修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院　　情報理工学研究科　情報・ネットワーク工学専攻　博士前期課程		
氏　　名	西山　舜	学籍番号	1831117
論 文 題 目	スタック機構を持つ木変換器の表現力		

要　　旨

関数プログラミングや文書処理などの構造化されたデータを扱うプログラムに対して，その解析や最適化のために様々な計算モデルが考案されてきた．木変換器は，そのような計算モデルのひとつであり，その理論は非常に有用である．しかし，既知の木変換器理論は，構文解析器をはじめ扱えないプログラムも多い．このため，より広い範囲のプログラムを扱える計算モデル，スタック木変換器が中野により提案された．しかし，スタック木変換器の性質はまだ分かっていない部分が多く，応用を考える上で不十分な点が多い．

本研究では，スタック木変換器に関する性質と表現力の調査を行い，既知の木変換器との違いを明らかにする．また，応用を考える上で重要である，スタック属性付き木変換器とスタックマクロ木変換器の表現力の違いを明らかにする．

2019 年度修士論文

スタック機構を持つ木変換器の表現力

電気通信大学
大学院情報理工学研究科
情報・ネットワーク工学専攻
コンピュータサイエンスプログラム

学籍番号　　: 1831117
氏名　　: 西山　舜
主任指導教員 : 岩崎 英哉 教授
指導教員　　: 小林 聡 教授
提出日　　: 2020 年 1 月 27 日

概要

関数プログラミングや文書処理などの構造化されたデータを扱うプログラムに対して，その解析や最適化のために様々な計算モデルが考案されてきた．木変換器は，そのような計算モデルのひとつであり，その理論は非常に有用である．しかし，既知の木変換器理論は，構文解析器をはじめ扱えないプログラムも多い．このため，より広い範囲のプログラムを扱える計算モデル，スタック木変換器が中野により提案された．しかし，スタック木変換器の性質はまだ分かっていない部分が多く，応用を考える上で不十分な点が多い．

本研究では，スタック木変換器に関する性質と表現力の調査を行い，既知の木変換器との違いを明らかにする．また，応用を考える上で重要である，スタック属性付き木変換器とスタックマクロ木変換器の表現力の違いを明らかにする．

目次

第 1 章　はじめに

構造化されたデータに対する変換は，関数プログラミングや文書処理において重要な役割を果たしている．そのため，そのような変換に対する形式的な**計算モデル** (*model of computation*) を考え，その計算モデルに対する理論を構築することは，関数プログラミング言語における最適化 [Küh98][VK04] や XML 文書の妥当性検証 [MSV03][FH07] などの手法の確立に大きな役割を果たす．そのような計算モデルによる理論のひとつに，**木変換器理論** (*the theory of tree transducers*) [FV98][Eng15] がある．この理論では，構造化されたデータを木構造とみなし，その間の変換を表す構文的な構造，**木変換器** (*tree transducer*) を計算モデルとして扱う．そして，ある変換のモデルである木変換器に対して，構文的な側面からの解析を行うことで，その変換の表現力や性質を解き明かす．

木変換器理論では，原始的な再帰しか行えない**トップダウン木変換器** (*top-down tree transducer, TDTT*)，木に属性を付与しその間の依存関係を辿ることで木の走査を行う**属性付き木変換器** (*attributed tree transducer, ATT*)，コンテキストを保持しながら再帰を行う**マクロ木変換器** (*macro tree transducer, MTT*) の，代表的な 3 つの木変換器によって特徴付けされた変換を主に扱っている [FV98]．これらの木変換器による特徴付けは非常に強力で，多くの変換を形式化できる反面，入力データの構造が明確であることを前提とする計算モデルであり，例えば構文解析前の入力データを扱う変換を表すことはできない．具体的な例として，以下の変換を表すことができない．

$$
\begin{aligned}
\mathrm{parse}(Ax,(y_1,(y_2,y))) &= \mathrm{parse}\left(x,\left(\begin{array}{ccc} & A & \\ y_1 & & y_2 \end{array},y\right)\right)\\
\mathrm{parse}(Bx,y) &= \mathrm{parse}(x,(B,y))\\
\mathrm{parse}(\epsilon,(y,())) &= y
\end{aligned}
$$

この変換は，次のように A, B による文字列を二分木表現として構文解析し，結果を返す．

$$
\mathrm{parse}(BBABA,()) = \begin{array}{ccccc} & & A & & \\ & \swarrow & & \searrow & \\ B & & & & A \\ & \swarrow & & \searrow & \\ & B & & & B \end{array}
$$

上記のような変換に対しては，木変換器理論の適用を行うことはできない．この問題を解決すべく，中野 [Nak09] は木変換器を拡張し，上記の例を表すことを可能にする**スタック木変換器**を提案した．

木変換器が木から木への変換に対する計算モデルであるのに対し，スタック木変換器は木から**スタック構造**に対する計算モデルになっている．スタック構造とは，単連結リストに特殊な分解操作を許容したデータ構造で，木を複数個保持したり，先頭の有限個分の木から新たに作成した

木を先頭に加える操作が可能である．最終的にこのスタック構造から先頭のひとつの木を取り出し，それを出力することで，木からスタック構造への変換を木から木への変換とみなすことができる．中野はさらに，木変換器理論で知られている結果が，部分的にスタック木変換器においても成り立つことを示した [Nak09]．ここから，木変換器理論を利用した応用技術を，スタック木変換器に対しても適用できるようにすることで，既存の技術を拡張しより広い範囲の変換を扱えるようになることが期待されている．さらに，中野 [Nak09] は，スタック木変換器が XML 文書のストリーム処理 [NN01] に対する形式的なモデルとしても使用できることを示唆している．これは，既存の木変換器理論による応用技術を拡張できる他に，スタック木変換器についても有用な応用技術が存在することを示している．

しかし，スタック木変換器はまだ性質が明らかになっていない部分が多く，特に応用を考える上で重要な

- 逆像の木言語としての複雑性
- 表現力の階層

がまだ十分には研究されていない．本研究は，これら 2 つの性質を明らかにするため，スタック木変換器と木変換器の関係性について調査する．そして，スタック木変換器に対し，正規木言語の逆像が，木変換器と異なり正規木言語に収まらず，さらに文脈自由木言語にさえ収まらないことを示す．またスタック木変換器に対して，表現力の階層を明らかにするうえで重要な，樹高性と呼ばれる，変換に対して出力木の高さが入力木のどのような特徴量に依存するかを明らかにする．そして，樹高性により特徴づけられるスタック木変換器の階層が，木変換器での階層に対応することを示す．さらに，属性付き木変換器及び滞在と呼ばれる拡張を施したマクロ木変換器，**滞在付きマクロ木変換器** (*stay macro tree transducer, stayMTT*) のスタック木変換器への拡張に対しては，木変換器理論の一部の結果をそのまま適用できないことを示す．

トップダウン木変換器，属性付き木変換器，マクロ木変換器，滞在付きマクロ木変換器をそれぞれスタック木変換器に拡張したものを，**スタックトップダウン木変換器** (*StkTT*)，**スタック属性付き木変換器** (*StkATT*)，**スタックマクロ木変換器** (*StkMTT*)，**滞在付きスタックマクロ木変換器** (*stayStkMTT*) と呼ぶ．また，これらが表現できる変換のクラスを，TDTT，ATT，MTT，stayMTT，StkTT，StkATT，StkMTT，stayStkMTT と表記する．また，スタック属性付き木変換器に対して，**非巡回制約** (*non-circular*) と呼ばれる制約を付けた時の変換のクラスを $\mathrm{StkATT}_{\mathrm{ncr}}$ と表記する．この時，これらのクラスの階層は図 1.1 のようになる．

- (1)，(2)，(3) はそれぞれ，[Fül81]，[Eng80]，[EM03a] で既に示された，木変換理論における既知の事実である．これらの事実については，第 3 章でまとめた．
- (4)，(5)，(6) は一部中野が示しており [Nak09]，本研究では 5.2 の定理 326 で完全な証明を与えた．
- (7)，(8)，(11) は，本研究で示した関係になる．それぞれ 5.2 で完全な証明を与えており，

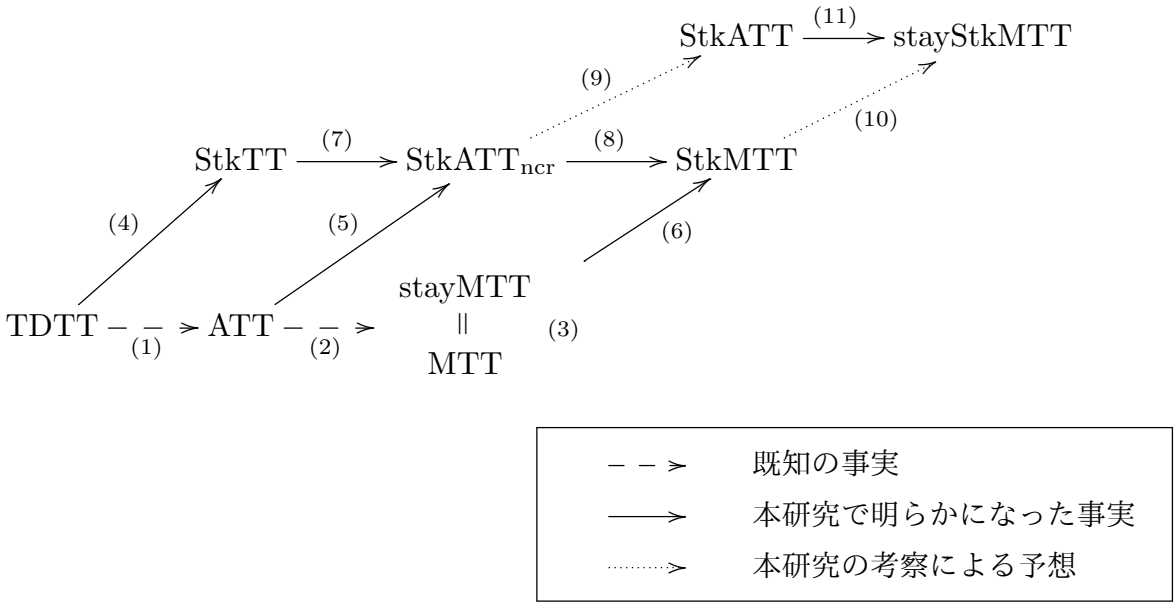


図 1.1　変換クラスの階層 (矢印は $\subsetneq$ を示す)

(7)，(8) は定理 323 で，(11) は定理 340 で示した．

- (9)，(10) は $\subsetneq$ が成り立つという予想であり，本研究では示すに至らなかった．この考察は，5.3 で述べている．

本論文の構成は，次のとおりである．2 章では，基本的な用語の定義及び定理について述べる．3 章では，3 つの基本的な木変換器の定義及び，木変換器理論において知られている定理について述べる．4 章では，3 章で述べた木変換器の定義及び [Nak09] を元に，スタック木変換器の定義について述べる．5 章では，スタック木変換器と木変換器の関係性及び，スタック木変換器における逆像の木言語としての複雑性，表現力の階層について述べる．6 章では，木変換器理論を拡張する関連研究の紹介と本研究との関係性について述べる．

第 2 章　準備

基本的な表記，定義を示す．基本的な原理は [FV98] に従う．

2.1　基本事項

量化子 (*quantifier*) の束縛をコンマ (,) で続けて書く．束縛の終わりをピリオド (.) で示す．例えば，

$$\forall x_1 \in X_1, x_2 \in X_2 \,.\, \exists y_1 \in Y_1, y_2 \in Y_2 \,.\, x_1 = y_1 \wedge x_2 = y_2$$

は，

$$\forall x_1 \in X_1 \,.\, \forall x_2 \in X_2 \,.\, \exists y_1 \in Y_1 \,.\, \exists y_2 \in Y_2 \,.\, x_1 = y_1 \wedge x_2 = y_2$$

と等しい．また，量化子の束縛において，*s.t.* (*such that*) を省略し，コンマ (,) で繋げて書く．例えば，

$$\forall x \in \{0,1\}, x \neq 0 \,.\, x = 1$$

は，

$$\forall x \in \{0,1\} \,.\, x \neq 0 \implies x = 1$$

と等しい．また，$\implies$，$\iff$ が他の記号と混同する場合，それぞれ implies，iff を使用する．

集合 (*set*)，**クラス** (*class*)，**文字列** (*string*)，**述語** (*predicate*)，**関係** (*relation*)，**族** (*indexed family*) に関して，本論文で使用する表記を示す．

定義 1 (集合に対する表記). 集合に関して，以下の表記を用いる．

- 集合 A について，その**濃度** (*cardinality*) を $|A|$ と表記する．なお，A が**有限集合** (*finite set*) の時，濃度とは要素の個数のことである．
- 集合 A について，$a \in A$ を $a : A$ と表記する．
- **自然数**の集合 $\{0, 1, ...\}$ を $\mathbb{N}$ と表記する．
- 集合 A の**冪集合** $\{X \mid X \subseteq A\}$ を $\mathcal{P}(A)$，有限冪集合 $\{X \in \mathcal{P}(X) \mid X \text{ は有限集合 }\}$ を $\mathcal{P}_{fin}(A)$ と表記する．
- 自然数 $n \in \mathbb{N}$ について，集合 $\{1, ..., n\}$ を $[n]$ と表記する．
- 集合 $A_1, ..., A_n$ の**直積** (*cartesian product*) $\{(a_1, ..., a_n) \mid a_1 \in A_1, ..., a_n \in A_n\}$ を $A_1 \times \cdots \times A_n$ と表記する．集合 A の n 直積 $\underbrace{A \times \cdots \times A}_{n \text{ 項}}$ を A^n と表記する．特に，$A^0 = \{\epsilon\}$ である．

- 集合 $A_1, \ldots, A_n$ の**直和** (*disjoin union*) $(A_1 \times \{1\}) \cup \cdots \cup (A_n \times \{n\})$ を $A_1 \uplus \cdots \uplus A_n$ と表記する．なお，文脈から明らかな場合，直和の添字を省略し，$a \in A_i$ に対して，$a \in A_1 \uplus \cdots \uplus A_n$ と表記する．
- 集合 A の B との**差集合** $\{a \in A \mid a \notin B\}$ を $A \setminus B$ と表記する．

□

定義 2 (文字列に対する表記). 有限集合を**アルファベット** (*alphabet*) と呼び，その要素を**記号** (*symbol*) と呼ぶ．アルファベット Σ について，$\Sigma^* = \bigcup_{n \in \mathbb{N}} \Sigma^n$ と定める．この時，$\alpha \in \Sigma^*$ を文字列と呼ぶ．

また，以下の表記を用いる．

- 文字列 $(\sigma_1, \ldots, \sigma_n) \in \Sigma^n$ を $\sigma_1 \cdots \sigma_n$ と表記する．
- 文字列 $\alpha = \sigma_1 \cdots \sigma_n \in \Sigma^*$ について，その長さ n を $|\alpha|$ と表記する．
- 文字列 $\alpha, \beta \in \Sigma^*$ について，以下を満たす時 $\alpha \sqsubseteq \beta$ と表記する．

$$\exists \beta' \in \Sigma^* \,.\, \alpha\beta' = \beta$$

□

定義 3 (述語に対する表記). 集合 A について，$P \subseteq A$ を述語と呼ぶ．

また，以下の表記を用いる．

- 述語 $P \subseteq A$ について，$a \in P$ の時 $P(a)$ と表記する．

□

定義 4 (関係に対する表記). 集合 A, B について，$R \subseteq A \times B$ を関係と呼ぶ．また，

$$A \rightharpoonup B \stackrel{\text{def}}{=} \{R \in \mathcal{P}(A \times B) \mid \forall x \in A, (x, y_1), (x, y_2) \in R \,.\, y_1 = y_2\}$$

と定義する．$f : A \rightharpoonup B$ を A から B への**部分関数** (*partial function*) と呼ぶ．さらに，

$$A \to B \stackrel{\text{def}}{=} \{f : A \rightharpoonup B \mid \forall x \in A \,.\, \exists y \in B \,.\, (x, y) \in f\}$$

と定義する．$f : A \to B$ を (**全**) **関数** (*function*) と呼ぶ．

また，以下の表記を用いる．

- 関係 $R \subseteq A \times B$ について，$(a, b) \in R$ の時 $a \, R \, b$ と表記する．
- 部分関数 $f : A \rightharpoonup B$ について，$(a, b) \in f$ の時 $f(a) = b$ と表記する．
- 関係 $R_1 \subseteq A \times B$，$R_2 \subseteq B \times C$ について，その合成 $\{(x, z) \in A \times C \mid \exists y \in B \,.\, (x, y) \in R_1, (y, z) \in R_2\}$ を $R_1; R_2$ または $R_2 \circ R_1$ と表記する．

