, P)$

例5.3：式の文法G_{exp}

- （単純化した）式を表現するCFGを考える
  - 演算子と識別子により構成
- 演算子：加算と乗算に限る (+ と * のみ)
  - 括弧の使用も許す
- 識別子：構成要素はa, b, 0, 1に限る
  - （制約1）最初の文字はaまたはbに限定
  - （制約2）その後に{a,b,0,1}*の任意の列を追加
  - 正則表現で書くと　(a+b)(a+b+0+1)*
  - 識別子の例：a, b, a0, bab002
- 式の例： a, a1b+b0a, ba*(a0+b1), a+(a*b)

文脈自由文法で書くと...

- 式のクラスを変数E，識別子のクラスを変数Iで表現する

- $G_{exp} = (V, T, P, S)$
  - V = {E, I}
  - T = {a, b, 0, 1, +, *, (,)}
  - P (次スライド)
  - S = E

G_{exp}の生成規則集合P

1. E → I
2. E → E + E
3. E → E * E
4. E → (E)
5. I → a
6. I → b
7. I → Ia
8. I → Ib
9. I → I0
10. I → I1

規則1～4:式の構成法に関する規則
規則5～10:識別子の構成法に関する規則

生成規則の簡潔な表現法

- Pの要素をきちんと書くと10行になる
  - 簡潔に記述したい…

→頭部の変数が同一の生成規則を1つにまとめる
  - 本体を縦棒で区切って列挙
    - $E \rightarrow I \mid E+E \mid E*E \mid (E)$
    - $I \rightarrow Ia \mid Ib \mid I0 \mid I1 \mid a \mid b$

- $G_{exp} = (\{E, I\}, \{a, b, 0, 1, +, *, (,)\},$
  $\{E \rightarrow I \mid E+E \mid E*E \mid (E), I \rightarrow Ia \mid Ib \mid I0 \mid I1 \mid a \mid b\}, E)$

5.1.3 文法による導出

- 生成規則の適用目的
  - ある文字列が言語に属しているかを判定
  - 生成規則の適用法は2つある

- 方法1：再帰的推論（逆方向）
  - 本体から頭部へと生成規則を適用
- 方法2：導出（順方向）
  - 頭部から本体へと生成規則を適用

方法1:再帰的推論(逆方向)

- 本体から頭部へと生成規則を適用
  - [START] 終端記号だけの列
  - 本体と合致する文字列を頭部の変数に置き換えることを繰り返す
  - [GOAL] 出発記号 (1つの変数)

- 例:
  1. 00100 から 00P00 を推論
  2. 00P00 から 0P0 を推論
  3. 0P0 から P を推論

方法2: 導出（順方向）

- 頭部から本体へと生成規則を適用
  - [START] 出発記号 (1つの変数)
  - 記号列中の変数に生成規則を適用し置換
  - [GOAL] 終端記号だけの列

- 例: P ⇒ 0P0 ⇒ 00P00 ⇒ 00100

記法

- 本講義(およびテキスト)で用いる記法
  - 英小文字の最初の方 (a, b, ...): 終端記号
    - 数字や演算記号(+や括弧など)も終端記号
  - 英大文字の最初の方(A, B, ...) : 変数
  - 英小文字で最後の方 (w, zなど): 終端記号の列
  - 英大文字で最後の方 (X, Yなど): 終端記号または変数
  - ギリシャ小文字(α, βなど) : 終端記号と変数の一方または両方が含まれる列

導出を表す関係記号

- $\underset{G}{\Rightarrow}$: 生成規則を頭部から本体へと(1回)適用する過程を記述する記号

- $\alpha A \beta \underset{G}{\Rightarrow} \alpha\gamma\beta$
  - 文脈自由文法 G=(V, T, P, S)
  - α, β: (VUT)*中の列
  - A∈V　: 変数
  - (A→ γ) ∈ P : 生成規則

- 特にGが明らかなとき, $\underset{G}{\Rightarrow}$ を⇒ と記す

複数回の導出

- $\overset{*}{\underset{G}{\Rightarrow}}$: $\underset{G}{\Rightarrow}$を0回以上に拡張したもの

