形式言語理論

Formal Language Theory

12.　文脈自由言語の性質

九州大学
KYUSHU UNIVERSITY

本日の内容

- 文脈自由言語に対する反復補題
- 文脈自由言語に対する反復補題を使う
- 文脈自由言語の閉性
- 文脈自由言語に関する決定手続き
- 文脈自由文法の曖昧性
- 文脈自由言語に関する決定不能な問題

文脈自由言語に対する反復補題

文脈自由言語に対する反復補題

【補題6.1】 言語 $L \subseteq \Sigma^*$ が CFL ならば，以下を満たす正整数 n が存在する:
$|w| \geq n$ となる任意の文字列 $w \in L$ に対して，以下を満たす文字列 $u, v, x, y, z \in \Sigma^*$ が存在する.

① $w = uvxyz$.

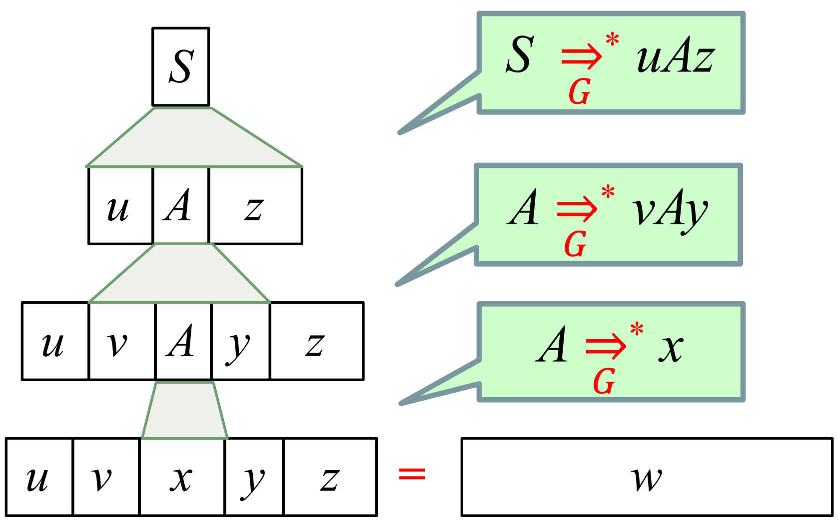
② $|vxy| \leq n$ かつ $vy \neq \varepsilon$.

③ すべての k=0,1,2,… に対して $uv^kxy^kz \in L$.