- 関係のクラス F, G について，その合成のクラス $\{R_1; R_2 \mid R_1 : A \to B \in F, R_2 : B \to C \in G\}$ を $F; G$ または $G \circ F$ と表記する．
- 関係 $R \subseteq A \times B$ について，$X \subseteq A$ に対する**像** $\{b \in B \mid \exists x \in X \,.\, (x, b) \in R\}$ を $R[X]$，単なる像 $R[A]$ を $\operatorname{Im} R$ と表記する．
- $Y \subseteq B$ に対する**逆像** $\{a \in A \mid \exists y \in Y \,.\, (a, y) \in R\}$ を $R^{-1}[Y]$ と表記する．
- 関係 $R \subseteq A \times B$ について，$S \subseteq A$ への制限 $\{(x, y) \mid (x, y) \in R, x \in S\} \subseteq S \times B$ を $R|_S$ と表記する．
- $a \in A$，$b \in B$ についてその組 (a, b) を $a \mapsto b$，関数 $f : A \to B$ を $x \mapsto f(x)$ と表記する．

□

定義 5 (族に対する表記). クラス I について，その要素で添字付けられた対象の列 $\{a_i\}_{i \in I}$ を族と呼ぶ．

なお，$(a_1, \ldots, a_n) = \{a_i\}_{i \in [n]}$ であり，また族の対象がある集合 Y の要素である時，X で添字付けられた族は関数 $f : X \to Y = \{f(x)\}_{x \in X} = x \mapsto f(x)$ である．

また，以下の表記を用いる．

- 族のクラス $\{\{a_i\}_{i \in I} \mid \forall i \in I, a_i \in A_i\}$ を $\prod_{i \in I} A_i$ と表記する．
- 集合の族 $A = \{A_i\}_{i \in I}$ について，以下を満たす時 A は**互いに素** (*pairwise disjoint*) であるという．

$$\forall i_1, i_2 \in I, i_1 \neq i_2 \,.\, A_{i_1} \cap A_{i_2} = \emptyset$$

- 集合 $I = \{i_1, \ldots, i_n\}$ について，集合の族 $A = \{A_i\}_{i \in I}$ を $\{i_1 \mapsto A_{i_1}, \ldots, i_n \mapsto A_{i_n}\}$ と表記する．

□

また，幾つかの数値に関する演算子を導入する．

定義 6 (基本的な演算子). $X \in \mathcal{P}_{fin}(\mathbb{N})$ について，$\max X$ を以下を満たす最小の自然数 m として定義する．

$$\forall x \in X \,.\, x \leq m$$

特に，$\max \emptyset = 0$ である．また，$X \neq \emptyset$ の時，$\min X$ を以下を満たす最大の自然数 m として定義する．

$$\forall x \in X \,.\, m \leq x$$

さらに，$\exp(n) \stackrel{\text{def}}{=} 2^n$，$\log(n)$ は $\exp(\log(n)) = n$ を満たす実数として定義する． □

構文要素として使用するための変数記号を導入する．

定義 7 (変数記号). 以下の変数記号の集合を導入する．

- $\mathrm{X}_n \stackrel{\mathrm{def}}{=} \{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$
- $\mathrm{Y}_n \stackrel{\mathrm{def}}{=} \{\mathrm{y}_1, \ldots, \mathrm{y}_n\}$
- $\mathrm{Z}_n \stackrel{\mathrm{def}}{=} \{\mathrm{z}_1, \ldots, \mathrm{z}_n\}$

また，w も変数記号として導入する． □

これらの変数記号は，メタ変数ではなく，それぞれがひとつの具体的なオブジェクトであることに注意せよ．これらは後ほど，穴付き木における穴や，木変換器における構文要素の一部として用いるため導入している．

証明のため，**数学的帰納法** (*mathematical induction*) の原理を導入する．

原理 8 (数学的帰納法). 述語 $P \subseteq \mathbb{N}$ について，以下が成り立つ時，任意の $n \in \mathbb{N}$ について $P(n)$ である．

IB　$P(0)$ が成り立つ．
IS　$n \in \mathbb{N}$ について，$P(n)$ ならば $P(n+1)$ が成り立つ．

この時，P は数学的帰納法により示されるという． □

なお，**帰納法の仮定** (*induction hypothesis*) を，i.h. と表記する．

決定不能問題として，*Post Correspondence Problem* (*PCP*) が知られている．本論文では，決定不能性を示す際，PCP に帰着させることでそれを示す．

定義 9 (Post Correspondence Problem (PCP)). $|\Sigma| \geq 2$ を満たすアルファベット Σ，自然数 $n \in \mathbb{N}$ について，$(\alpha_1, \ldots, \alpha_n), (\beta_1, \ldots, \beta_n) \in (\Sigma^*)^n$ が与えられるとする．この時，以下を満たす $i_1 \cdots i_k \in [n]^* \setminus \{\epsilon\}$ を答える問題を Post Correspondence Problem と呼ぶ．

$$\alpha_{i_1} \cdots \alpha_{i_k} = \beta_{i_1} \cdots \beta_{i_k}$$

□

定理 10 ([Pos46]). PCP は，決定不能． □

指数関数は一次関数より増大が速いことが知られている．

補題 11. 以下を満たす $a, b \in \mathbb{N}$ は存在しない．

$$\forall n \in \mathbb{N} \,.\, \exp(n) \leq an + b$$

□

証明. 任意の $a, b \in \mathbb{N}$ について，$f(n) = an + b - \exp(n)$ とする．この時，任意の $n \in \mathbb{N}$ について，

$$\begin{aligned} f(n+1) - f(n) &= a(n+1) + b - \exp(n+1) - an - b + \exp(n) \\ &= a - 2\exp(n) + \exp(n) \\ &= a - \exp(n) \end{aligned}$$

となり，$a \leq \exp(m)$ となる m について $n > m$ を考えると，

$$\begin{aligned} f(n+1) - f(n) &= a - \exp(n) \\ &\leq \exp(m) - \exp(n) \\ &< 0 \end{aligned}$$

となる．ここから，$n > m$ において $f(n)$ は単調減少より，$f(k) < 0$ となる k が存在し，

$$\exp(k) > ak + b$$

となる．よって，題意は示された． ■

2.2 型付き木

S で添字付けられた集合の族において，互いに素であるものを，**型付き集合** (*many-sorted set*) と呼ぶ．

定義 12 (型付き集合). 基本型のアルファベット S について，互いに素な集合の族 $X = \{X_s\}_{s \in S}$ を，S-型付き集合と呼ぶ． □

また，アルファベットについても，型を付けたものを**型付きアルファベット** (*many-sorted ranked alphabet*) と呼ぶ．

定義 13 (型付きアルファベット). 基本型のアルファベット S について，アルファベット Σ と関数 $\mathrm{arity}_\Sigma : \Sigma \to S^* \times S$ の組 $\langle \Sigma, \mathrm{arity}_\Sigma \rangle$ を，S-型付きアルファベットと呼ぶ． arity_Σ が文脈から分かる場合，Σ を単に型付きアルファベットと呼ぶこともある．

また，以下の表記を用いる．

- $n \in \mathbb{N}$，$s_1, \ldots, s_n, s \in S$ について，型の集合 $\{\sigma \in \Sigma \mid \mathrm{arity}_\Sigma(\sigma) = ((s_1, \ldots, s_n), s)\}$ を $(s_1, \ldots, s_n) \twoheadrightarrow_\Sigma s$ と表記する．
- 型付きアルファベット Σ について，関数 $\mathrm{sort}_\Sigma : \Sigma \to S$ を以下を満たすものとして定義する．

$$\forall \sigma \in \Sigma \,.\, \mathrm{arity}_\Sigma(\sigma) = ((s_1, \ldots, s_n), s) \implies \mathrm{sort}_\Sigma(\sigma) = s$$

- 型付きアルファベット Σ について，関数 $\mathrm{rank}_\Sigma : \Sigma \to \mathbb{N}$ を以下を満たすものとして定義する．

$$\forall \sigma \in \Sigma \,.\, \mathrm{arity}_\Sigma(\sigma) = ((s_1, \ldots, s_n), s) \implies \mathrm{rank}_\Sigma(\sigma) = n$$

- 型付きアルファベット Σ，自然数 $n \in \mathbb{N}$ について，集合 $\{\sigma \in \Sigma \mid \mathrm{rank}_\Sigma(\sigma) = n\}$ を $\Sigma^{(n)}$ と表記する．$\sigma \in \Sigma^{(n)}$ を満たす時，$\sigma^{(n)} \in \Sigma$ と表記する．
- 型付きアルファベット Σ について，ランクの最大値 $\max\{\mathrm{rank}_\Sigma(\sigma) \mid \sigma \in \Sigma\}$ を $\mathrm{maxrank}(\Sigma)$ と表記する．

□

例 14. 集合 $S = \{\mathrm{L}, \mathrm{N}\}$，アルファベット $\Sigma = \{1, +, \mathrm{Cons}, \mathrm{Nil}\}$ について，$\mathrm{arity}_\Sigma : \Sigma \to S^* \times S$ を以下を満たすように定義する．

$$\begin{aligned}
\mathrm{arity}_\Sigma(1) &= ((), \mathrm{N}) \\
\mathrm{arity}_\Sigma(+) &= ((\mathrm{N}, \mathrm{N}), \mathrm{N}) \\
\mathrm{arity}_\Sigma(\mathrm{Cons}) &= ((\mathrm{N}, \mathrm{L}), \mathrm{L}) \\
\mathrm{arity}_\Sigma(\mathrm{Nil}) &= ((), \mathrm{L})
\end{aligned}$$

この時，$\langle \Sigma, \mathrm{arity}_\Sigma \rangle$ は S-型付きアルファベットで，

$$\begin{aligned}
1 &: () \rightarrowtriangle_\Sigma \mathrm{N} \\
\mathrm{Cons} &: (\mathrm{N}, \mathrm{L}) \rightarrowtriangle_\Sigma \mathrm{L}
\end{aligned}$$

となる． □

型付きアルファベットにおいて，単項な記号しか含んでいないものを**単項型付きアルファベット** (*monadic many-sorted ranked alphabet*) と呼ぶ．

定義 15 (単項型付きアルファベット). S-型付きアルファベット Σ が単項であるとは，$\Sigma = \bigcup_{n \leq 1} \Sigma^{(n)}$ であることをいう． □

なお，型付き集合において，有限なものは単項型付きアルファベットとみなせる．型付きアルファベットから，帰納的にその型を満たすよう構築された木を**型付き木** (*many-sorted ranked tree*) と呼ぶ．

定義 16 (型付き木). S-型付きアルファベット Σ と S-型付き集合 $X = \{X_s\}_{s \in S}$ について，S-型付き木の集合の族 $T_\Sigma(X) = \{T_\Sigma(X)_s\}_{s \in S}$ は，以下のように帰納的に定義された族 $\{A_s\}_{s \in S}$ である．

IB1　$s \in S$，$x \in X_s$ について，$x \in A_s$．

IB2　$s \in S$，$\sigma : () \rightarrowtriangle_\Sigma s$ について，$\sigma \in A_s$．

IS　$n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \twoheadrightarrow_\Sigma s$，$t_1 \in A_{s_1}, \ldots, t_n \in A_{s_n}$ について，$\sigma(t_1, \ldots, t_n) \in A_s$.

また，以下の表記を用いる．

- $\bigcup_{s \in S} T_\Sigma(X)_s$ を $\hat{T}_\Sigma(X)$，$T_\Sigma(\{\emptyset\}_{s \in S})$ を T_Σ，$\hat{T}_\Sigma(\{\emptyset\}_{s \in S})$ を $\hat{T}_\Sigma$ と表記する．
- $n \in \mathbb{N}$ について，$\underbrace{\sigma(\cdots\sigma}_{n \text{ 項}}(t)\cdots)$ を $\sigma^n(t)$ と表記する．特に $\sigma^0(t) = t$ である．

□

例 17. 例 14 の集合 S，S-型付きアルファベット Σ，S-型付き集合 $X = \{\mathrm{N} \mapsto \{n\}, \mathrm{L} \mapsto \{l\}\}$ について，型付き木 $T_\Sigma(X)$ を考える．この時，

$$\begin{array}{ll} \mathrm{Cons}(+(1, +(n, 1)), \mathrm{Cons}(1, l)) & \in T_\Sigma(X)_\mathrm{L} \\ \mathrm{Cons}(+(1, 1), \mathrm{Nil}) & \in T_\Sigma(X)_\mathrm{L} \\ +(n, n) & \in T_\Sigma(X)_\mathrm{N} \end{array}$$

となる．□

型付き木のように帰納的に定義された構造に対しては，**構造的帰納法** (*structural induction*) の原理が成り立つ．

原理 18 (型付き木に関する構造的帰納法). S-型付きアルファベット Σ と S-型付き集合 $X = \{X_s\}_{s \in S}$，述語 $P \in \mathcal{P}(\hat{T}_\Sigma(X))$ について，

IB1　$s \in S$，$x \in X_s$ について，$P(x)$ が成り立つ．

IB2　$s \in S$，$\sigma : () \twoheadrightarrow_\Sigma s$ について，$P(\sigma)$ が成り立つ．

IS　$n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \twoheadrightarrow_\Sigma s$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，$P(t_1), \ldots, P(t_n)$ ならば $P(\sigma(t_1, \ldots, t_n))$ が成り立つ．

が成り立つ時，任意の $t \in \hat{T}_\Sigma(X)$ について $P(t)$ である．この時，P は型付き木に関する構造的帰納法により示されるという．□

また，構造的帰納法を元に拡張した，**同時帰納法** (*simultaneous induction*) の原理も成り立つ．同時帰納法は，相互再帰的な性質を示すのに有用である．

原理 19 (型付き木に関する同時帰納法). 集合 A，集合の族 $\{B_\sigma\}_{\sigma \in \Sigma}$，
述語の族 $K \in \prod_{a \in A} \mathcal{P}(\hat{T}_\Sigma(X))$，$L \in \prod_{\sigma^{(n)} \in \Sigma, b \in B_\sigma} \mathcal{P}((\hat{T}_\Sigma(X))^n)$ について，

IB1　$a \in A$，$s \in S$，$x \in X_s$ について，$K_a(x)$ が成り立つ．

IB2　$s \in S$，$\sigma : () \twoheadrightarrow_\Sigma s$，$b \in B_\sigma$ について，$L_{\sigma,b}(\epsilon)$ が成り立つ．

IS1　$a \in A$，$n \in \mathbb{N}$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \twoheadrightarrow_\Sigma s$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，任意の $b \in B_\sigma$ について $L_{\sigma,b}((t_1, \ldots, t_n))$ ならば，$K_a(\sigma(t_1, \ldots, t_n))$ が成り

立つ．

IS2　$n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \twoheadrightarrow_\Sigma s$，$b \in B_\sigma$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，任意の $a_1, \ldots, a_n \in A$ について $K_{a_1}(t_1), \ldots, K_{a_n}(t_n)$ ならば，$L_{\sigma,b}((t_1, \ldots, t_n))$ が成り立つ．

が成り立つ時，任意の $a \in A$，$t \in \hat{T}_\Sigma(X)$ について $K_a(t)$ である．この時，K_a はランク付き木に関する同時帰納法により示されるという． □

以下では，木に関する重要な演算を幾つか導入する．まず，**パス** (*path*) を導入する．パスは木の**ノード** (*node*) を指す自然数の列である．

定義 20 (木のパス). 関数 $\mathrm{paths} : \hat{T}_\Sigma(X) \to \mathcal{P}(\mathbb{N}^*)$ を，以下のように構造的帰納法で定義する．

IB1　$s \in S, x \in X_s$ について，$\mathrm{paths}(x) \stackrel{\mathrm{def}}{=} \{\epsilon\}$．

IB2　$s \in S, \sigma : () \twoheadrightarrow_\Sigma s$ について，$\mathrm{paths}(\sigma) \stackrel{\mathrm{def}}{=} \{\epsilon\}$．

IS　$n \geq 1, s_1, \ldots, s_n, s \in S, \sigma : (s_1, \ldots, s_n) \twoheadrightarrow_\Sigma s, t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，$\mathrm{paths}(\sigma(t_1, \ldots, t_n)) \stackrel{\mathrm{def}}{=} \{\epsilon\} \cup \bigcup_{i \in [n]} \{i\pi \mid \pi \in \mathrm{paths}(t_i)\}$．

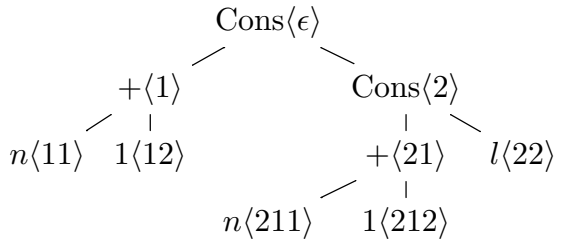
この時，$\pi \in \mathrm{paths}(t)$ を t のパスと呼ぶ． □

あるノードのパスは木の**根** (*root*) からそのノードに行き着く道順を示している．

例 21. 例 17 の $T_\Sigma(X)$ について，$t \in T_\Sigma(X)_{\mathrm{L}}$ を以下のように定義する．

$$t \stackrel{\mathrm{def}}{=} \mathrm{Cons}(+(n, 1), \mathrm{Cons}(+(n, 1), l))$$

この時，t の各ノードを指すパスは，以下のようになる．



⟨ と ⟩ で囲まれた部分が，各ノードを指すパスである．よって，パスの集合は $\mathrm{paths}(t) = \{\epsilon, 1, 11, 12, 2, 21, 211, 212, 22\}$ となる． □

また，木に対して**部分木** (*subtree*) という概念を導入する．部分木は，ある木に含まれる木のことである．

定義 22 (部分木). 木 $t \in \hat{T}_\Sigma(X)$ について，関数 $\mathrm{subtree}_t : \mathrm{paths}(t) \to \hat{T}_\Sigma(X)$ を，以下のように構造的帰納法で定義する．

IB1　$s \in S$，$x \in X_s$ について，$\mathrm{subtree}_x(\epsilon) \stackrel{\mathrm{def}}{=} x$.