- 再帰的定義
  - 基礎: $\alpha \overset{*}{\underset{G}{\Rightarrow}} \alpha$
  - 再帰: $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$ かつ $\beta \underset{G}{\Rightarrow} \gamma$ ならば $\alpha \overset{*}{\underset{G}{\Rightarrow}} \gamma$

- 特にGが明らかなとき, $\overset{*}{\underset{G}{\Rightarrow}}$ を$\overset{*}{\Rightarrow}$ と記す

導出の例

$$E \underset{G_{exp}}{\Rightarrow} E * E$$
$$\underset{G_{exp}}{\Rightarrow} I * E$$
$$\underset{G_{exp}}{\Rightarrow} a * E$$
$$\underset{G_{exp}}{\Rightarrow} a * (E)$$
$$\underset{G_{exp}}{\Rightarrow} a * (E + E)$$
$$\underset{G_{exp}}{\Rightarrow} a * (I + E)$$
$$\underset{G_{exp}}{\Rightarrow} a * (a + E)$$
$$\underset{G_{exp}}{\Rightarrow} a * (a + I)$$
$$\underset{G_{exp}}{\Rightarrow} a * (a + I0)$$
$$\underset{G_{exp}}{\Rightarrow} a * (a + b0)$$

よって

$$E \overset{*}{\underset{G_{exp}}{\Rightarrow}} a * (a + b0)$$

5.1.4 最左導出と最右導出

- 最左導出：常に<u>最も左</u>の変数に生成規則を適用する導出方法

- 関係記号：$\underset{左}{\Rightarrow}$ あるいは $\underset{左}{\overset{*}{\Rightarrow}}$
- 特に文法を明示するとき：$\underset{左G}{\Rightarrow}$ あるいは $\underset{左G}{\overset{*}{\Rightarrow}}$

- 例：01<u>A</u>0B1C10 $\underset{左}{\Rightarrow}$ 01<u>2</u>0B1C10

最左導出の例（文法はG_{exp}だが省略）

$$E \underset{左}{\Rightarrow} E * E$$
$$\underset{左}{\Rightarrow} I * E$$
$$\underset{左}{\Rightarrow} a * E$$
$$\underset{左}{\Rightarrow} a * (E)$$
$$\underset{左}{\Rightarrow} a * (E + E)$$
$$\underset{左}{\Rightarrow} a * (I + E)$$
$$\underset{左}{\Rightarrow} a * (a + E)$$
$$\underset{左}{\Rightarrow} a * (a + I)$$
$$\underset{左}{\Rightarrow} a * (a + I0)$$
$$\underset{左}{\Rightarrow} a * (a + b0)$$

よって

$$E \overset{*}{\underset{左}{\Rightarrow}} a * (a + b0)$$

最右導出

- 最右導出 : 常に最も右の変数に生成規則を適用する導出方法

- 関係記号 : $\underset{右}{\Rightarrow}$ あるいは $\overset{*}{\underset{右}{\Rightarrow}}$
- 特に文法を明示するとき : $\underset{右G}{\Rightarrow}$ あるいは $\overset{*}{\underset{右G}{\Rightarrow}}$

- 例 : 01A0B1C10 $\underset{右}{\Rightarrow}$ 01A0B1210

5.1.5 ある文法の言語

- L(G)： 文法G=(V, T, P, S) が生成する言語
  - – Gの出発記号から導出できる終端記号列の集合

- 形式的定義： $L(G) = \{w \in T^* \mid S \overset{*}{\underset{G}{\Rightarrow}} w\}$

文脈自由言語

- 文法Gが文脈自由文法のとき，
  言語 L(G) は文脈自由言語

- 文脈自由言語（Context-Free Language: CFL）
  – 文脈自由文法Gにより定義される言語

L ⇔ L(G)であることの証明

- L : ある言語（文字列の集合）
- G : ある文法

が与えられたときに，L ⇔ L(G) を示す

- 証明方法 : L ⇒ L(G) と L ⇐ L(G)を示す
  - L⇒L(G) : 任意のw∈Lに対してw∈L(G)を示す（十分性）
  - L ⇐ L(G) : 任意のw∈L(G)に対してw∈Lを示す（必要性）

例：$L(G_{pal})$は回文の集合（定理5.7）

- 証明すべきこと:
  - 任意の$w \in \{0,1\}^*$に対して
    wは回文 $\Leftrightarrow$ $w \in L(G_{pal})$