補題6.1の証明：概略

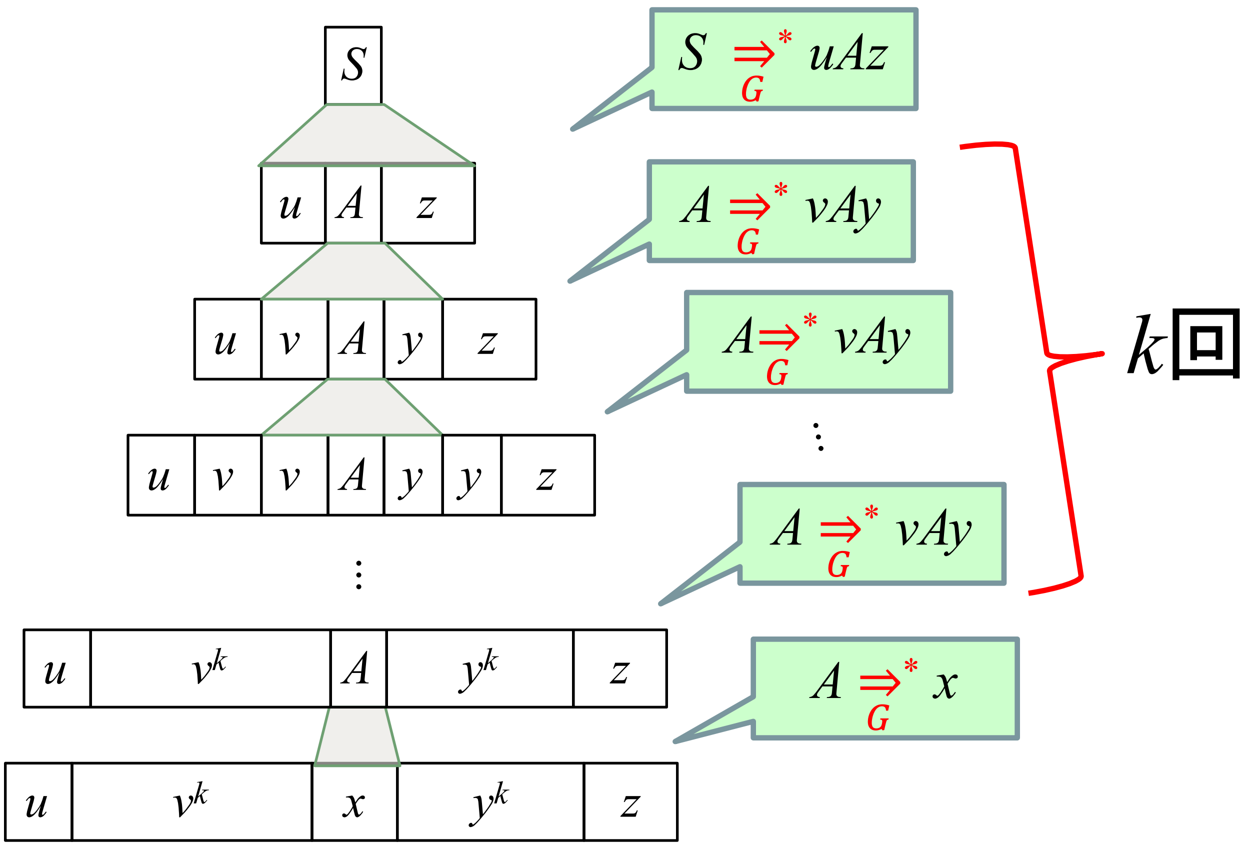
$|w| \geq n$

- S から導出される文字列 w が十分長いならば非終端記号 A と文字列 u, v, x, y, z が存在して以下のようになる.



補題6.1の証明：概略

- 導出 $A \underset{G}{\Rightarrow}^{*} vAy$ を k 回繰り返せば S から $u\,v^k\,x\,y^k\,z$ が導出される.



「w が十分長いならば．．．になる」の証明

- w の導出木の高さを h とする．生成規則の右辺の長さの最大値を r とすると，葉の個数は高々 r^h．
- w の長さは葉の個数を超えないため，$|w| \le r^h$．
- ここで，$|w|>r^{|N|}$ とすると，$h>|N|$でなければならない．したがって，根から葉へ向かう経路のうちで長さ $|N|+1$ 以上のものが存在する．
- 鳩小屋原理より，この経路上に2回以上現れる非終端記号がある．
- すなわち，$r^{|N|}$ より大きな n をとれば補題は成立する．
  - 厳密には$|vxy|\le n$, $|vy|>0$を示す必要あり．

文脈自由言語に対する反復補題を使う

例題

【例題6.1】言語$L_1 = \{a^n b^n c^n \mid n > 0\}$は文脈自由言語でないことを証明せよ.

$L_1=\{a^n b^n c^n \mid n>0\}$がCFLでないことの証明

- L_1 が文脈自由言語であると仮定する．反復補題の n に対して $w=a^n b^n c^n$ とおく．
- 分割 $w = uvxyz$ が存在して反復補題の②③を満たす．
- ③より$w' = uxz \in L_1$ である． $m=|vy|>0$とおく．
- $|vxy| \leq n$ であるから次の2つの場合を考えればよい．
  - vxyが$a^n b^n$に含まれる場合．
    $w'=a^{n-i}b^{n-j}c^n$ $(i+j=m>0, i \geq 0, j \geq 0)$となり， $w' \notin L_1$.
  - vxyが$b^n c^n$に含まれる場合．
    $w'= a^n b^{n-i} c^{n-j}$ $(i+j=m>0, i \geq 0, j \geq 0)$となり， $w' \notin L_1$.
- よって背理法により， L_1 は文脈自由言語ではない．

QED

例題

【例題6.2】言語$L_2 = \{uu \mid u \in \{a, b\}^*\}$は文脈自由言語でないことを証明せよ.