IB2　$s \in S$，$\sigma : () \to_\Sigma s$ について，$\mathrm{subtree}_\sigma(\epsilon) \stackrel{\mathrm{def}}{=} \sigma$.

IS　$n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \to_\Sigma s$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，

$$\mathrm{subtree}_{\sigma(t_1,\ldots,t_n)}(\pi) \stackrel{\mathrm{def}}{=} \begin{cases} \sigma(t_1, \ldots, t_n) & (\pi = \epsilon) \\ \mathrm{subtree}_{t_i}(\pi') & (\pi = i\pi') \end{cases} .$$

この時，$\pi \in \mathrm{paths}(t)$ について，$\mathrm{subtree}_t(\pi)$ を t の π での部分木と呼ぶ. □

subtree はあるノードを指すパスを受け取り，そのノードを根とする部分木を返す.

例 **23.** 例 21 の t について，

$$\begin{aligned} \mathrm{subtree}_t(\epsilon) &= t \\ \mathrm{subtree}_t(21) &= +(n, 1) \\ \mathrm{subtree}_t(22) &= l \end{aligned}$$

となる. □

構造的帰納法は，部分木全体に対しての i.h. が使えるよう拡張することができる.

定理 24 (型付き木の部分木全体に関する構造的帰納法). S-型付きアルファベット Σ，S-型付き集合 $X = \{X_s\}_{s \in S}$，述語 $P \in \mathcal{P}(\hat{T}_\Sigma(X))$ について，述語 P' を以下のようにおく.

$$P'(t) \stackrel{\mathrm{def}}{\iff} \forall \pi \in \mathrm{paths}(t) \,.\, P(\mathrm{subtree}_t(\pi))$$

以下が成り立つ時，任意の $t \in \hat{T}_\Sigma(X)$ について $P(t)$ である.

IB1　$s \in S$，$x \in X_s$ について，$P(x)$ が成り立つ.

IB2　$s \in S$，$\sigma : () \to_\Sigma s$ について，$P(\sigma)$ が成り立つ.

IS　$n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \to_\Sigma s$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，$P'(t_1), \ldots, P'(t_n)$ ならば $P(\sigma(t_1, \ldots, t_n))$ が成り立つ.

□

証明. 任意の $t \in \hat{T}_\Sigma(X)$ について $P'(t)$ を，構造的帰納法で示す.

IB1　$s \in S$，$x \in X_s$ について，$P(x)$ より $P'(x)$.

IB2　$s \in S$，$\sigma : () \to_\Sigma s$ について，$P(\sigma)$ より $P'(\sigma)$.

IS　$n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \to_\Sigma s$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，i.h. より $P(\sigma(t_1, \ldots, t_n))$ であり，$\sigma(t_1, \ldots, t_n)$ 以外の部分木に対しても i.h. よりただちに P が成り立つ．よって，$P'(\sigma(t_1, \ldots, t_n))$.

■

よって，構造的帰納法について，特に断りなく部分木に対しての i.h. を使用することがある．さらに，木に対して**ラベル** (*label*) の概念を導入する．ラベルは，木のノードを構築する記号のことである．

定義 25 (木のラベル). 木 $t \in \hat{T}_\Sigma(X)$ について，関数 $\mathrm{label}_t : \mathrm{paths}(t) \to \Sigma \cup X$ を，以下のように構造的帰納法で定義する．

IB1 $s \in S$，$x \in X_s$ について，$\mathrm{label}_x(\epsilon) \stackrel{\mathrm{def}}{=} x$.

IB2 $s \in S$，$\sigma : () \to_\Sigma s$ について，$\mathrm{label}_\sigma(\epsilon) \stackrel{\mathrm{def}}{=} \sigma$.

IS $n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \to_\Sigma s$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，

$$\mathrm{label}_{\sigma(t_1,\ldots,t_n)}(\pi) \stackrel{\mathrm{def}}{=} \begin{cases} \sigma & (\pi = \epsilon) \\ \mathrm{label}_{t_i}(\pi') & (\pi = i\pi') \end{cases} .$$

この時，$\pi \in \mathrm{paths}(t)$ について，$\mathrm{label}_t(\pi)$ を t の π でのラベルと呼ぶ． □

label はあるノードを指すパスを受け取り，そのノードのラベルを返す．

例 26. 例 21 の t について，

$$\begin{aligned} \mathrm{label}_t(\epsilon) &= \mathrm{Cons} \\ \mathrm{label}_t(21) &= + \\ \mathrm{label}_t(211) &= n \\ \mathrm{label}_t(22) &= l \end{aligned}$$

となる． □

また，**出現** (*occurrence*) の概念を導入する．出現は，部分木が現れるような木のパスのことである．

定義 27 (木での出現). 木 $t \in \hat{T}_\Sigma(X)$ について，関数 $\mathrm{occ}_t : \hat{T}_\Sigma(X) \to \mathcal{P}(\mathrm{paths}(t))$ を，以下のように定義する．

$$\mathrm{occ}_t \stackrel{\mathrm{def}}{=} \eta \mapsto \{\pi \in \mathrm{paths}(t) \mid \mathrm{subtree}_t(\pi) = \eta\}$$

この時，$\eta \in \hat{T}_\Sigma(X)$ について，$\pi \in \mathrm{occ}_t(\eta)$ を t における η の出現と呼ぶ． □

occ は部分木に対してその全出現を返す．

例 28. 例 21 の t について，

$$\mathrm{occ}_t(+(n, 1)) = \{1, 21\}$$

$$\mathrm{occ}_t(1) = \{12, 212\}$$
$$\mathrm{occ}_t(t) = \{\epsilon\}$$
$$\mathrm{occ}_t(+(1,1)) = \emptyset$$

となる．□

また，**ラベル出現**の概念を導入する．ラベル出現は，ある記号を持つノードを指す，木のパスのことである．

定義 29 (木でのラベル出現). 木 $t \in \hat{T}_\Sigma(X)$ について，関数 $\mathrm{occ}_t : \Sigma \cup X \to \mathcal{P}(\mathrm{paths}(t))$ を，以下のように定義する．

$$\mathrm{occ}_t \stackrel{\mathrm{def}}{=} \eta \mapsto \{\pi \in \mathrm{paths}(t) \mid \mathrm{label}_t(\pi) = \eta\}$$

この時，$\sigma \in \Sigma \cup X$ について，$\pi \in \mathrm{occ}_t(\eta)$ を t における σ のラベル出現と呼ぶ．□

occ は記号に対してその全ラベル出現を返す．なお，$\sigma \in \Sigma^{(0)} \cup X$ について，σ の出現とラベル出現の集合は一致する．

例 30. 例 21 の t について，

$$\mathrm{occ}_t(+) = \{1, 21\}$$
$$\mathrm{occ}_t(1) = \{12, 212\}$$
$$\mathrm{occ}_t(\mathrm{Nil}) = \emptyset$$

となる．□

さらに，**高さ** (*height*) の概念を導入する．木の高さとは，根から一番遠い葉への距離のことである．

定義 31 (木の高さ). 関数 $\mathrm{height} : \hat{T}_\Sigma(X) \to \mathbb{N}$ を，以下のように構造的帰納法で定義する．

IB1　$s \in S$，$x \in X_s$ について，$\mathrm{height}(x) \stackrel{\mathrm{def}}{=} 0$.

IB2　$s \in S$，$\sigma : () \to_\Sigma s$ について，$\mathrm{height}(\sigma) \stackrel{\mathrm{def}}{=} 1$.

IS　$n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \to_\Sigma s$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，$\mathrm{height}(\sigma(t_1, \ldots, t_n)) \stackrel{\mathrm{def}}{=} 1 + \max\{\mathrm{height}(t_i) \mid i \in [n]\}$.

この時，$t \in \hat{T}_\Sigma(X)$ について，$\mathrm{height}(t)$ を t の高さと呼ぶ．□

木の高さは，木のパスの中で最長なものの長さに，1 を足したものと一致する．

命題 32. 任意の $t \in \hat{T}_\Sigma$ について，$\mathrm{height}(t) = 1 + \max\{|\pi| \mid \pi \in \mathrm{paths}(t)\}$. □

証明. 構造的帰納法で示す．

IB1　適用不能.

IB2　$s \in S$，$\sigma : () \rightarrowtriangle_\Sigma s$ について，

$$\mathrm{height}(\sigma) = 1 = 1 + \max\{|\epsilon|\} = 1 + \max\{|\pi| \mid \pi \in \mathrm{paths}(\sigma)\}$$

より正しい.

IS　$n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \rightarrowtriangle_\Sigma s$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，

$$\begin{aligned}
\mathrm{height}(\sigma(t_1, \ldots, t_n)) &= 1 + \max\{\mathrm{height}(t_i) \mid i \in [n]\} \\
&= 1 + \max\{1 + \max\{|\pi| \mid \pi \in \mathrm{paths}(t_i)\} \mid i \in [n]\} \quad (\because \text{i.h.}) \\
&= 1 + \max\{\max\{|i\pi| \mid \pi \in \mathrm{paths}(t_i)\} \mid i \in [n]\} \\
&= 1 + \max\{|\epsilon|\} \cup \{|i\pi| \mid i \in [n], \pi \in \mathrm{paths}(t_i)\} \\
&= 1 + \max\{|\pi| \mid \pi \in \mathrm{paths}(\sigma(t_1, \ldots, t_i))\}
\end{aligned}$$

より正しい.

■

また，**サイズ** (*size*) の概念を導入する．木のサイズとは，ノードの数のことである.

定義 33 (木のサイズ). 関数 $\mathrm{size} : \hat{T}_\Sigma(X) \to \mathbb{N}$ を，以下のように構造的帰納法で定義する.

IB1　$s \in S, x \in X_s$ について，$\mathrm{size}(x) \stackrel{\mathrm{def}}{=} 0$.

IB2　$s \in S, \sigma : () \rightarrowtriangle_\Sigma s$ について，$\mathrm{size}(\sigma) \stackrel{\mathrm{def}}{=} 1$.

IS　$n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \rightarrowtriangle_\Sigma s$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，$\mathrm{size}(\sigma(t_1, \ldots, t_n)) \stackrel{\mathrm{def}}{=} 1 + \sum_{i \in [n]} \mathrm{size}(t_i)$.

この時，$t \in \hat{T}_\Sigma(X)$ について，$\mathrm{size}(t)$ を t のサイズと呼ぶ. □

木のサイズは，木のパスの数と一致する.

命題 34. 任意の $t \in \hat{T}_\Sigma$ について，$\mathrm{size}(t) = |\mathrm{paths}(t)|$. □

証明. 構造的帰納法で示す.

IB1　適用不能.

IB2　$s \in S$，$\sigma : () \rightarrowtriangle_\Sigma s$ について，

$$\mathrm{size}(\sigma) = 1 = |\{\epsilon\}| = |\mathrm{paths}(\sigma)|$$

より正しい.

IS　$n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \to_\Sigma s$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，

$$\begin{aligned}
\mathrm{size}(\sigma(t_1, \ldots, t_n)) &= 1 + \sum_{i \in [n]} \mathrm{size}(t_i) \\
&= 1 + \sum_{i \in [n]} |\mathrm{paths}(t_i)| && (\because \text{i.h.}) \\
&= |\{\epsilon\}| + \sum_{i \in [n]} |\{i\pi \mid \pi \in \mathrm{paths}(t_i)\}| \\
&= |\{\epsilon\} \cup \{i\pi \mid i \in [n], \pi \in \mathrm{paths}(t_i)\}| \\
&= |\mathrm{paths}(\sigma(t_1, \ldots, t_n))|
\end{aligned}$$

より正しい．

■

さて，ある木のサイズは，その木の高さとその木を構築するアルファベットによって定まる値によって抑えられる．この値とは，最もランクの高い記号だけから作られる木のサイズである．

命題 35. 任意の $t \in \hat{T}_\Sigma(X)$ について，以下が成り立つ．

$$\mathrm{size}(t) \leq \sum_{i=0}^{\mathrm{height}(t)-1} (\mathrm{maxrank}(\Sigma))^i$$

□

証明. 構造的帰納法で示す．なお，スペースの都合上 maxrank を m，height を h と略記する．

IB1　$s \in S$，$x \in X_s$ について，

$$\mathrm{size}(x) = 0 \leq 1 = \sum_{i=0}^{0} (m(\Sigma))^i = \sum_{i=0}^{h(x)-1} (m(\Sigma))^i$$

より正しい．

IB2　$s \in S$，$\sigma : () \to_\Sigma s$ について，

$$\mathrm{size}(\sigma) = 1 = \sum_{i=0}^{0} (m(\Sigma))^i = \sum_{i=0}^{h(\sigma)-1} (m(\Sigma))^i$$

より正しい．

IS　$n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \to_\Sigma s$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，$t = \sigma(t_1, \ldots, t_n)$ として，

$$\mathrm{size}(t) = 1 + \sum_{i \in [n]} \mathrm{size}(t_i)$$

$$
\begin{aligned}
&\leq 1 + \sum_{j\in[n]} \sum_{j=0}^{h(t_i)-1} (m(\Sigma))^j && (\because \text{i.h.})\\
&\leq 1 + n \sum_{j=0}^{h(t)-2} (m(\Sigma))^j && (\because \forall i \in [n] \,.\, h(t_i) \leq h(t) - 1)\\
&\leq (m(\Sigma))^0 + m(\Sigma) \sum_{j=0}^{h(t)-2} (m(\Sigma))^j && (\because n \leq m(\Sigma))\\
&= \sum_{j=0}^{h(t)-1} (m(\Sigma))^j
\end{aligned}
$$

より正しい.

■

また，ある木の高さは，その木のサイズにより抑えられる.

命題 36. 任意の $t \in \hat{T}_\Sigma(X)$ について，$\mathrm{height}(t) \leq \mathrm{size}(t)$. □

証明. 構造的帰納法で示す.

IB1 $s \in S$，$x \in X_s$ について，

$$\mathrm{height}(x) = 0 = \mathrm{size}(x)$$

より正しい.

IB2 $s \in S$，$\sigma : () \to_\Sigma s$ について，

$$\mathrm{height}(\sigma) = 1 = \mathrm{size}(\sigma)$$

より正しい.

IS $n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \to_\Sigma s$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，

$$
\begin{aligned}
\mathrm{height}(\sigma(t_1, \ldots, t_n)) &= 1 + \max\{\mathrm{height}(t_i) \mid i \in [n]\}\\
&\leq 1 + \max\{\mathrm{size}(t_i) \mid i \in [n]\} && (\because \text{i.h.})\\
&\leq 1 + \sum_{i=0}^{n} \mathrm{size}(t_i)\\
&= \mathrm{size}(\sigma(t_1, \ldots, t_n))
\end{aligned}
$$

より正しい.

■

なお，木の高さとサイズは，単項なアルファベットから木が作られる場合，一致する．

命題 37. 単項型付きアルファベット Σ について，以下が成り立つ．

$$\forall t \in \hat{T}_\Sigma(X) \,.\, \mathrm{height}(t) = \mathrm{size}(t)$$

□

証明. $\mathrm{maxrank}(\Sigma) \leq 1$ より，任意の $t \in \hat{T}_\Sigma(X)$ について，命題 35，命題 36 より，

$$\mathrm{height}(t) \leq \mathrm{size}(t) \leq \sum_{i=0}^{\mathrm{height}(t)-1} (\mathrm{maxrank}(\Sigma))^i \leq \sum_{i=0}^{\mathrm{height}(t)-1} 1 = \mathrm{height}(t)$$

となる．よって，題意は示された． ■

さらに，**代入** (*substitution*) という概念を導入する．代入は，木の変数部分に別の木をあてがった木を作成する変換のことである．

定義 38 (木に対する代入). 関数の族 $f : \{X_s \to T_\Sigma(Y)_s\}_{s \in S}$ について，$[f] : \hat{T}_\Sigma(X) \to \hat{T}_\Sigma(Y)$ を，以下のように構造的帰納法で定義する．

IB1 $s \in S$，$x \in X_s$ について，$[f](x) \stackrel{\mathrm{def}}{=} f(x)$.

IB2 $s \in S$，$\sigma : () \twoheadrightarrow_\Sigma s$ について，$[f](\sigma) \stackrel{\mathrm{def}}{=} \sigma$.