- これを証明するためには
  - 十分性: wは回文 $\Rightarrow$ $w \in L(G_{pal})$
  - 必要性: wは回文 $\Leftarrow$ $w \in L(G_{pal})$

  を証明すればよい

$G_{pal} = (\{P\}, \{0, 1\},$
$\{P \to \varepsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1\}, P)$

十分性の証明 (1/2)

- 証明したいこと: wは回文 ⇒ $w \in L(G_{pal})$

→ 回文の長さ |w| に関する帰納法で証明

- 基礎: |w| = 0 または |w| = 1のとき
  - wはε, 0, 1のいずれか
  - いずれも生成規則P → ε, P → 0, P → 1 により生成可能. つまり$L(G_{pal})$ に属する

十分性の証明 (2/2)

- 帰納: $|w| \geq 2$のとき
  - wが回文であるためには，w=0x0 か w=1x1 の形でなければならない
  - また，xも回文でなければならない
  - このとき，$|x| = |w|-2$であり，帰納法の仮定より xはG_{pal}により生成可能（$P \stackrel{*}{\Rightarrow} x$）
  - $w = 0x0$ のとき， $P \Rightarrow 0P0 \stackrel{*}{\Rightarrow} 0x0$
  - $w = 1x1$ のとき， $P \Rightarrow 1P1 \stackrel{*}{\Rightarrow} 1x1$
  - つまり， $|w| \geq 2$のときもwは $L(G_{pal})$に属する
- よって，任意の回文wは G_{pal}で導出可能

必要性の証明 (1/2)

- 証明したいこと: $w \in L(G_{pal}) \Rightarrow w$は回文
- $P \overset{*}{\Rightarrow} w$ ならばwは回文であることを示せばよい
  → 生成規則の適用回数nに関する帰納法
- 基礎 : n=1のとき
  - $P \Rightarrow \varepsilon$, $P \Rightarrow 0$, $P \Rightarrow 1$ のいずれか
  - ε, 0, 1いずれも回文であるため成立

必要性の証明 (1/2)

- 帰納 : n+1のとき (n≥1)
  - 帰納法の仮定: 生成規則をn回以下適用してできた文字列xは回文である
  - 2回以上導出を適用するとき，導出は P⇒0P0または P⇒1P1のいずれかで始まる
  - P⇒0P0のとき， $P \Rightarrow 0P0 \overset{*}{\Rightarrow} 0x0$
  - P⇒1P1のとき， $P \Rightarrow 1P1 \overset{*}{\Rightarrow} 1x1$
  - 仮定より，n回適用して出来た文字列xは回文であり，そのとき，0x0と1x1はいずれも回文である
  - よってn+1のときに成立
- 従って， $w \in L(G_{pal}) \Rightarrow$ wは回文

5.1.6 文形式

- 文法G=(V, T, P, S)から得られる列α

$$S \overset{*}{\underset{G}{\Rightarrow}} \alpha$$

ただし $\alpha \in (V \cup T)^*$

を文形式と呼ぶ

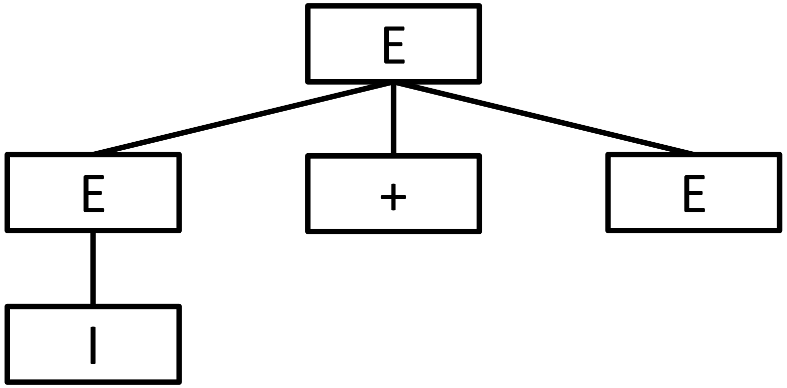
– 変数と終端記号のいずれを含んでもよい

– 最左導出で得られる文形式を 左文形式,
最右導出で得られる文形式を 右文形式 と呼ぶ

5.2 構文木

構文木 (parse tree)