$L_2=\{uu \mid u \in \{a, b\}^*\}$がCFLでないことの証明

- L_2 が文脈自由であると仮定する．反復補題の n に対して $w=a^nb^na^nb^n$ とおく．分割 $w = uvxyz$ が存在して反復補題の②③を満たす．
- ③より $w' = uxz \in L_2$である． $m=|vy|>0$とおく．
- $|vxy| \leq n$ であるから次の3つの場合を考えればよい．
  - vxyがwの前半のa^nb^nに含まれる場合．
    $w'=a^{n-i}b^{n-j}a^nb^n$ $(i+j=m>0, i \geq 0, j \geq 0)$となり， $w' \notin L_2$.
  - vxyがwの後半のa^nb^nに含まれる場合．
    $w'=a^nb^na^{n-i}b^{n-j}$ $(i+j=m>0, i \geq 0, j \geq 0)$となり， $w' \notin L_2$.
  - vxyがwのb^na^nに含まれる場合．
    $w'=a^nb^{n-i}a^{n-j}b^n$ $(i+j=m>0, i \geq 0, j \geq 0)$となり， $w' \notin L_2$.
- よって背理法により， L_2 は文脈自由言語ではない．

QED

文脈自由言語の閉性

和集合・連接・Kleene閉包に関する閉性

【定理6.1】文脈自由言語の族は和集合，連接，Kleene閉包に関して閉じている．

定理6.1の証明

- 言語L_1, L_2が文脈自由ならば言語 $L_1 \cup L_2$, $L_1 L_2$, L_1^* は文脈自由言語であることを示す.
  - $G_1 = (N_1, \Sigma, P_1, S_1)$, $G_2 = (N_2, \Sigma, P_2, S_2)$ を，それぞれ，言語 L_1, L_2 を生成するCFGとする．一般性を失わずに，$N_1 \cap N_2 = \varnothing$ としてよい.
  - CFG $G = (N, \Sigma, P, S)$ $(S \notin N_1 \cup N_2)$について次が成り立つ.
    - $N = N_1 \cup N_2 \cup \{S\}, P = P_1 \cup P_2 \cup \{S \to S_1 | S_2\}$ならば $L(G) = L_1 \cup L_2$.
    - $N = N_1 \cup N_2 \cup \{S\}, P = P_1 \cup P_2 \cup \{S \to S_1 S_2\}$ ならば $L(G) = L_1 L_2$.
    - $N = N_1 \cup \{S\}, P = P_1 \cup \{S \to SS_1 | \varepsilon\}$ ならば $L(G) = L_1^*$.

QED

積集合に関する閉性

【定理6.2】文脈自由言語の族は積集合に関して閉じていない.

定理6.2の証明

- $\Sigma = \{a, b, c\}$とする．Σ 上の言語
$L_1=\{a^n b^n c^m \mid n>0, m>0\}$, $L_2=\{a^n b^m c^m \mid n>0, m>0\}$
について，$L_3 = L_1 \cap L_2 = \{a^n b^n c^n \mid n>0\}$ とおく．
- L_1, L_2 は CFL であるが，L_3 は CFL ではない．
- よって，文脈自由言語の族は積集合に関して閉じていない．

QED

補集合に関する閉性

【定理6.3】文脈自由言語の族は補集合に関して閉じていない.

定理6.3の証明

- 任意の $X \subseteq \Sigma^*$ に対して X の補集合$\Sigma^* - X$を X^c と書くことにする.
- de Morganの法則により，任意のCFL L_1, L_2 に対して $L_1 \cap L_2 = (L_1{}^c \cup L_2{}^c)^c$ である.
- 定理6.1より，文脈自由言語の族は和集合演算について閉じている.
- 文脈自由言語の族が補集合演算について閉じていると仮定すると，積集合についても閉じていることになり定理6.2に矛盾.
- よって背理法により，文脈自由言語の族は補集合に関して閉じていない.

復習：準同型写像

【定義】Σ, Δ をアルファベットとする．写像 $h: \Sigma^* \to \Delta^*$ が準同型写像であるとは，次を満たすときをいう．

すべての $x, y \in \Sigma^*$ に対して $h(xy) = h(x)h(y)$

- 上式で $x{=}y{=}\varepsilon$ とおくことにより，$h(\varepsilon) = \varepsilon$ を得る．
- 準同型写像 hは，各 $a{\in}\Sigma$ に対する値 $h(a){\in}\Delta^*$ が定まれば一意に定まる．

- 以下のように定める．
  - $h(L) = \cup_{x\in L} h(x) \quad (\forall L \subseteq \Sigma^*)$
  - $h^{-1}(L) = \{x \in \Sigma^* \mid h(x) \in L\} \ \ (\forall L{\subseteq}\Delta^*)$.