IS $n \geq 1$，$s_1, \ldots, s_n, s \in S$，$\sigma : (s_1, \ldots, s_n) \twoheadrightarrow_\Sigma s$，$t_1 \in T_\Sigma(X)_{s_1}, \ldots, t_n \in T_\Sigma(X)_{s_n}$ について，$[f](\sigma(t_1, \ldots, t_n)) \stackrel{\mathrm{def}}{=} \sigma([f](t_1), \ldots, [f](t_n))$.

この時 $[f]$ を代入と呼ぶ．

また，$X = \{x_1, \ldots, x_n\}$ の時，$[f]$ を $[x_1 \leftarrow f(x_1), \ldots, x_n \leftarrow f(x_n)]$ または $[x_i \leftarrow f(x_i)]_{i \in [n]}$ と表記する．また，$[f](x)$ を $x[f]$ と表記する． □

例 39. 例 21 の t について，

$$t[n \leftarrow 1, l \leftarrow \mathrm{Cons}(1, \mathrm{Nil})] = \mathrm{Cons}(+(1,1), \mathrm{Cons}(+(1,1), \mathrm{Cons}(1, \mathrm{Nil})))$$

となる． □

あるパスの部分木を，別の木に置換する操作も導入する．

定義 40 (木のパスに対する代入). $t \in \hat{T}_\Sigma(X)$，$\pi \in \mathrm{paths}(t)$，$\eta \in T_\Sigma(X)_{\mathrm{sort}_\Sigma(\mathrm{label}_t(\pi))}$ について，$t[\pi \leftarrow \eta]$ を以下を満たす $t' \in \hat{T}_\Sigma(X)$ として定義する．

$$\forall w \in \mathrm{paths}(t) \,.\, \begin{cases} \mathrm{subtree}_{t'}(w) = s & (w = \pi) \\ \pi \sqsubseteq w \vee \mathrm{label}_t(w) = \mathrm{label}_{t'}(w) & (\text{otherwise}) \end{cases}$$

□

例 41. 例 21 の t について，

$$t[2 \leftarrow \mathrm{Nil}] = \mathrm{Cons}(+(n, 1), \mathrm{Nil})$$

となる． □

さらに，**穴付き木** (*tree with holes*) という概念を導入する．穴付き木は，木の幾つかの葉の部分に，木を当てがえるような木のことである．

定義 42 (穴付き木). 型付き有限集合 Y について，$\hat{T}_\Sigma(X, Y)$ を以下のように定義する．

$$\hat{T}_\Sigma(X, Y) \stackrel{\mathrm{def}}{=} \{t \in \hat{T}_\Sigma(X \uplus Y) \mid \forall y \in Y \,.\, \exists! \pi \in \mathrm{occ}_t(y)\}$$

この時，$t \in \hat{T}_\Sigma(X, Y)$ を Y による穴付き木と呼ぶ．

また，$t \in \hat{T}_\Sigma(X, Y)$，$\eta \in \prod_{y \in Y} T_\Sigma(X)_{\mathrm{sort}_Y(y)}$ について $t[y \leftarrow \eta_y]_{y \in Y} \in \hat{T}_\Sigma(X)$ を $t(\eta)$ と表記する． □

例 43. 例 17 の Σ，X について，$Y = \{\mathrm{N} \mapsto \mathrm{Y}_2, \mathrm{L} \mapsto \emptyset\}$ として，$t \in \hat{T}_\Sigma(X, Y)$ を以下のように定義する．

$$t \stackrel{\mathrm{def}}{=} \mathrm{Cons}(+(n, +(\mathrm{y}_1, \mathrm{y}_2)), \mathrm{Nil})$$

この t について，

$$t(\{\mathrm{y}_1 \mapsto +(1, 1), \mathrm{y}_2 \mapsto n\}) = \mathrm{Cons}(+(n, +(+(1, 1), n)), \mathrm{Nil})$$

□

型付き木について，型が 1 点集合の場合，その木を**ランク付き木** (*ranked tree*) と呼ぶ．

定義 44 (ランク付き木). $\{*\}$-型付きアルファベットをランク付きアルファベット，$\{*\}$-型付き木をランク付き木と呼ぶ．

また，以下の表記を用いる．

- ランク付きアルファベット $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$ について，Σ を $\{\sigma_1^{(\mathrm{rank}_\Sigma(\sigma_1))}, \ldots, \sigma_n^{(\mathrm{rank}_\Sigma(\sigma_n))}\}$ と表記する．
- 型付きアルファベット Σ について，集合 Σ と以下を満たす $\mathrm{arity}_{\bar{\Sigma}}$ で定義されるランク付きアルファベットを $\bar{\Sigma}$ と表記する．

$$\forall \sigma^{(n)} \in \Sigma \,.\, \mathrm{arity}_{\bar{\Sigma}}(\sigma) = (\underbrace{(*, \ldots, *)}_{n \text{ 項}}, *)$$

□

ランク付き木に対して，型を満たすよう型付き木に射影する操作を，以下で定義する．

定義 45 (ランク付き木の射影). 関数の族 $p : \prod_{s_1,s_2\in S} T_\Sigma(X)_{s_1} \to T_\Sigma(X)_{s_2}$ について，関数の族 $\mathrm{rankproj}_i : \prod_{s\in S} \hat{T}_{\bar{\Sigma}}(\bar{X}) \to T_\Sigma(X)_s$ を，以下のように構造的帰納法で定義する．

IB1 $s_1, s_2 \in S$，$x \in X_{s_1}$ について，$\mathrm{rankproj}_{p,s_2}(x) \stackrel{\mathrm{def}}{=} p_{s_1,s_2}(x)$.

IB2 $s_1, s_2 \in S$，$\sigma : () \to_\Sigma s_1$ について，$\mathrm{rankproj}_{p,s_2}(\sigma) \stackrel{\mathrm{def}}{=} p_{s_1,s_2}(\sigma)$.

IS $n \geq 1$，$s_1, \ldots, s_n, s'_1, s'_2 \in S$，$\sigma : (s_1, \ldots, s_n) \to_\Sigma s'_1$，$t_1 \in \hat{T}_{\bar{\Sigma}}(\bar{X}), \ldots, t_n \in \hat{T}_{\bar{\Sigma}}(\bar{X})$ について，

$$\mathrm{rankproj}_{p,s'_2}(\sigma(t_1, \ldots, t_n)) \stackrel{\mathrm{def}}{=} p_{s'_1,s'_2}(\sigma(\mathrm{rankproj}_{p,s_1}(t_1), \ldots, \mathrm{rankproj}_{p,s_n}(t_n))).$$

□

例 46. 例 17 の S，Σ，X について，$p : \prod_{s_1,s_2\in S} T_\Sigma(X)_{s_1} \to T_\Sigma(X)_{s_2}$ を以下のように定義する．

$$\begin{aligned}
p_{s,s} &\stackrel{\mathrm{def}}{=} x \mapsto x && (s \in S)\\
p_{\mathrm{N,L}} &\stackrel{\mathrm{def}}{=} x \mapsto \mathrm{Cons}(x, \mathrm{Nil})\\
p_{\mathrm{L,N}} &\stackrel{\mathrm{def}}{=} x \mapsto \begin{cases} x' & (x = \mathrm{Cons}(x', \ldots))\\ 1 & (x = \mathrm{Nil}) \end{cases}
\end{aligned}$$

この時，$t \stackrel{\mathrm{def}}{=} \mathrm{Cons}(\mathrm{Cons}(+(\mathrm{Cons}(1,1),\mathrm{Nil}),1),1)$ として，

$$\begin{aligned}
\mathrm{rankproj}_{p,\mathrm{L}}(t) &= \mathrm{Cons}(\mathrm{rankproj}_{p,\mathrm{N}}(\mathrm{Cons}(+(\mathrm{Cons}(1,1),\mathrm{Nil}),1), \mathrm{rankproj}_{p,\mathrm{L}}(1)))\\
&= \mathrm{Cons}(\mathrm{rankproj}_{p,\mathrm{N}}(+(\mathrm{Cons}(1,1),\mathrm{Nil})), \mathrm{rankproj}_{p,\mathrm{L}}(1)))\\
&= \cdots\\
&= \mathrm{Cons}(+(1,1), \mathrm{rankproj}_{p,\mathrm{L}}(1)))\\
&= \mathrm{Cons}(+(1,1), \mathrm{Cons}(1,\mathrm{Nil}))\\
\mathrm{rankproj}_{p,\mathrm{N}}(t) &= p_{\mathrm{L,N}}(\mathrm{rankproj}_{p,\mathrm{L}}(t))\\
&= p_{\mathrm{L,N}}(\mathrm{Cons}(+(1,1), \mathrm{Cons}(1,\mathrm{Nil})))\\
&= +(1,1)
\end{aligned}$$

となる． □

2.3 簡約システム

集合とその上の関係の組を，**簡約システム** (*reduction system*) と呼ぶ．

定義 47 (簡約システム). 集合 A とその上の関係 $\Rightarrow \subseteq A^2$ の組 $(A, \Rightarrow)$ を簡約システムと呼ぶ．なお，A が文脈から分かる場合，$\Rightarrow$ を単に簡約システムと呼ぶこともある．

また，以下の表記を用いる．

- $n \in \mathbb{N}$ について，

$$\exists a_0, a_1, \ldots, a_n \in A \,.\, a_0 = a \wedge a_n = b \wedge \forall i \in [n] \,.\, a_{i-1} \Rightarrow a_i$$

  を満たす時 $a \Rightarrow^n b$ と表記する．
- $a \Rightarrow^n b$ となる $n \in \mathbb{N}$ が存在する時，$a \Rightarrow^* b$ と表記する．
- $a \Rightarrow^n b$ となる $n > 0$ が存在する時，$a \Rightarrow^+ b$ と表記する．
- $a \Rightarrow^n b$ となる $n \in \{0, 1\}$ が存在する時，$a \Rightarrow^? b$ と表記する．

特に，$a \Rightarrow^* b$ の時，a を b に簡約する，または a は b に簡約できるという．また，$a \Rightarrow^n b$ について，n を簡約の長さと呼び，この時 a は b に n 回で簡約できるという． □

簡約システムにおいて，簡約できる余地があるものを**可約** (*reducible*) と呼ぶ．もはや簡約できないものを**既約** (*irreducible*) と呼ぶ．

定義 48 (可約). 簡約システム $(A, \Rightarrow)$，$a \in A$ について，

$$\exists b \in A \,.\, a \Rightarrow b$$

を満たす時 a は可約であるという．a が可約でない時，a は既約であるという． □

簡約システムにおいて，ある要素から可能な限り簡約を進め既約になったものを**正規形** (*normal form*) と呼ぶ．

定義 49 (正規形). 簡約システム $(A, \Rightarrow)$，$a \in A$ について，b が $a \Rightarrow^* b$ かつ既約である時，b を a の正規形と呼ぶ．正規形の集合 $\{b \mid b \text{ は } a \text{ の正規形 }\}$ を $\mathrm{nforms}(\Rightarrow, a)$ と表記する．また，a の正規形が一意である時，その正規形を $\mathrm{nf}(\Rightarrow, a)$ と表記する． □

簡約システムにおいて，無限に簡約が可能なものを**無限簡約可能** (ω-*reducible*) と呼ぶ．

定義 50 (無限簡約可能). 簡約システム $(A, \Rightarrow)$，$a \in A$ について，以下を満たす $\eta = \{\eta_n\}_{n \in \mathbb{N}}$ が存在する時 a は無限簡約可能という．

- $a = \eta_0$.
- 任意の $n \in \mathbb{N}$ について，$\eta_n \Rightarrow \eta_{n+1}$.

この時，η を a の無限簡約列と呼ぶ． □

ある要素が正規形を持たない時，その要素は無限簡約可能である．

補題 51. 簡約システム $(A, \Rightarrow)$ について，以下が成り立つ．

- $a \in A$ が正規形を持たない時，a は無限簡約可能．
- $a \in A$ が無限簡約可能でない時，a は正規形を持つ．

□

証明. a が正規形を持たないとする．この時，$\eta = \{\eta_n\}_{n\in\mathbb{N}}$ を，以下のように数学的帰納法で定義する．

IB　$\eta_0 \stackrel{\text{def}}{=} a$ ．この時，η_0 が既約とすると，η_0 は a の正規形となるが，これは仮定に矛盾するため，η_0 は可約.

IS　η_n は可約なため，$\eta_n \Rightarrow \eta'$ を満たす $\eta' \in A$ が存在する．この時，$\eta_{n+1} \stackrel{\text{def}}{=} \eta'$ とおく．η_{n+1} が既約とすると，$a \Rightarrow^{n+1} \eta_{n+1}$ より η_{n+1} は a の正規形となるが，これは仮定に矛盾するため，η_{n+1} は可約.

この η は a の無限簡約列なため，a は無限簡約可能．また，その対偶より $a \in A$ が無限簡約可能でない時，a は正規形を持つ．よって，題意は示された． ■

簡約システムにおいて，無限簡約可能な要素がない時，**停止性** $(terminating)$ を持つという．

定義 52 (停止性). 簡約システム $(A, \Rightarrow)$ について，無限簡約可能な $a \in A$ が存在しない時，$\Rightarrow$ は停止性を持つという． □

停止性を持つ簡約システムにおいて，すべての要素は必ず正規形を持つ.

系 53. 停止性を持つ簡約システム $(A, \Rightarrow)$，$a \in A$ について，a の正規形が存在する． □

証明. a の正規形が存在しないとすると，補題 51 より a は無限簡約可能になるが，これは停止性を持つという条件に矛盾する．よって，題意は示された． ■

簡約システムにおいて，全ての要素の 1 回の簡約結果が全てある要素へ簡約できる時，**局所合流性** $(locally\ confluent)$ を持つという．

定義 54 (局所合流性). 簡約システム $(A, \Rightarrow)$ について，

$$\forall a, b_1, b_2 \in A\ .\ (a \Rightarrow b_1) \wedge (a \Rightarrow b_2) \text{ implies } \exists c \in A\ .\ (b_1 \Rightarrow^* c) \wedge (b_2 \Rightarrow^* c)$$

を満たす時，$\Rightarrow$ は局所合流性を持つという． □

また，簡約システムにおいて，全ての要素の簡約結果が全てある同一の要素へ簡約できる時，**合流性** $(confluent)$ を持つという．

定義 55 (合流性). 簡約システム $(A, \Rightarrow)$ について，

$$\forall a, b_1, b_2 \in A\ .\ (a \Rightarrow^* b_1) \wedge (a \Rightarrow^* b_2) \text{ implies } \exists c \in A\ .\ (b_1 \Rightarrow^* c) \wedge (b_2 \Rightarrow^* c)$$

を満たす時，$\Rightarrow$ は合流性を持つという． □

停止性を持つ簡約システムにおいて，局所合流性を持つことと合流性を持つことは同値である．

補題 56. 停止性を持つ簡約システム $(A, \Rightarrow)$ について，以下の 2 条件は同値である．

- $\Rightarrow$ は局所合流性を持つ．
- $\Rightarrow$ は合流性を持つ．

□

証明. [FV98] の補題 2.5 より． ■

また，停止性と合流性を持つ簡約システムにおいて，正規形は必ず一意に存在する．

補題 57. 停止性と合流性を持つ簡約システム $(A, \Rightarrow)$ において，以下が成り立つ．

$$\forall a \in A \,.\, \exists\, \mathrm{nf}(\Rightarrow, a)$$

□

証明. 停止性より a の正規形は存在する．a の正規形 b_1, b_2 について，合流性より $b_1 \Rightarrow^* c$, $b_2 \Rightarrow^* c$ を満たす c が存在するが，b_1, b_2 は正規形より $c = b_1 = b_2$ となり，a の正規形は一意になる． ■

2.4 木言語と木変換

ランク付き木の集合を，**木言語** (*tree language*) と呼ぶ．

定義 58 (木言語). ランク付きアルファベット Σ について，$L \subseteq \hat{T}_\Sigma$ を木言語と呼ぶ． □

木言語を定める文法として，**正規木文法** (*regular tree grammar*) がある．

定義 59 (正規木文法). 正規木文法は，以下の要素の組 $G = (N, \Sigma, S, R)$ である．

N　　非終端記号のランク付きアルファベットで $N = N^{(0)}$．
Σ　　終端記号のランク付きアルファベット．
$S \in N$　開始記号．
R　　書き換え規則の集合で，以下を満たす．

$$R \in \mathcal{P}_{fin}(\{A \to \eta \mid A \in N, \eta \in \hat{T}_{\Sigma \cup N}\})$$

□

正規木文法は木言語を意味論として持つ．その意味論には，開始記号から書き換え規則により非終端記号を書き換えていき，最終的に得られる終端記号のみから構成される木が含まれる．正規木文法で定義できる木言語を，**正規木言語** (*regular tree language, RTL*) と呼ぶ．