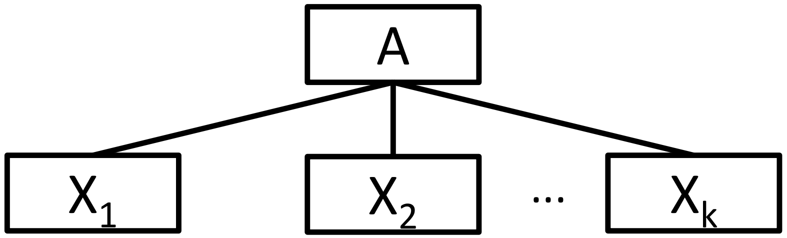
- 構文木(parse tree): 導出を表現する木構造



- コンパイラにおいては，ソースコードを表現するデータ構造として用いられる

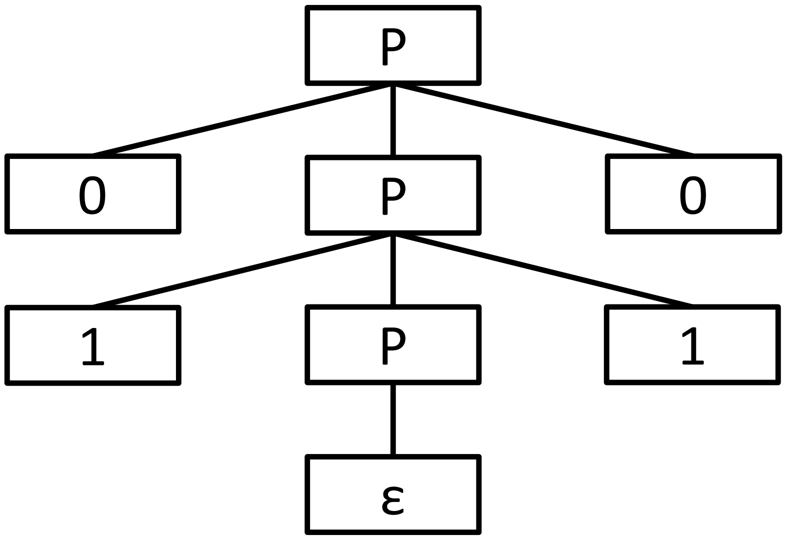
5.2.1 構文木の構成

- Given: G=(V, T, P, S)
- Gの構文木とは次の条件を満たす木
  1. 各内部節点のラベル：V中の変数
  2. 各葉のラベル：V中の変数，T中の終端記号，ε
     - εはその葉以外に兄弟節点がない場合のみ用いる
  3. 内部節点(親)のラベルがA，子節点のラベルが左から順に $X_1, X_2, \ldots, X_k$ ならば，$A \rightarrow X_1 X_2 \ldots X_k$ が生成規則集合Pに含まれる



例5.10：構文木の例

- 回文文法における1つの構文木
  - 導出 $P \overset{*}{\Rightarrow} 0110$ を示す構文木
  - $P \Rightarrow 0P0 \Rightarrow 01P10 \Rightarrow 01\varepsilon10$