準同型写像に関する閉性

【定理6.4】文脈自由言語の族は準同型写像に関して閉じている.

すなわち，任意の準同型写像 $h : \Sigma^* \to \Delta^*$ に対し，言語 $L \subseteq \Sigma^*$ が文脈自由ならば言語 $h(L) \subseteq \Delta^*$ は文脈自由である.

定理6.4の証明

- $L \subseteq \Sigma^*$を任意の CFL とし, $G = (N, \Sigma, P, S)$ を $L=L(G)$となる CFG とする.
- 準同型写像 $h: \Sigma^* \to \Delta^*$ を $(N \cup \Sigma)^*$から $(N \cup \Delta)^*$への準同型写像に次で拡張する.
  - $h(A)=A \ \ (\forall A \in N)$
- CFG $G' = (N, \Delta, P', S)$ を次のように定める.
  - $P' = \{ A \to h(\gamma) \mid A \to \gamma \in P \}$
- $h(L) = L(G')$が成立する.

QED

逆準同型写像に関する閉性

【定理6.5】 文脈自由言語の族は準同型写像の逆写像に関して閉じている.

すなわち，任意の準同型写像 $h : \Sigma^* \to \Delta^*$ に対して，言語 $L \subseteq \Delta^*$ が文脈自由ならば言語 $h^{-1}(L) \subseteq \Sigma^*$ は文脈自由である.

証明は省略する.

正規言語との積集合

- 定理6.2に示したとおり，文脈自由言語の族は積集合に関しては閉じていない.
- しかし，正規言語との積集合に関しては閉じている.

【定理6.6】任意の文脈自由言語 L と任意の正規言語 R に対して，$L \cap R$ は文脈自由言語である.

定理6.6の証明：概略

- $M = (Q_M, \Sigma, \Gamma, \delta_M, q_0, Z_0, F_M)$ を $L=L(M)$ となる PDA, $A = (Q_A, \Sigma, \delta_A, p_0, F_A)$ を $R=L(A)$ となるDFA とする.
- M と A の動作を同時に模倣する PDA $M' = (Q_A \times Q_M, \Sigma, \Gamma, \delta, (p_0, q_0), Z_0, F_A \times F_M)$ を次で定義する.
  - $(p, q) \in Q_A \times Q_M$, $Z \in \Gamma$ に対し：
    - $a \in \Sigma$ ならば
      $\delta((p, q), a, Z) = \{ ((p', q'), \gamma) \mid p' = \delta_A(p, a) \wedge (q', \gamma) \in \delta(q, a, Z) \}$
    - $a=\varepsilon$ ならば
      $\delta((p, q), a, Z) = \{ ((p, q'), \gamma) \mid (q', \gamma) \in \delta(q, a, Z) \}$
- $L(M') = L(M) \cap L(A)$ が成り立つ.

QED

文脈自由言語に関する決定手続き

空性判定問題

【定義】(空性判定問題).
入力: CFG $G = (N, \Sigma, P, S)$.
出力: $L(G)=\varnothing$ ならYes，そうでないならNo.

【定理6.7】 CFLに対する空性判定問題を解くアルゴリズムが存在する.

定理6.7の証明

- 次のアルゴリズムによって空性判定問題を解くことができる.
  1. 定理4.3の証明に示した手法により，有用な記号の集合を求める.
  2. 開始記号 S が有用ならば No，そうでなければ Yes を出力する.

QED

無限性判定問題

【定義】(無限性判定問題).
入力: CFG $G = (N, \Sigma, P, S)$.
出力: $L(G)$ が無限集合 なら Yes，そうでないなら No.

【定理6.8】 CFL に対する無限性判定問題を解くアルゴリズムが存在する.

定理6.8の証明

- 次のアルゴリズムによって無限性判定問題を解くことができる.
  1. 定理4.3の証明に示した手法により，G と等価で既約な CFG を求め，改めてこれを $G=(N, \Sigma, P, S)$ とする.
  2. すべての$A\in N$に対し
     $Reach(A)=\{B\in N \mid \exists\alpha, \exists\beta\in(N\cup\Sigma)^*, A\underset{G}{\Rightarrow}^*\alpha B\beta\}$
     を求める.
  3. $A\in Reach(A)$ となる $A\in N$ が存在するならば Yes,
     そうでなければ No を出力する.