定義 60 (正規木文法の意味論). 正規木文法 $G = (N, \Sigma, S, R)$ について，簡約システム $(U, \Rightarrow_G)$ を，$U \stackrel{\text{def}}{=} \hat{T}_{\Sigma \cup N}$，$\Rightarrow_G$ は以下のように構造的帰納法で定義する．

IB1　適用不能．

IB2　以下のように場合分けを行う．

- $\sigma \in \Sigma^{(0)}$ について，σ は既約．
- $A \in N$，$A \to \eta \in R$ について，$A \Rightarrow_G \eta$．

IS　$n \geq 1$，$\sigma \in \Sigma^{(n)}$，$\eta_1, \ldots, \eta_n \in U$，$i \in [n]$ について，$\eta_1, \ldots, \eta_{i-1}$ が $\Rightarrow_G$ で既約で，$\eta_i \Rightarrow_G \eta_i'$ の時，

$$\sigma(\eta_1, \ldots, \eta_i, \ldots, \eta_n) \Rightarrow_G \sigma(\eta_1, \ldots, \eta_i', \ldots, \eta_n).$$

この時，$[\![G]\!] = \{t \in \hat{T}_\Sigma \mid S \Rightarrow_G^* t\}$ と表記する．なお，木言語クラスを $\mathrm{RTL} = \{[\![G]\!] \mid G$ は正規木文法 $\}$ と表記し，$L \in \mathrm{RTL}$ を正規木言語と呼ぶ． □

有限の木言語は，RTL である．

定理 61. ランク付きアルファベット Σ，$L \in \mathcal{P}_{fin}(\hat{T}_\Sigma)$ について，$L \in \mathrm{RTL}$． □

証明. 正規木文法 $G = (N, \Sigma, S, R)$ を以下のように定義する．

$$N \stackrel{\text{def}}{=} \{A\}$$
$$S \stackrel{\text{def}}{=} A$$
$$R \stackrel{\text{def}}{=} \{A \to t \mid t \in L\}$$

この時，$[\![G]\!] = L$ を示す．任意の $t \in [\![G]\!]$ において，$A \Rightarrow_G t$ より，$t \in L$．よって，$[\![G]\!] \subseteq L$ である．また，逆も同様．よって，題意は示された． ■

RTL に対応する計算モデルとして，**有限木オートマトン** (*finite tree automata, FTA*) がある．

定義 62 (有限木オートマトン (FTA)). 有限木オートマトンは，以下の要素の組 $M = (Q, \Sigma, q_0, R)$ である．

Q　状態記号のランク付きアルファベットで，$Q = Q^{(1)}$．

Σ　入力記号のランク付きアルファベット．

$q_0 \in Q$　開始記号．

R　　書き換え規則の集合で，以下を満たす．

$$R \in \mathcal{P}_{fin}(\{q(\sigma(\mathrm{x}_1, \dots, \mathrm{x}_n)) \to \sigma(q_1(\mathrm{x}_1), \dots, q_n(\mathrm{x}_n)) \mid \sigma^{(n)} \in \Sigma, q, q_1, \dots, q_n \in Q\})$$

□

FTA は木言語を意味論に持つ．その意味論には，木の根から書き換え規則に則ってノードに状態を割り当てていき，最終的に全てのノードに状態が割り当てられる木が含まれる．

定義 63 (FTA の意味論). FTA $M = (Q, \Sigma, q_0, R)$, 集合 X について, 簡約システム $(U, \Rightarrow_{M,X})$ を，以下のように定義する．

$$U \stackrel{\text{def}}{=} \hat{T}_{\Sigma \cup Q}(X)$$

$$\eta_1 \Rightarrow_{M,X} \eta_2 \text{ iff } \left(\begin{array}{l} \exists q(\sigma(\mathrm{x}_1, \dots, \mathrm{x}_n)) \to \eta \in R, \\ \quad t_1, \dots, t_n \in \hat{T}_\Sigma(X), \\ \quad \pi \in \mathrm{paths}(\eta_1), \mathrm{subtree}_{\eta_1}(\pi) = q(\sigma(t_1, \dots, t_n)) \ . \\ \eta_2 = \eta_1[\pi \leftarrow \eta[\mathrm{x}_i \leftarrow t_i]_{i \in [n]}] \end{array} \right)$$

この時，$\Rightarrow_{M,\emptyset}$ を $\Rightarrow_M$ と表記する．また，M の意味論 $[\![M]\!]$ を $\{t \in \hat{T}_\Sigma \mid q_0(t) \Rightarrow_M^* t\}$ で定義する．さらに，木言語のクラス $\{[\![M]\!] \mid M \text{ は FTA}\}$ を FTA と表記する．□

FTA による木言語のクラスと RTL のクラスは等しいことが知られている [Eng15].

定理 64. RTL = FTA □

証明. [Eng15] の定理 3.17，3.25 より． ■

また，木言語が RTL に属するかを判別する方法のひとつとして，**反復補題** (*pumping lemma*) が知られている [Eng15].

補題 65 (RTL の反復補題). ランク付きアルファベット Σ，$L \in \mathcal{P}(\hat{T}_\Sigma) \cap \mathrm{RTL}$ について，ある $p_L \in \mathbb{N}$ が存在し，任意の $\mathrm{height}(t) \geq p_L$ を満たす $t \in L$ について，以下を満たす $u_1, u_2 \in \hat{T}_\Sigma(\emptyset, \{*\})$，$u_3 \in \hat{T}_\Sigma$ が存在する．

- $t = u_1(u_2(u_3))$.
- $\mathrm{height}(u_2(u_3)) \leq p_L$.
- $\mathrm{height}(u_2) \geq 1$.
- 任意の $n \in \mathbb{N}$ について，$u_1(u_2^n(u_3)) \in L$.

□

この定理は，以下の一般化された定理から導かれる．

補題 66 (RTL の一般化反復補題). ランク付きアルファベット Σ，$L \in \mathcal{P}(\hat{T}_\Sigma) \cap \mathrm{RTL}$ について，ある $p_L \in \mathbb{N}$ が存在し，任意の $t_1(t_2) \in L$，$\mathrm{height}(t_1) \geq p_L$ を満たす $t_1 \in \hat{T}_\Sigma(\emptyset, \{*\})$，$t_2 \in \hat{T}_\Sigma$ について，以下を満たす $u_1, u_2, u_3 \in \hat{T}_\Sigma(\emptyset, \{*\})$ が存在する．

- $t_1 = u_1(u_2(u_3))$.
- $\mathrm{height}(u_2(u_3)) \leq p_L$.
- $\mathrm{height}(u_2) \geq 1$.
- 任意の $n \in \mathbb{N}$ について，$u_1(u_2^n(u_3(t_2))) \in L$.

□

証明. $[\![M]\!] = L$ となる FTA $M = (Q, \Sigma, q_0, R)$ について，$p_L = |Q| + 1$ とおく．この時，$t_1 = t_1'(\sigma_1(\ldots, \sigma_2(\ldots, \cdots \sigma_{p_L}(\ldots, *, \ldots) \cdots, \ldots), \ldots))$ を満たす，$\sigma_1, \ldots, \sigma_{p_L}$ が存在し，また，$q_1, \ldots, q_{p_L+1}$ が存在して，

- 任意の $i \in [p_L]$ で $q_i(\sigma_i(\ldots)) \to \sigma_i(\ldots, q_{i+1}(\ldots), \ldots) \in R$
- $q_0(t_1'(*)) \Rightarrow^*_{M,\{*\}} t_1'(q_1(*))$
- $q_{p_L+1}(t_2) \Rightarrow^*_M t_2$

を満たす．この時，$q_{i_1} = q_{i_2}$ を満たす $1 \leq i_1 < i_2 \leq p_L$ が存在し，
$u_1 = t_1'(\sigma_1(\ldots, \cdots \sigma_{i_1-1}(\ldots, *, \ldots) \cdots, \ldots))$，$u_2 = \sigma_{i_1}(\ldots, \cdots \sigma_{i_2-1}(\ldots, *, \ldots) \cdots, \ldots)$，$u_3 = \sigma_{i_2}(\ldots, \cdots \sigma_{p_L}(\ldots, *, \ldots) \cdots, \ldots)$ とおくと，これは条件を満たす．よって，題意は示された． ■

補題 65 は，t_1 の最長のパスを持つ葉を t_2 とおき，補題 66 を適用して作られた u_1, u_2, u_3 において，$u_1, u_2, u_3(t_2)$ を新たな u_1, u_2, u_3 とおくことで得られる．補題 65 より，同じ回数異なる 2 つの記号が連続した木からなる木言語は，RTL に属さないことが分かる．

命題 67. $\Sigma = \{a^{(1)}, b^{(1)}, \$^{(0)}\}$，$L = \{a^n(b^n(\$)) \mid n \in \mathbb{N}\} \subseteq \hat{T}_\Sigma$ について，$L \notin \mathrm{RTL}$ □

証明. $L \in \mathrm{RTL}$ と仮定する．この時，補題 65 の p_L，$t = a^{p_L}(b^{p_L}(\$)) \in L$ について，以下を満たす u_1, u_2, u_3 が存在する．

- $t = u_1(u_2(u_3))$.
- $\mathrm{height}(u_2(u_3)) \leq p_L$.
- $\mathrm{height}(u_2) \geq 1$.
- 任意の $n \in \mathbb{N}$ について，$u_1(u_2^n(u_3)) \in L$.

特に $u_1(u_3) \in L$ である．この時，$m = \mathrm{height}(u_2)$ とすると $u_2 = b^m(*)$ と書け，さらに $m \geq 1$ より，$u_1(u_3) = a^{p_L}(b^{p_L - m}(\$))$ かつ $p_L \neq p_L - m$ だが，これは $u_1(u_3) \in L$ に矛盾する．よって，$L \notin \mathrm{RTL}$ である． ■

木言語を定めるより強力な文法として，**文脈自由木文法** (*context-free tree grammar*) がある．文脈自由木文法が，正規木文法と異なる点は，非終端記号が木を受け取れる点にある．

定義 68 (文脈自由木文法). 文脈自由木文法は，以下の要素の組 $G = (N, \Sigma, S, R)$ である．

N　　非終端記号のランク付きアルファベット
Σ　　終端記号のランク付きアルファベット
$S \in N^{(0)}$　開始記号．
R　　書き換え規則の集合で，以下を満たす．

$$R \in \mathcal{P}_{fin}(\{A(\mathrm{x}_1, \ldots, \mathrm{x}_n) \to \eta \mid A^{(n)} \in N, \eta \in \hat{T}_{\Sigma \cup N}(\mathrm{X}_n)\})$$

□

文脈自由木文法は木言語を意味論として持つ．正規木文法と同じく，その意味論には，開始記号から書き換え規則により非終端記号を書き換えていき，最終的に得られる終端記号のみから構成される木が含まれる．文脈自由木文法で定義できる木言語を，**文脈自由木言語** (*context-free tree language, CFTL*) と呼ぶ．

定義 69 (文脈自由木文法の意味論). 文脈自由木文法 $G = (N, \Sigma, S, R)$ について，簡約システム $(U, \Rightarrow_G)$ を，$U \stackrel{\mathrm{def}}{=} \hat{T}_{\Sigma \cup N}$，$\Rightarrow_G$ は以下のように構造的帰納法で定義する．

IB1　適用不能．
IB2　以下のように場合分けを行う．
- $\sigma \in \Sigma^{(0)}$ について，σ は既約．
- $A \in N^{(0)}$，$A \to \eta \in R$ について，$A \Rightarrow_G \eta$．

IS　以下のように場合分けを行う．
- 任意の $n \geq 1$，$\sigma \in \Sigma^{(n)}$，$\eta_1, \ldots, \eta_n \in U$，$i \in [n]$ について，$\eta_1, \ldots, \eta_{i-1}$ が $\Rightarrow_G$ で既約で，$\eta_i \Rightarrow_G \eta_i'$ の時，

$$\sigma(\eta_1, \ldots, \eta_i, \ldots, \eta_n) \Rightarrow_G \sigma(\eta_1, \ldots, \eta_i', \ldots, \eta_n).$$

- 任意の $n \geq 1$，$A \in N^{(n)}$，$\eta_1, \ldots, \eta_n \in U$，$A(\mathrm{x}_1, \ldots, \mathrm{x}_n) \to \eta \in R$ について，

$$A(\eta_1, \ldots, \eta_n) \Rightarrow_G \eta[\mathrm{x}_i \leftarrow \eta_i]_{i \in [n]}.$$

この時，G の意味論 $[\![G]\!]$ を $\{t \in \hat{T}_\Sigma \mid S \Rightarrow_G^* t\}$ で定義する．なお，木言語クラス $\{[\![G]\!] \mid G$ は文脈自由木文法 $\}$ を CFTL と表記し，$L \in$ CFTL を文脈自由木言語と呼ぶ．□

命題 67 の木言語は，CFTL に属する．

命題 70. $\Sigma = \{a^{(1)}, b^{(1)}, \$^{(0)}\}$，$L = \{a^n(b^n(\$)) \mid n \in \mathbb{N}\} \subseteq \hat{T}_\Sigma$ について，$L \in$ CFTL　□

証明. $G = (N, \Sigma, S, R)$ を以下のように定義する.

- $N = \{S^{(0)}, A^{(1)}\}$.
- $R = \{S \to A(\$), A(\mathrm{x}_1) \to a(A(b(\mathrm{x}_1))), A(\mathrm{x}_1) \to \mathrm{x}_1\}$.

この時, $[\![G]\!] = L$ は, $t \in L$ の導出木と呼ばれる木に関する構造的帰納法より示せる [HMU00].
■

また, 文脈自由木文法において非終端記号をランク 0 の記号のみに制限すると, それは正規木文法になることから, RTL は CFTL より小さい. さらに, 命題 70 から RTL は CFTL より真に小さい.

定理 71. RTL $\subsetneq$ CFTL □

証明. $L \in$ RTL について, $[\![G]\!] = L$ を満たす正規木文法 $G = (N, \Sigma, S, R)$ が存在する. この時, G は意味論同値な文脈自由木文法でもある. よって, RTL $\subseteq$ CFTL である. 命題 67, 命題 70 より, $L \notin$ RTL かつ $L \in$ CFTL を満たす木言語 L が存在するため, CFTL $\nsubseteq$ RTL である. よって, 題意は示された. ■

CFTL に対応する計算モデルとして, **プッシュダウン木オートマトン** (*pushdown tree automata, PDTA*) [Gue83] がある.

定義 72 (プッシュダウン木オートマトン (PDTA)). プッシュダウン木オートマトンは, 以下の要素の組 $M = (Q, \Sigma, \Gamma, q_0, u_0, R)$ である.

Q 状態記号のランク付きアルファベットで, $Q = Q^{(2)}$.
Σ 入力記号のランク付きアルファベット.
Γ スタック記号のランク付きアルファベット.
$q_0 \in Q$ 開始記号.
$u_0 \in \hat{T}_\Gamma$ 初期のスタックの木.
R 書き換え規則の集合で, 以下を満たす.

$$R \in \mathcal{P}_{fin}\left(\begin{array}{c}\{q(\sigma(\mathrm{x}_1,\dots,\mathrm{x}_n),\gamma(\mathrm{z}_1,\dots,\mathrm{z}_m)) \to \sigma(q_1(\mathrm{x}_1,u_1),\dots,q_n(\mathrm{x}_n,u_n)) \\ \mid \sigma^{(n)} \in \Sigma, q, q_1,\dots,q_n \in Q, \gamma^{(m)} \in \Gamma, u_1,\dots,u_n \in \hat{T}_\Gamma(\mathrm{Z}_m)\} \\ \cup \\ \{q(\mathrm{x},\gamma(\mathrm{z}_1,\dots,\mathrm{z}_m)) \to q'(\mathrm{x},u) \mid q, q' \in Q, \gamma^{(m)} \in \Gamma, u \in \hat{T}_\Gamma(\mathrm{Z}_m)\}\end{array}\right)$$

□

PDTA は木言語を意味論に持つ. その意味論には, 木の根から書き換え規則に則って, スタックを書き換えながらノードに状態を割り当てていき, 最終的に全てのノードに状態が割り当てられる木が含まれる.

定義 73 (PDTA の意味論). PDTA $M = (Q, \Sigma, \Gamma, q_0, u_0, R)$ について，簡約システム $(U, \Rightarrow_M)$ を，以下のように定義する．

$$U \stackrel{\text{def}}{=} \hat{T}_{\Sigma \cup Q \cup \Gamma}$$

$$\eta_1 \Rightarrow_M \eta_2 \text{ iff } \left(\begin{array}{l} \exists q(t, \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \to \eta \in R, t \in \hat{T}_\Sigma(X) \\ \quad t' \in \prod_{x \in X} \hat{T}_\Sigma, \\ \quad u_1, \ldots, u_m \in \hat{T}_\Gamma, \\ \quad \pi \in \mathrm{paths}(\eta_1), \mathrm{subtree}_{\eta_1}(\pi) = q(t[x \leftarrow t'_x]_{x \in X}, \gamma(u_1, \ldots, u_m)) \, . \\ \eta_2 = \eta_1[\pi \leftarrow \eta[x \leftarrow t'_x]_{x \in X}[\mathrm{z}_i \leftarrow u_i]_{i \in [m]}] \end{array} \right)$$

この時，M の意味論 $[\![M]\!]$ を $\{t \in \hat{T}_\Sigma \mid q_0(t, u_0) \Rightarrow_M^* t\}$ で定義する．また，木言語のクラス $\{[\![M]\!] \mid M \text{ は PDTA}\}$ を PDTA と表記する． □

PDTA による木言語のクラスと CFTL のクラスは等しいことが知られている [Gue83].