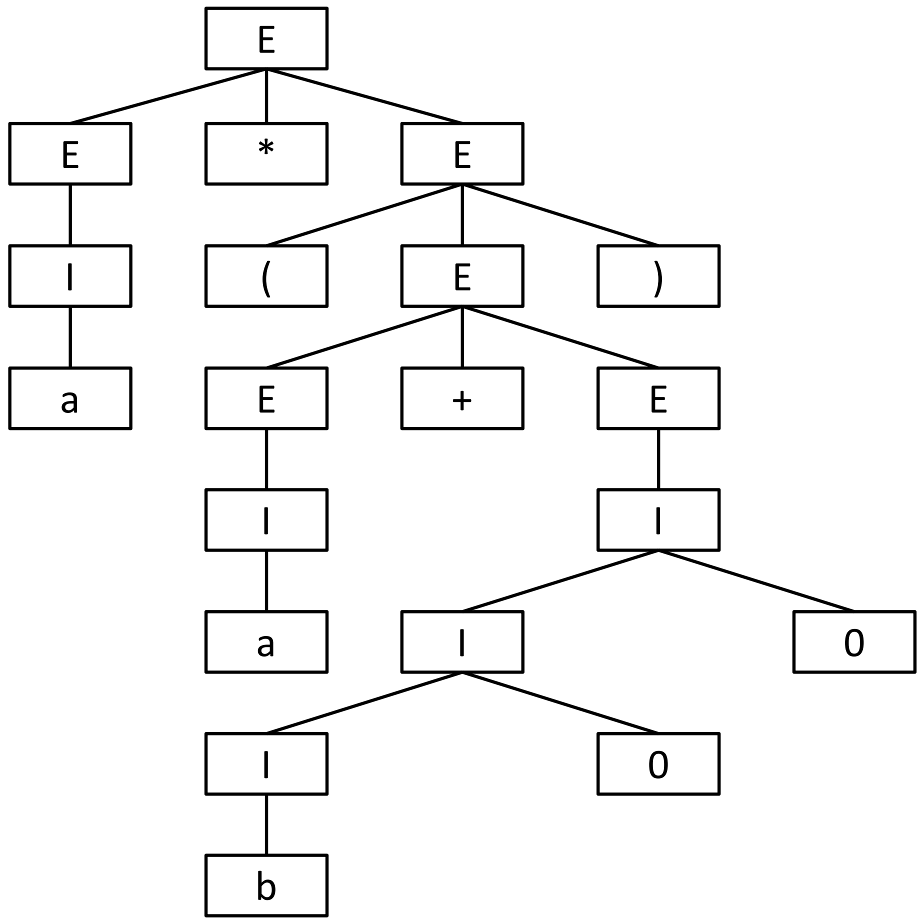


構文木の成果

- 構文木の成果：葉のラベルを左から右に並べて得られる文字列のこと
  - 根の変数から導かれる文形式の一つ
- 特に重要な構文木
  - (1) 成果が終端記号列であるもの
    - 葉のラベルがすべて終端記号かε
  - (2) 根のラベルが出発記号のもの
- (1),(2)とも満たす木の成果は，当該文法が生成する言語に含まれる列
  - 言語を，「出発記号を根とし，終端記号を成果とするような構文木の成果の集合」と定義することができる

例5.11

- $E \overset{*}{\Rightarrow} a*(a+b00)$
  - この構文木は，$a*(a+b00)$がG_{exp}の言語に属していることを示している

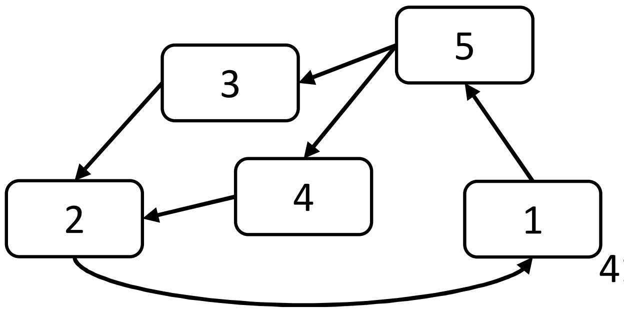


再帰的推論･導出と構文木

以下の5つは同値

1. 再帰的推論：本体から頭部への変換（推論）により，変数Aを出発記号とする終端記号列wが決定できる
2. 導出：$A \overset{*}{\Rightarrow} w$
3. 最左導出：$A \overset{*}{\underset{左}{\Rightarrow}} w$
4. 最右導出：$A \overset{*}{\underset{右}{\Rightarrow}} w$
5. 構文木：Aを根としwを成果とする構文木が存在

証明は省略（テキストp.209～）



ミニレポート

ミニレポート: 8-1

- テキスト p.204　問5.1.1(a)
- 次の言語に対する文脈自由文法を作れ

  $\{0^n 1^n \mid n \geq 1\}$

ミニレポート

ミニレポート: 8-2

- テキスト p.204　問5.1.2 a), b), c)
- 次の文法は正則表現0*1(0+1)*と同じ言語を生成する
  - S→A1B
  - A→0A|ε
  - B→0B|1B|ε

次の列の再左導出と最右導出を示せ.

- a) 00101
- b) 1001
- c) 00011