QED

所属性判定問題

【定義】(所属性判定問題).

入力: CFG $G = (N, \Sigma, P, S)$, 文字列 $w \in \Sigma^*$.

出力: $w \in L(G)$ なら Yes, そうでないならNo.

【定理6.9】 CFLに関する所属性判定問題を解くアルゴリズムが存在する.

定理6.9の証明

【補題6.2】 既約で広義の Greibach 標準形の CFG $G = (N, \Sigma, P, S)$ と文字列 $w \in \Sigma^*$ に対し，次が成り立つ．

$$w \in L(G) \Leftrightarrow S \underset{G}{\Rightarrow}^{|w|} w .$$

- 補題6.2に基づき，次のアルゴリズムによって所属性判定問題を解くことができる．
  1. G を既約な CFG に変換し，さらに広義の Greibach 標準形に変換する．
  2. ちょうど $|w|$ ステップの最左導出をすべて枚挙し，w が導出されるならば Yes，そうでなければ No を出力する．

QED

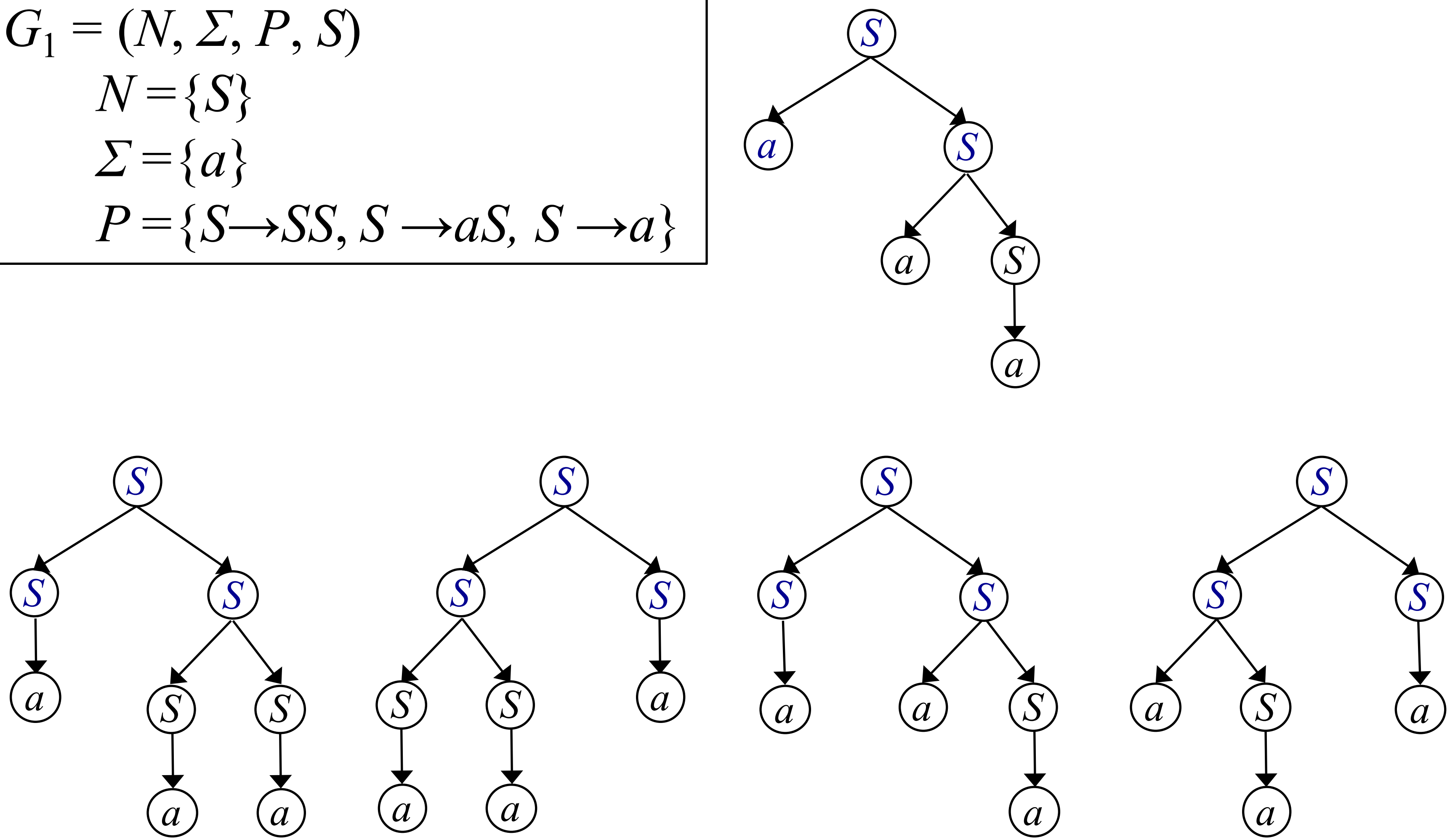
文脈自由文法の曖昧性

CFGの曖昧性

【定義】 CFG G が曖昧であるとは，文字列 $w \in L(G)$ が存在して w に対する導出木が2つ以上存在するときをいう.

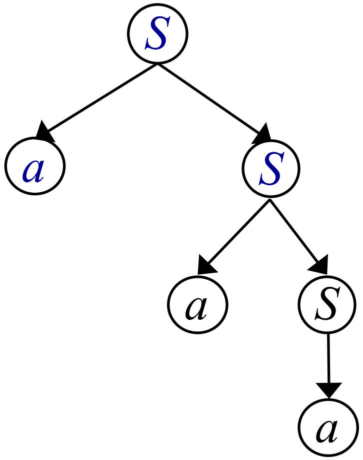
例　曖昧なCFG

$$G_1 = (N, \Sigma, P, S)$$
$$N = \{S\}$$
$$\Sigma = \{a\}$$
$$P = \{S \to SS,\ S \to aS,\ S \to a\}$$



例 同じ言語を生成する曖昧でないCFG

$$G_2 = (N, \Sigma, P, S)$$
$$N = \{S\}$$
$$\Sigma = \{a\}$$
$$P = \{S \to aS,\ S \to a\}$$



本質的に曖昧なCFL

【定理6.10】 $L = L(G)$ を満たすどの CFG G も曖昧であるような CFL L が存在する.