定理 74. CFTL = PDTA □

証明. [Gue83] の定理 1 より． ■

また，単項の木による木言語が CFTL に属するかを判別する方法のひとつとして，反復補題が知られている [Eng15]．これは，単項の木による CFTL に対応する RTL の反復補題から導かれる．

補題 75 (CFTL の反復補題). 単項ランク付きアルファベット Σ，$L \in \mathcal{P}(\hat{T}_\Sigma) \cap \text{CFTL}$ について，ある $p_L \in \mathbb{N}$ が存在し，任意の $\mathrm{height}(t) \geq p_L$ を満たす $t \in L$ について，以下を満たす $u_1, u_2, u_3, u_4 \in \hat{T}_\Sigma(\emptyset, \{*\})$，$u_5 \in \hat{T}_\Sigma$ が存在する．

- $t = u_1(u_2(u_3(u_4(u_5))))$.
- $\mathrm{height}(u_2(u_3(u_4))) \leq p_L$.
- $\mathrm{height}(u_2(u_4)) \geq 1$.
- 任意の $n \in \mathbb{N}$ について，$u_1(u_2^n(u_3(u_4^n(u_5)))) \in L$.

□

証明. [Eng15] の系 3.72 より． ■

補題 75 より，同じ回数異なる 3 つの記号が連続した木からなる木言語は，CFTL に属さないことが分かる．

命題 76. $\Sigma = \{a^{(1)}, b^{(1)}, c^{(1)}, \$^{(0)}\}$, $L = \{a^n(b^n(c^n(\$))) \mid n \in \mathbb{N}\} \subseteq \hat{T}_\Sigma$ について，$L \notin \text{CFTL}$. □

証明. $L \in \text{CFTL}$ と仮定する．この時，補題 75 の p_L について，$t = a^{p_L}(b^{p_L}(c^{p_L}(\$))) \in L$ を考えると，以下を満たす u_1, u_2, u_3, u_4, u_5 が存在する．

- $t = u_1(u_2(u_3(u_4(u_5))))$.
- $\mathrm{height}(u_2(u_3(u_4))) \leq p_L$.
- $\mathrm{height}(u_2(u_4)) \geq 1$.
- 任意の $n \in \mathbb{N}$ について，$u_1(u_2^n(u_3(u_4^n(u_5)))) \in L$.

特に，$u_1(u_3(u_5)) \in L$ である．ある $m_1, m_2 \in \mathbb{N}$ について，$u_2(u_3(u_4)) = a^{m_1}(b^{m_2}(*))$ または $u_2(u_3(u_4)) = b^{m_1}(c^{m_2}(*))$ であり，$u_1(u_3(u_5)) = a^{k_1}(b^{k_2}(c^{k_3}(*)))$ とすると，$k_1 = p_L$ かつ $k_2 < p_L$ または $k_3 < p_L$，$k_3 = p_L$ かつ $k_1 < p_L$ または $k_2 < p_L$ のいずれかが成り立つ．よって，$u_1(u_3(u_5)) \notin L$ だがこれは矛盾．よって，題意は示された． ■

これは，CFTL が積で閉じていない [HMU00] ことを意味する．

命題 77. $\Sigma = \{a^{(1)}, b^{(1)}, c^{(1)}, \$^{(0)}\}$, $L_1 = \{a^n(b^n(c^m(\$))) \mid n, m \in \mathbb{N}\}$, $L_2 = \{a^n(b^m(c^m(\$))) \mid n, m \in \mathbb{N}\}$ について，以下が成り立つ．

- $L_1, L_2 \in \mathrm{CFTL}$
- $L_1 \cap L_2 \notin \mathrm{CFTL}$

□

証明. 文脈自由木文法 $G_1 = (N_1, \Sigma, S_1, R_1)$ を以下のように定義する．

$$N_1 \stackrel{\mathrm{def}}{=} \{S_1^{(0)}, A_1^{(1)}, C_1^{(0)}\}$$
$$R_1 \stackrel{\mathrm{def}}{=} \{S_1 \to A(C), A(\mathrm{x}_1) \to a(A(b(\mathrm{x}_1))), A(\mathrm{x}_1) \to \mathrm{x}_1, C \to c(C), C \to \$\}$$

この時，$[\![G_1]\!] = L_1$ は，命題 70 と同様に示せる．同様に $[\![G_2]\!] = L_2$ となる文脈自由木文法 G_2 も定義できる．

また，

$$\begin{aligned} t \in L_1 \cap L_2 &\iff \exists n_1, n_2, n_3 \in \mathbb{N} \,.\, t = a^{n_1}(b^{n_2}(c^{n_3}(\$))) \wedge n_1 = n_2 = n_3 \\ &\iff \exists n \in \mathbb{N} \,.\, t = a^n(b^n(c^n(\$))) \end{aligned}$$

より，$L_1 \cap L_2 = \{a^n(b^n(c^n(\$))) \mid n \in \mathbb{N}\}$ であり，命題 76 より，$L_1 \cap L_2 \notin \mathrm{CFTL}$. ■

また，ランク付き木からランク付き木への関数を，**木変換** (*tree transformation*) と呼ぶ．

定義 78 (木変換). ランク付きアルファベット Σ, Δ について，$f : \hat{T}_\Sigma \to \hat{T}_\Delta$ を木変換と呼ぶ．

□

第 3 章　木変換器

本章では，**トップダウン木変換器** [Rou68]，**属性付き木変換器** [Fül81]，**マクロ木変換器** [EV85] の 3 つの木変換器の定義とその拡張，知られている定理について，[FV98] に倣って導入する．

トップダウン木変換器は入力木を根から葉に向けて辿っていく素朴な木変換器であり，定義 79 で示している．属性付き木変換器及びマクロ木変換器はトップダウン木変換器の入力木の辿り方を拡張した木変換器になる．属性付き木変換器は属性文法 [Knu68] を元にした木変換器であり，定義 98 で示している．トップダウン木変換器が親から子へのノードの辿り方しかできないのに対し，属性付き木変換器では子から親や，子から別の子へのノードへ辿ることも可能になる．その代わり，簡約が下降型にならず同じノードを何度も辿るような巡回性を持つようになり，停止性を持たなくなってしまう．ただし，属性付き木変換器の簡約が巡回を含むかの判定は属性文法の巡回性判定をそのまま用いることで決定可能である．定理 110 では停止性を持つことと巡回する簡約を持つことが同値であることを示している．また，定理 111 では，その方法と正当性の概略を [FV98] に倣って示している．これは (停止性を持つ) 属性付き木変換器について，意味論同値なマクロ木変換器に変換できることやその樹高性を示す際に，重要な性質となる．マクロ木変換器は，トップダウン木変換器と同じく構造的再帰下降型の木変換器であるが，再帰時にコンテキストパラメータと呼ばれるパラメータを持つことを許されている点が異なる．その定義は，定義 123 で示している．コンテキストパラメータは言及できない代わりに，再帰時に値を指定でき，その値にさらなる再帰を含めることができる．そのため，木を根から葉に辿る途中で擬似的に過去に辿ったノードへ戻ることが可能になる．この動作を使うことで，入力木を子から親や別の子へ辿る動作を模倣できる，つまり属性付き木変換器の動作を模倣できることが知られている [Eng80]．補題 153 では，その具体的な模倣方法を示している．また，トップダウン木変換器と属性付き木変換器，マクロ木変換器はそれぞれ別の樹高性を持ち，そこから互いのクラスが異なることが分かっている [Fül81][Eng80][FV98]．定理 94，定理 121，定理 134 では，それぞれの樹高性を示している．また，定理 152，定理 155 では以上の結果から，トップダウン木変換器と属性付き木変換器，属性付き木変換器とマクロ木変換器とのクラスの間に真の部分関係が成り立つことを，[FV98] に倣って示している．これはそれぞれの木変換器の表現力が，トップダウン木変換器，属性付き木変換器，マクロ木変換器の順に真に大きいことを示す，既存の木変換器理論における重要な結果である．

3.1　トップダウン木変換器

トップダウン木変換器 (TDTT) は，入力の木に対して，根からノードのラベルを読み取って出力木を構築する操作を，部分木に対して原始的再帰的に行っていく計算の構文的なモデルに

なっている．

定義 79 (トップダウン木変換器 (TDTT)). トップダウン木変換器は，以下の要素の組 $M = (Q, \Sigma, \Delta, e, R)$ である．

Q　状態記号のランク付きアルファベットで，$Q = Q^{(1)}$ ．
Σ　入力記号のランク付きアルファベット．
Δ　出力記号のランク付きアルファベット．
e　初期式で，$e \in \mathrm{RHS}_M(\mathrm{X}_1)$．
R　書き換え規則の集合で，以下を満たす．

$$R \in \mathcal{P}_{fin}(\{q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n)) \to \eta \mid q \in Q, \sigma^{(n)} \in \Sigma, \eta \in \mathrm{RHS}_M(\mathrm{X}_n)\})$$

ただし，$\mathrm{RHS}_M(X) \subseteq \hat{T}_{\Delta \cup Q}(X)$ は，以下のように帰納的に定義される集合 U である．

IB1　$q \in Q$，$x \in X$ について，$q(x) \in U$
IB2　$\delta \in \Delta^{(0)}$ について，$\delta \in U$
IS　$n \geq 1$，$\delta^{(n)} \in \Delta$，$\eta_1, \ldots, \eta_n \in U$ について，$\delta(\eta_1, \ldots, \eta_n) \in U$

□

TDTT は，現在の状態と入力木の部分木の根のラベルを見て適用する規則を選び，その規則の右辺の通りに出力を構築しながら新たに適用する規則を選んでいく．

例 80. TDTT $M = (Q, \Sigma, \Delta, e, R)$ を，以下のように定義する．

$$\begin{aligned}
Q &\stackrel{\mathrm{def}}{=} \{q\} \\
\Sigma &\stackrel{\mathrm{def}}{=} \{\mathrm{A}^{(2)}, \mathrm{B}^{(0)}, \mathrm{C}^{(0)}\} \\
\Delta &\stackrel{\mathrm{def}}{=} \Sigma \\
e &\stackrel{\mathrm{def}}{=} q(\mathrm{x}_1) \\
R &\stackrel{\mathrm{def}}{=} \{q(\mathrm{A}(\mathrm{x}_1, \mathrm{x}_2)) \to \mathrm{A}(q(\mathrm{x}_2), q(\mathrm{x}_1)), \\
&\qquad q(\mathrm{B}) \to \mathrm{B}, \\
&\qquad q(\mathrm{C}) \to \mathrm{C}\}
\end{aligned}$$

この M は，与えられた木を左右反転する TDTT で，以下のような変換を定義する．

$$[\![M]\!] \left(\begin{array}{ccccc} & & \mathrm{A} & & \\ & \swarrow & & \searrow & \\ \mathrm{C} & & & & \mathrm{A} \\ & & & \swarrow & \searrow \\ & & \mathrm{B} & & \mathrm{C} \end{array} \right) = \begin{array}{ccccc} & & \mathrm{A} & & \\ & \swarrow & & \searrow & \\ & & \mathrm{A} & & \mathrm{C} \\ & \swarrow & & \searrow & \\ \mathrm{C} & & & & \mathrm{B} \end{array}$$

なお，M の意味論 $[\![M]\!]$ の形式的な定義は，後ほど導入する．　□

一般に TDTT は，2 つの木の間の関係を定義し，その関係は関数でない場合がある．しかし，本論文では，この定義される関係が関数の場合のみを扱う．関数を定義するような TDTT は構文的制約のみで定められることが知られている．その制約を満たす TDTT を，**全域的決定的** (*total deterministic*) TDTT と呼ぶ．

定義 81 (全域的決定的 TDTT). TDTT $M = (Q, \Sigma, \Delta, e, R)$ が

$$\forall q \in Q, \sigma^{(n)} \in \Sigma \,.\, \exists! q(\sigma(\vec{\mathrm{x}})) \to \eta \in R$$

を満たすとき，全域的決定的であるという． □

全域的決定的 TDTT とは，入力木のアルファベットと状態を規則の左辺が全て網羅しており，またただひとつだけの規則が対応するという制約である．例 80 は全域的決定的の例になっている．以降，特に断りのない限り TDTT は全域的決定的であるものとする．TDTT について，意味論を決定づける簡約システムを導入する．

定義 82 (TDTT の簡約システム). TDTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X について，簡約システム $(\mathrm{SF}_M(X), \Rightarrow_{M,X})$ を，以下のように定義する．

$$\mathrm{SF}_M(X) \stackrel{\text{def}}{=} \mathrm{RHS}_M(\hat{T}_\Sigma(X))$$

$$\varphi_1 \Rightarrow_{M,X} \varphi_2 \text{ iff } \left(\begin{array}{l} \exists q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n)) \to \eta \in R, \\ \quad t_1, \ldots, t_n \in \hat{T}_\Sigma, \\ \quad \pi \in \mathrm{paths}(\eta_1), \mathrm{subtree}_{\varphi_1}(\pi) = q(\sigma(t_1, \ldots, t_n)) \,. \\ \varphi_2 = \varphi_1[\pi \leftarrow \eta[\mathrm{x}_i \leftarrow t_i]_{i \in [n]}] \end{array} \right)$$

この時，$\Rightarrow_{M,\emptyset}$ を $\Rightarrow_M$ と表記する． □

この簡約システムは局所合流性を持つ．

補題 83. TDTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X について，$\Rightarrow_{M,X}$ は局所合流性を持つ． □

証明. [FV98] の補題 3.9 より． ■

また，この簡約システムは停止性を持つ．

補題 84. TDTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X について，$\Rightarrow_{M,X}$ は停止性を持つ． □

証明. [FV98] の補題 3.13 より． ■

以上から，TDTT の簡約システムは合流性を持つ．

系 85. TDTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X について，$\Rightarrow_{M,X}$ は合流性を持つ． □

証明. 補題 84，補題 83，補題 56 より． ■

TDTT の簡約システムは，合流性と停止性を持つため，補題 57 より一意な正規形を持つ．この正規形は，$\Rightarrow_M$ においては必ず出力アルファベットのみからなる木になる．

補題 86. TDTT $M = (Q, \Sigma, \Delta, e, R)$ について，以下が成り立つ．

$$\forall \eta \in \mathrm{SF}_M(\emptyset) \,.\, \mathrm{nf}(\Rightarrow_M, \eta) \in \hat{T}_\Delta$$

□

証明. [FV98] の定理 3.16 より． ■

よって，TDTT の簡約システムにおける変数を持たない項の正規形は，必ず一意でかつ出力木になる．この時 TDTT に対して，入力木を受け取り，初期式にその木を割り当てたものから TDTT の簡約システムで得られる木を出力とする，木変換が考えられる．この木変換を，意味論として導入する．

定義 87 (TDTT の意味論). TDTT $M = (Q, \Sigma, \Delta, e, R)$ について，$[\![M]\!] : \hat{T}_\Sigma \to \hat{T}_\Delta$ を以下のように定義する．

$$[\![M]\!] \stackrel{\mathrm{def}}{=} t \mapsto \mathrm{nf}(\Rightarrow_M, e[\mathrm{x}_1 \leftarrow t])$$

また，木変換のクラス $\{[\![M]\!] \mid M$ は (全域的決定的) TDTT$\}$ を TDTT と表記する． □

なお，TDTT が全域的決定的な場合，TDTT の簡約システムを決定的にすることができる．

命題 88 (TDTT の決定的な簡約システム). TDTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X について，簡約システム $(\mathrm{SF}_M(X), \Rightarrow^D_{M,X})$ を，$U \stackrel{\mathrm{def}}{=} \mathrm{SF}_M(X)$ とおいて，以下のように構造的帰納法で定義する．

IB1　以下のように場合分けを行う．

- $q \in Q$，$x \in X$ について，$q(x)$ は既約．
- $q \in Q$，$\sigma \in \Sigma$，$\sigma(\vec{t}) \in \hat{T}_\Sigma(X), q(\sigma(\vec{\mathrm{x}})) \to \eta \in R$ について，

$$q(\sigma(\vec{t})) \Rightarrow^D_{M,X} \eta[\vec{\mathrm{x}} \leftarrow \vec{t}].$$