- このような CFL を本質的に曖昧であるという.
- 本質的に曖昧なCFLの例として，次が知られている.（証明は省く．）

$$L=\{a^n b^n c^m d^m \mid n>0, m>0\} \cup \{a^n b^m c^m d^n \mid n>0, m>0\}$$

文脈自由言語に関する決定不能な問題

全文字列生成判定問題

【定義】(全文字列生成判定問題).
入力: CFG $G = (N, \Sigma, P, S)$.
出力: $L(G) = \Sigma^*$ なら Yes, そうでないなら No.

【定理6.11】 CFGに関する全文字列生成判定問題を解くアルゴリズムは存在しない.

- 証明は「計算可能性理論」の内容となるためここでは省略する.
  - 決定不能である「Turing機械の受理判定問題」からの帰着を用いて証明できる.

等価性判定問題

【定義】(等価性判定問題).
入力: CFG G_1, G_2.
出力: $L(G_1)=L(G_2)$ なら Yes, そうでないなら No.

【定理6.12】 CFGに関する等価性判定問題を解くアルゴリズムは存在しない.

定理6.12の証明

- 等価性判定問題を解くアルゴリズムが存在すると仮定する.
- G_{ALL}を $L(G_{\mathrm{ALL}})=\Sigma^*$ となる自明な CFG とする.
- 任意の CFG G に対し G と G_{ALL}の等価性判定を行うことで, $L(G)=\Sigma^*$ であるか否かが判定できる.
- これは, 定理6.11に矛盾.
- よって背理法により, 等価性判定問題を解くアルゴリズムは存在しない.

QED

曖昧性判定問題

【定義】(曖昧性判定問題).
入力: CFG G.
出力: Gが曖昧なら Yes，そうでないなら No.

【定理6.13】 CFGに関する曖昧性判定問題を解くアルゴリズムは存在しない.

- 証明は「計算可能性理論」の内容となるため ここでは省略する.
  - 決定不能である「POSTの対応問題」からの帰着を用いて証明できる.