IB2　$\delta^{(0)} \in \Delta$ について，δ は既約．

IS　$n \geq 1$，$\delta^{(n)} \in \Delta$，$\eta_1, \ldots, \eta_n \in U$，$i \in [n]$ について，$\eta_1, \ldots, \eta_{i-1}$ が $\Rightarrow^D_{M,X}$ で既約で，$\eta_i \Rightarrow^D_{M,X} \eta'_i$ の時，

$$\delta(\eta_1, \ldots, \eta_i, \ldots, \eta_n) \Rightarrow^D_{M,X} \delta(\eta_1, \ldots, \eta'_i, \ldots, \eta_n).$$

この時，任意の $\eta \in U$ に対して，$\mathrm{nf}(\Rightarrow_{M,X}, \eta) = \mathrm{nf}(\Rightarrow^D_{M,X}, \eta)$. □

証明. $\mathrm{nf}(\Rightarrow^D_{M,X}, \eta)$ が $\Rightarrow_{M,X}$ での正規形であることを示せば，正規形の一意性から題意は導かれる．これは，容易な η に関する構造的帰納法より示せる． ■

例 89. 例 80 の TDTT $M = (Q, \Sigma, \Delta, e, R)$ について，以下のような簡約例が考えられる．

$$\begin{aligned} e[\mathrm{x}_1 \leftarrow \mathrm{A}(\mathrm{C}, \mathrm{A}(\mathrm{B}, \mathrm{C}))] &= q(\mathrm{A}(\mathrm{C}, \mathrm{A}(\mathrm{B}, \mathrm{C}))) \\ &\Rightarrow_M \mathrm{A}(q(\mathrm{A}(\mathrm{B}, \mathrm{C})), q(\mathrm{C})) \\ &\Rightarrow_M \mathrm{A}(\mathrm{A}(q(\mathrm{C}), q(\mathrm{B}))), q(\mathrm{C})) \\ &\Rightarrow_M \mathrm{A}(\mathrm{A}(\mathrm{C}, q(\mathrm{B}))), q(\mathrm{C})) \\ &\Rightarrow_M \mathrm{A}(\mathrm{A}(\mathrm{C}, \mathrm{B})), q(\mathrm{C})) \\ &\Rightarrow_M \mathrm{A}(\mathrm{A}(\mathrm{C}, \mathrm{B})), \mathrm{C}) \end{aligned}$$

ここから，意味論による変換は以下のようになる．

$$[\![M]\!] \left(\begin{array}{ccccc} & & \mathrm{A} & & \\ & \swarrow & & \searrow & \\ \mathrm{C} & & & & \mathrm{A} \\ & & & \swarrow & \searrow \\ & & \mathrm{B} & & \mathrm{C} \end{array} \right) = \begin{array}{ccccc} & & & & \mathrm{A} \\ & & & \swarrow & \searrow \\ & & \mathrm{A} & & \mathrm{C} \\ & \swarrow & & \searrow & \\ \mathrm{C} & & & & \mathrm{B} \end{array}$$

□

TDTT で表現できる木変換のひとつに，恒等変換がある．

定義 90 (恒等変換). ランク付きアルファベット Σ について，TDTT $\mathrm{Id}_\Sigma = (Q, \Sigma, \Sigma, e, R)$ を以下のように定義する．

$$\begin{aligned} Q &\stackrel{\mathrm{def}}{=} \{q\} \\ e &\stackrel{\mathrm{def}}{=} q(\mathrm{x}_1) \\ R &\stackrel{\mathrm{def}}{=} \{q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n)) \to \sigma(q(\mathrm{x}_1), \ldots, q(\mathrm{x}_n)) \mid \sigma^{(n)} \in \Sigma\} \end{aligned}$$

□

命題 91. $[\![\mathrm{Id}_\Sigma]\!] : \hat{T}_\Sigma \to \hat{T}_\Sigma = t \mapsto t$ □

証明. $M \stackrel{\mathrm{def}}{=} \mathrm{Id}_\Sigma$ とおく．

$$\forall t \in \hat{T}_\Sigma \,.\, \mathrm{nf}(\Rightarrow_M, q(t)) = t$$

を，構造的帰納法で示す．

IB1　適用不能．

IB2　$\sigma \in \Sigma^{(0)}$ について，$q(\sigma) \Rightarrow_M \sigma$ より正しい．

IS　$n \geq 1$，$\sigma^{(n)} \in \Sigma$，$t_1, \ldots, t_n \in \hat{T}_\Sigma$ について，

$$\begin{aligned} q(\sigma(t_1, \ldots, t_n)) &\Rightarrow_M \sigma(q(t_1), \ldots, q(t_n)) \\ &\Rightarrow^*_M \sigma(t_1, \ldots, t_n) \qquad (\because \text{i.h.}) \end{aligned}$$

より正しい．

よって，$[\![M]\!](t) = \mathrm{nf}(\Rightarrow, q(t)) = t$. ■

他の木変換として，入力木を複製する木変換がある．この木変換は，入力木を 2 つに複製し，それぞれをランク 2 の記号の部分木として出力する．

例 92. ランク付きアルファベット Σ について，TDTT $\mathrm{InvRE}_\Sigma = (Q, \Sigma, \Delta, e, R)$ を以下のように定義する．

$$
\begin{aligned}
Q &\stackrel{\mathrm{def}}{=} \{q\} \\
\Delta &\stackrel{\mathrm{def}}{=} \{\mathrm{S}^{(2)}\} \cup \Sigma \\
e &\stackrel{\mathrm{def}}{=} \mathrm{S}(q(\mathrm{x}_1), q(\mathrm{x}_1)) \\
R &\stackrel{\mathrm{def}}{=} \{q(\sigma(\mathrm{x}_1, \dots, \mathrm{x}_n)) \to \sigma(q(\mathrm{x}_1), \dots, q(\mathrm{x}_n)) \mid \sigma^{(n)} \in \Sigma\}
\end{aligned}
$$

□

この木変換は，2 つの木言語クラスの積による木言語を，逆像で表現できる例になる．

命題 93. 例 92 の InvRE_Σ について，任意の木言語 $T_1, T_2 \in \mathcal{P}(\hat{T}_\Sigma)$ は以下を満たす．

$$[\![\mathrm{InvRE}_\Sigma]\!]^{-1}[\{\mathrm{S}(t_1, t_2) \mid t_1 \in T_1, t_2 \in T_2\}] = T_1 \cap T_2$$

□

証明.

$$[\![\mathrm{InvRE}_\Sigma]\!] = t \mapsto \mathrm{S}(t, t)$$

であることは，命題 91 と同様に示せる．よって，

$$
\begin{aligned}
t \in [\![\mathrm{InvRE}_\Sigma]\!]^{-1}[\{\mathrm{S}(t_1, t_2) \mid t_1 \in T_1, t_2 \in T_2\}] &\iff (\exists t_1 \in T_1, t_2 \in T_2 \,.\, t = t_1 = t_2) \\
&\iff t \in T_1 \cap T_2
\end{aligned}
$$

より，題意は示された． ■

木変換が TDTT で表現できるかを判別する方法のひとつとして，**樹高性** (*height property*) を使う方法がある．樹高性は，出力木の高さが入力木の特性にどう依存するかを示す性質である．TDTT は，出力木の高さが入力木の高さに対して線形になるという，樹高性を持つ．

定理 94 (TDTT の樹高性 [FV98]). $[\![M]\!] : \hat{T}_\Sigma \to \hat{T}_\Delta \in \mathrm{TDTT}$ について，

$$\forall t \in \hat{T}_\Sigma \,.\, \mathrm{height}([\![M]\!](t)) \leq c_M \cdot \mathrm{height}(t)$$

を満たす $c_M \in \mathbb{N}$ が存在する． □

証明. TDTT $M = (Q, \Sigma, \Delta, e, R)$ について，

$$\begin{aligned} c_1 &\stackrel{\text{def}}{=} \mathrm{height}(e) \\ c_2 &\stackrel{\text{def}}{=} \max\{\mathrm{height}(\eta) \mid q(...) \to \eta \in R\} \\ c_M &\stackrel{\text{def}}{=} c_1 + c_2 \end{aligned}$$

とおく．この時，任意の $q \in Q$，$t \in \hat{T}_\Sigma$ について，

$$\mathrm{height}(\mathrm{nf}(\Rightarrow_M, q(t))) \leq c_2 \cdot \mathrm{height}(t)$$

を，t に関する構造的帰納法で示す．

IB1　適用不能．

IB2　$\sigma \in \Sigma^{(0)}$，$q(\sigma) \to \eta \in R$ について，

$$\mathrm{height}(\mathrm{nf}(\Rightarrow_M, q(\sigma))) = \mathrm{height}(\eta) \leq c_2 = c_2 \cdot \mathrm{height}(\sigma)$$

より正しい．

IS　$n \geq 1$, $\sigma^{(n)} \in \Sigma$, $t_1, ..., t_n \in \hat{T}_\Sigma$, $q(\sigma(...)) \to \eta \in R$ について，$c_2' = \max\{c_2 \cdot \mathrm{height}(t_i) \mid i \in [n]\}$ として，

$$\mathrm{height}(\mathrm{nf}(\Rightarrow_M, q(\sigma(t_1, ..., t_n)))) \leq \mathrm{height}(\eta) + c_2'$$

を，η に関する構造的帰納法で示す．

IB1　$q \in Q$，$i \in [n]$ について，$\eta = q(\mathrm{x}_i)$ の時，

$$\begin{aligned} \mathrm{height}(\mathrm{nf}(\Rightarrow_M, q(\sigma(t_1, ..., t_n)))) &= \mathrm{height}(\mathrm{nf}(\Rightarrow_M, q(t_i))) \\ &\leq c_2 \cdot \mathrm{height}(t_i) \\ &\qquad (\because \text{外側の帰納法の i.h.}) \\ &\leq \mathrm{height}(q(\mathrm{x}_i)) + c_2' \end{aligned}$$

より正しい．

IB2　$\delta \in \Delta^{(0)}$ について，$\eta = \delta$ の時，

$$\begin{aligned} \mathrm{height}(\mathrm{nf}(\Rightarrow_M, q(\sigma(t_1, ..., t_n)))) &= \mathrm{height}(\delta) \\ &\leq \mathrm{height}(\delta) + c_2' \end{aligned}$$

より正しい．

IS　$n \geq 1$，$\delta^{(k)} \in \Delta$ について，$\eta = \delta(\eta_1, ..., \eta_k)$ の時，$\eta_i' = \mathrm{nf}(\Rightarrow_M, \eta_i[\mathrm{x}_j \leftarrow t_j]_{j \in [n]})$ とすると，

$$\mathrm{height}(\mathrm{nf}(\Rightarrow_M, q(\sigma(t_1, ..., t_n)))) = 1 + \max\{\mathrm{height}(\eta_i') \mid i \in [k]\}$$

$$\begin{aligned}
&\leq 1 + \max\{\mathrm{height}(\eta_i) \mid i \in [k]\} + c_2' \\
&\qquad (\because \text{i.h.}) \\
&= \mathrm{height}(\delta(\eta_1, \dots, \eta_k)) + c_2'
\end{aligned}$$

より正しい．

よって，$\mathrm{height}(\eta) + c_2' \leq c_2 \cdot \mathrm{height}(\sigma(t_1, \dots, t_n))$ より正しい．

ここから，

$$\mathrm{height}(\mathrm{nf}(\Rightarrow_M, e[\mathrm{x}_1 \leftarrow t])) \leq c_1 + c_2 \cdot \mathrm{height}(t)$$

は e に関する容易な構造的帰納法で示せる．よって，

$$\begin{aligned}
\mathrm{height}([\![M]\!](t)) &= \mathrm{height}(\mathrm{nf}(\Rightarrow_M, e[\mathrm{x}_1 \leftarrow t])) \\
&\leq c_1 + c_2 \cdot \mathrm{height}(t) \\
&\leq (c_1 + c_2) \cdot \mathrm{height}(t) \\
&= c_M \cdot \mathrm{height}(t).
\end{aligned}$$

■

なお，$c_M = 1$ になる TDTT M として，恒等変換がある．

系 95. ランク付きアルファベット Σ について，以下が成り立つ．

$$\forall t \in \hat{T}_\Sigma \,.\, \mathrm{height}([\![\mathrm{Id}_\Sigma]\!](t)) = \mathrm{height}(t)$$

□

証明. 命題 91 より． ■

TDTT の部分クラスとして，線形 TDTT を導入する．線形 TDTT は線形サイズ増加 [EM03b]，正規木言語の保存などの有用な性質を満たす．

定義 96 (線形トップダウン木変換器). TDTT $M = (Q, \Sigma, \Delta, e, R)$ が線形とは，以下を満たすことを言う．

- $|\mathrm{occ}_e(\mathrm{x}_1)| \leq 1$.
- $\forall q(\sigma(\mathrm{x}_1, \dots, \mathrm{x}_n)) \to \eta \in R, i \in [n] \,.\, |\mathrm{occ}_\eta(\mathrm{x}_i)| \leq 1$.

この時，木変換のクラス $\{[\![M]\!] \mid M$ は線形 TDTT$\}$ を $\mathrm{TDTT}_{\mathrm{lu}}$ と表記する． □

特に，木から単項の木への TDTT は，線形である．

命題 97. TDTT $M = (Q, \Sigma, \Delta, e, R)$ について，Δ が単項ならば M は線形． □

証明. 規則の右辺に関する容易な構造的帰納法で示せる． ■

3.2 属性付き木変換器

属性付き木変換器 (ATT) は，入力の木に対して，各ノードに属性を割り当てその属性の依存関係を辿り，木を上下左右に走査しながら出力木を構築していく計算の，構文的なモデルになっている．

定義 98 (属性付き木変換器 (ATT)). $\mathrm{AS} = \{\mathrm{syn}, \mathrm{inh}\}$ と表記する．属性付き木変換器は，以下の要素の組 $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ である．

$A = \{A_k\}_{k \in \mathrm{AS}}$　互いに素な属性記号のランク付きアルファベットの族で，$A = \{A_k^{(1)}\}_{k \in \mathrm{AS}}$．
　　なお，$a \in A_{\mathrm{syn}}$ を合成属性，$b \in A_{\mathrm{inh}}$ を継承属性と呼ぶ．

Σ　　入力記号のランク付きアルファベット．

Δ　　出力記号のランク付きアルファベット．

$\sharp \notin \Sigma$　ルート記号のランク付き記号．$\Sigma \cup \{\sharp^{(1)}\}$ を $\Sigma_\sharp$ と表記する．

$a_0 \in A_{\mathrm{syn}}$　初期合成属性．

R　　書き換え規則の集合で，以下を満たす．

$$R \in \mathcal{P}_{fin}(\{\varphi \xrightarrow{\sigma} \eta \mid \sigma^{(n)} \in \Sigma_\sharp, \varphi \in \mathrm{GLHS}_{M,\sigma}([n]), \eta \in \mathrm{GRHS}_{M,\sigma}(\{\mathrm{w}\}, [n])\})$$

ただし，

$$\mathrm{GLHS}_{M,\sigma}(I) \stackrel{\mathrm{def}}{=} \{\varphi \in \mathrm{LHS}_M(I) \mid \sigma \neq \sharp \vee \forall a \in (A_{\mathrm{syn}} \setminus \{a_0\}) \,.\, \mathrm{occ}_\varphi(a) = \emptyset\}$$
$$\mathrm{GRHS}_{M,\sigma}(P, I) \stackrel{\mathrm{def}}{=} \{\eta \in \mathrm{RHS}_M(P, I) \mid \sigma \neq \sharp \vee \forall b \in A_{\mathrm{inh}} \,.\, \mathrm{occ}_\eta(b) = \emptyset\}$$
$$\mathrm{LHS}_M(I) \stackrel{\mathrm{def}}{=} \{a(\mathrm{w}) \mid a \in A_{\mathrm{syn}}\} \cup \{b(\mathrm{w}i) \mid b \in A_{\mathrm{inh}}, i \in I\}$$

で，$\mathrm{RHS}_M(P, I) \subseteq \hat{T}_{\Delta \cup \bigcup_{k \in \mathrm{AS}} A_k}(P \cup \{\pi i \mid \pi \in P, i \in I\})$ は，以下のように帰納的に定義される集合 U である．

IB1　$a \in A_{\mathrm{syn}}, \pi \in P, i \in I$ について，$a(\pi i) \in U$．

IB2　$b \in A_{\mathrm{inh}}, \pi \in P$ について，$b(\pi) \in U$．

IB3　$\delta \in \Delta^{(0)}$ について，$\delta \in U$．

IS　　$n \geq 1, \delta^{(n)} \in \Delta, \eta_1, \ldots, \eta_n \in U$ について，$\delta(\eta_1, \ldots, \eta_n) \in U$．

□

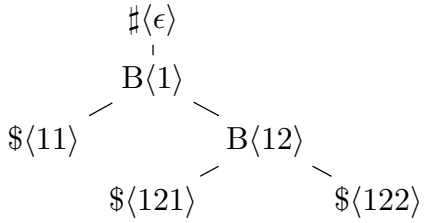
ATT は，現在の属性と入力木のパスが指すノードの根のラベルを見て適用する規則を選び，その規則の右辺の通りに出力を構築しながら新たに適用する規則を選んでいく．

例 99. ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ を以下のように定義する．

$$A_{\mathrm{syn}} \stackrel{\mathrm{def}}{=} \{a_0, a\}$$

$$
\begin{aligned}
A_{\mathrm{inh}} &\stackrel{\mathrm{def}}{=} \{b\} \\
\Sigma &\stackrel{\mathrm{def}}{=} \{\mathrm{B}^{(2)}, \$^{(0)}\} \\
\Delta &\stackrel{\mathrm{def}}{=} \{\mathrm{S}^{(1)}, \mathrm{SZ}^{(0)}\} \\
R &\stackrel{\mathrm{def}}{=} \{a_0(\mathrm{w}) \xrightarrow{\sharp} a_0(\mathrm{w1}), \\
&\qquad a_0(\mathrm{w}) \xrightarrow{\mathrm{B}} a(\mathrm{w1}), \\
&\qquad a_0(\mathrm{w}) \xrightarrow{\$} b(\mathrm{w}), \\
&\qquad a(\mathrm{w}) \xrightarrow{\mathrm{B}} \mathrm{S}(a(\mathrm{w1})), \\
&\qquad a(\mathrm{w}) \xrightarrow{\$} \mathrm{S}(b(\mathrm{w})), \\
&\qquad b(\mathrm{w1}) \xrightarrow{\sharp} \mathrm{SZ}, \\
&\qquad b(\mathrm{w1}) \xrightarrow{\mathrm{B}} a(\mathrm{w2}), \\
&\qquad b(\mathrm{w2}) \xrightarrow{\mathrm{B}} b(\mathrm{w})\}
\end{aligned}
$$

この M は，与えられた木のサイズを計算する ATT で，以下のような入力木を $\sharp$ に割り当てて作られる木に対して，パスによって遷移していく．



$\langle$ と $\rangle$ に囲まれた部分は，そのノードへのパスである．この時，M は次のような遷移をする．

$$
\begin{aligned}
a_0(\epsilon) &\stackrel{\sharp}{\Rightarrow} a_0(1) \stackrel{\mathrm{B}}{\Rightarrow} a(11) \stackrel{\$}{\Rightarrow} \mathrm{S}(b(11)) \stackrel{\mathrm{B}}{\Rightarrow} \mathrm{S}(a(12)) \\
&\stackrel{\mathrm{B}}{\Rightarrow} \cdots \stackrel{\mathrm{B}}{\Rightarrow} \mathrm{S}^4(b(12)) \stackrel{\mathrm{B}}{\Rightarrow} \mathrm{S}^4(b(1)) \stackrel{\sharp}{\Rightarrow} \mathrm{S}^4(\mathrm{SZ})
\end{aligned}
$$

ここから，以下のような変換を定義する．

$$
[\![M]\!]\left(\begin{array}{ccccc} & & \mathrm{B} & & \\ & \$ & & \mathrm{B} & \\ & & \$ & & \$ \end{array}\right) = \mathrm{S}^4(\mathrm{SZ})
$$

なお，M の意味論 $[\![M]\!]$ の形式的な定義は，後ほど導入する． □

TDTT と同様，ATT も一般に関数でない関係を意味論に持つが，本論文では関係が関数の場合のみを扱う．そして，そのような構文的制約として全域的決定的 ATT を導入する．

定義 100 (全域的決定的 ATT). ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ が

$$
\forall \sigma^{(n)} \in \Sigma_{\sharp}, \varphi \in \mathrm{GLHS}_{M,\sigma}([n]) \,.\, \exists! \varphi \xrightarrow{\sigma} \eta \in R
$$

を満たす時，全域的決定的であるという． □

全域的決定的 ATT とは，入力木のアルファベットと属性を規則の左辺が全て網羅しており，またただひとつだけの規則が対応するという制約である．以降，特に断りのない限り ATT は全域的決定的であるものとする．ATT について，意味論を決定づける簡約システムを導入する．

定義 101 (ATT の簡約システム). ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_{\Sigma_\sharp}$ について，簡約システム $(\mathrm{SF}_M(t), \Rightarrow_{M,t})$ を，以下のように定義する．

$$\mathrm{SF}_M(t) \stackrel{\mathrm{def}}{=} \mathrm{RHS}_M(\mathrm{paths}(t), \mathbb{N})$$

$$\varphi_1 \Rightarrow_{M,t} \varphi_2 \text{ iff } \left(\begin{array}{l} \exists a(w) \xrightarrow{\sigma} \eta \in R, \\ \quad p = w[\mathrm{w} \leftarrow p'], \mathrm{label}_t(p') = \sigma, \\ \quad \pi \in \mathrm{paths}(\varphi_1), \mathrm{subtree}_{\varphi_1}(\pi) = a(p)\ . \\ \varphi_2 = \varphi_1[\pi \leftarrow \eta[\mathrm{w} \leftarrow p']] \end{array} \right)$$

□

また，利便性のため，以下の表記を導入する．

定義 102. ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について，

$$\mathrm{attr}(M, t) \stackrel{\mathrm{def}}{=} \{a(p) \mid a \in \bigcup_{k \in \mathrm{AS}} A_k, p \in \mathrm{paths}(\sharp(t))\}$$

と定義する． □

一般の ATT において，簡約システムは停止性を持たない．これは，ATT が全域的決定的であっても意味論が関数にならない場合があることを意味する．そのような場合を除くため，*well-defined* の概念を導入する．

定義 103 (ATT の well-defined 性). ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，

$$\forall t \in \hat{T}_\Sigma, a(p) \in \mathrm{attr}(M, t)\ .\ a(p) \text{ は } \Rightarrow_{M,\sharp(t)} \text{ で無限簡約可能でない}$$

を満たす時，M は well-defined であるという． □

ATT の well-defined 性の同値条件に，**非巡回性** (*non-circular*) が存在する．非巡回性とは，全てのノードの全ての属性の簡約結果が，同じノードの同じ属性の簡約結果に依存しないことを意味する．

定義 104 (ATT の非巡回性). ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，

$$\exists \eta \in \mathrm{SF}_M(t), \mathrm{occ}_\eta(a(p)) \neq \emptyset\ .\ a(p) \Rightarrow^+_{M,\sharp(t)} \eta$$

を満たす $t \in \hat{T}_\Sigma$，$a(p) \in \mathrm{attr}(M, t)$ が存在する時，M は巡回であるという．M が巡回でない時，M は非巡回であるという． □

これらが同値条件であることを示すために，ATT の**依存グラフ** (*dependency graph*) の概念を導入する．依存グラフとは，属性同士の依存関係を表した有向グラフである．

定義 105 (ATT の依存グラフ). ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，集合の族 $\{\mathrm{attr_{in}}(M)_\sigma\}_{\sigma\in\Sigma_\sharp}$，$\{\mathrm{attr_{out}}(M)_\sigma\}_{\sigma\in\Sigma_\sharp}$ を以下のように定義する．

$$\mathrm{attr_{in}}(M)_{\sigma^{(n)}} \stackrel{\mathrm{def}}{=} \{a(\mathrm{w}) \mid a \in A_{\mathrm{syn}}\} \cup \{b(\mathrm{w}i) \mid b \in A_{\mathrm{inh}}, i \in [n]\}$$
$$\mathrm{attr_{out}}(M)_{\sigma^{(n)}} \stackrel{\mathrm{def}}{=} \{a(\mathrm{w}i) \mid a \in A_{\mathrm{syn}}, i \in [n]\} \cup \{b(\mathrm{w}) \mid b \in A_{\mathrm{inh}}\}$$

この時，M の依存グラフ $\mathrm{depgraph}(M)$ とは，次のように定義される関係の族 $G \in \prod_{\sigma\in\Sigma} \mathrm{attr_{in}}(M)_\sigma \times \mathrm{attr_{out}}(M)_\sigma$ のこと．

$$G_\sigma \stackrel{\mathrm{def}}{=} \{(a_1(w_1), a_2(w_2)) \mid a_1(w_1) \xrightarrow{\sigma} \eta \in R, \mathrm{occ}_\eta(a_2(w_2)) \neq \emptyset\}$$

この時，簡約システム $(\mathrm{attr}(M,t), \Rightarrow_{G,t})$ を次のように定義する．

$$a_1(p_1) \Rightarrow_{G,t} a_2(p_2) \text{ iff } \left(\begin{array}{l} \exists p \in \mathrm{paths}(\sharp(t)), \sigma = \mathrm{label}_{\sharp(t)}(p), \\ \quad (a_1(w_1), a_2(w_2)) \in G_\sigma \ . \\ w_1[\mathrm{w} \leftarrow p] = p_1 \wedge w_2[\mathrm{w} \leftarrow p] = p_2 \end{array} \right)$$

□

依存グラフの簡約システムにおいて，属性とパスの組が別の組に簡約できるということは，その組が簡約結果の組に依存しているということを示す．そして，属性とパスの組は，ATT の簡約システムにおいて依存関係を持つ属性とパスの組が出現する木に簡約できる．

補題 106. ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$，$G = \mathrm{depgraph}(M)$，$a_1(p_1), a_2(p_2) \in \mathrm{attr}(M,t)$ について，以下の 2 条件は同値である．

- $a_1(p_1) \Rightarrow^+_{G,t} a_2(p_2)$.
- $a_1(p_1) \Rightarrow^+_{M,\sharp(t)} \eta$ で，$\mathrm{occ}_\eta(a_2(p_2)) \neq \emptyset$ を満たす η が存在する．

□

証明. $a_1(p_1) \Rightarrow_{G,t} a_2(p_2)$ について，定義より $a_1(p_1) \Rightarrow_{M,\sharp(t)} \eta(a_2(p_2))$ と書ける η が存在することは明らか．よって，$a_1(p_1) \Rightarrow^m_{G,t} a_2(p_2)$ について，$a_1(p_1) \Rightarrow^+_{M,\sharp(t)} \eta(a_2(p_2))$ となる η が存在することは，容易な m についての数学的帰納法で示せる．逆も同様に示せる．よって，題意は示された． ■

ここから，依存グラフに対し，その非巡回性を定義できる．

定義 107. ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，依存グラフ $G = \mathrm{depgraph}(M)$ が巡回であるとは，$a(p) \Rightarrow^+_{G,t} a(p)$ を満たす $t \in \hat{T}_\Sigma$，$a(p) \in \mathrm{attr}(M,t)$ が存在することをいう． G が巡回でない時，G は非巡回であるという． □

依存グラフが非巡回であるとは，すなわち自身に依存する属性とパスの組が存在しないということである．ところで，依存グラフが非巡回であるならば，依存グラフの簡約システムは停止性を持つ．つまり，全ての簡約列は必ず有限の長さを持つ．なぜなら，属性とパスの組の集合は有限になるからである．

補題 108. ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，$G = \mathrm{depgraph}(M)$ が非巡回ならば，任意の $t \in \hat{T}_\Sigma$ について，以下を満たす $c_{M,t} \in \mathbb{N}$ が存在する．

$$\forall a(p) \in \mathrm{attr}(M,t)\ .\ a(p) \text{ の } \Rightarrow_{G,t} \text{ での任意の簡約の長さは } c_{M,t} \text{ 以下である}$$

□

証明. $c_{M,t} = |\mathrm{attr}(M,t)|$ とする．任意の $n > c_{M,t}$ について，以下を満たす $\{a_n(p_n)\}_{n\in\mathbb{N}}$ が存在したとする．

$$a_0(p_0) \Rightarrow_{G,t} a_1(p_1) \Rightarrow_{G,t} \cdots \Rightarrow_{G,t} a_n(p_n)$$

この時，$0 \leq i_1 < i_2 \leq c_{M,t}$ で $a_{i_1}(p_{i_1}) = a_{i_2}(p_{i_2})$ を満たすものが存在するが，これは G が非巡回に矛盾する．よって，題意は示された． ■

ところで，ATT の依存グラフが非巡回である時，ATT の簡約システムは停止性を持つ．

補題 109. ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，$\mathrm{depgraph}(M)$ が非巡回ならば，任意の $t \in \hat{T}_\Sigma$ について $\Rightarrow_{M,\sharp(t)}$ は停止性を持つ． □

証明. $G = \mathrm{depgraph}(M)$ が非巡回とする．この時，$a(p) \in \mathrm{attr}(M,t)$，$n \in \mathbb{N}$ について，$l(a(p), n) \in \mathbb{N}$ を，以下のように n に関する数学的帰納法で定義する．

IB　$l(a(p), 0) = 0$.

IS　以下のように場合分けを行う．

- $\mathrm{label}_{\sharp(t)}(p') = \sigma$ かつ $w[\mathrm{w} \leftarrow p'] = p$ となる $a(w) \xrightarrow{\sigma} \eta \in R$，$p' \in \mathrm{paths}(\sharp(t))$ が存在する時，$l(a(p), k+1) = 1 + \mathrm{count}(\eta[\mathrm{w} \leftarrow p'], k)$.
- それ以外の時，$l(a(p), k+1) = 0$.

ただし，$\mathrm{count}(\eta, n)$ は以下のように構造的帰納法で定義する．

IB1　$a \in A_{\mathrm{syn}}$，$p \in \mathrm{paths}(\sharp(t))$ について，$\mathrm{count}(a(p), n) = l(a(p), n)$.

IB2　$b \in A_{\mathrm{inh}}$，$p \in \mathrm{paths}(\sharp(t))$ について，$\mathrm{count}(b(p), n) = l(b(p), n)$.

IB3　$\delta \in \Delta^{(0)}$ について，$\mathrm{count}(\delta, n) = 0$.

IS　$k \geq 1$，$\delta^{(k)} \in \Delta$，$\eta_1, \ldots, \eta_k$ について，

$$\mathrm{count}(\delta(\eta_1, \ldots, \eta_k), n) = \sum_{i\in[k]} \mathrm{count}(\eta_i, n).$$

この時，任意の $n \in \mathbb{N}$，$a(p) \in \mathrm{attr}(M,t)$ について，$a(p)$ が $l(a(p),n)$ 回の簡約で正規形にならないとすると，$a(p) \Rightarrow_{G,t}^{n+1} a'(p')$ を満たす $a'(p') \in \mathrm{attr}(M,t)$ が存在することは，n に関する容易な数学的帰納法で確かめられる．

任意の $\eta \in \mathrm{SF}_M(t)$，補題 108 の $c_{M,t}$ について，η の簡約が $\mathrm{count}(\eta, c_{M,t})$ 回以下で終わることを，構造的帰納法で証明する．

IB1　$a \in A_{\mathrm{syn}}$，$p \in \mathrm{paths}(\sharp(t))$ について，$a(p)$ の簡約が $\mathrm{count}(a(p), c_{M,t}) = l(a(p), c_{M,t})$ 回以下で終わらないとする．この時，$a(p) \Rightarrow_{G,t}^{c_{M,t}+1} a'(p')$ となる $a'(p') \in \mathrm{attr}(M,t)$ が存在するが，これは補題 108 に矛盾する．よって，正しい．

IB2　IB1 と同様．

IB3　$\delta \in \Delta^{(0)}$ について，$\mathrm{count}(\delta, c_{M,t}) = 0$ で δ は正規形より正しい．

IS　$n \geq 1$，$\delta^{(n)} \in \Delta$，$\eta_1, \ldots, \eta_n$ について，

$$\mathrm{count}(\delta(\eta_1, \ldots, \eta_n), c_{M,t}) = \sum_{i \in [n]} \mathrm{count}(\eta_i, c_{M,t})$$

であり，i.h. より正しい．

よって，任意の $\eta \in \mathrm{SF}_M(t)$ で簡約は停止するため，題意は示された． ■

ここから，ATT の well-defined 性と，ATT が非巡回であること及び依存グラフが非巡回であることが，同値条件であることが分かる．

定理 110. ATT M について，以下の 3 条件は同値である．

1. M は well-defined.
2. M は非巡回.
3. $\mathrm{depgraph}(M)$ は非巡回.

□

証明. 2 ならば 3 を示す．M が非巡回である時，$G = \mathrm{depgraph}(M)$ が巡回であると仮定する．この時，$a(p) \Rightarrow_{G,t}^{+} a(p)$ を満たす $t \in \hat{T}_\Sigma$，$a(p) \in \mathrm{attr}(M,t)$ が存在するため，補題 106 より M は巡回だがこれは矛盾．よって，正しい．

1 ならば 2 を示す．M が well-defined である時，M は巡回であると仮定する．この時，以下を満たす $t \in \hat{T}_\Sigma$，$a(p) \in \mathrm{attr}(M,t)$，$n > 0$，$\eta_0, \ldots, \eta_n$，$\pi \in \mathrm{occ}_{\eta_n}(a(p))$ が存在する．

- $a(p) = \eta_0$.
- $\eta_0 \Rightarrow_{M,\sharp(t)} \eta_1 \Rightarrow_{M,\sharp(t)} \cdots \Rightarrow_{M,\sharp(t)} \eta_n$.

$\varphi = \eta_n[\pi \leftarrow *]$ とすると，$n' > n$ について，$n' = c_1 n + c_2$ と書ける $c_1 \in \mathbb{N}$，$0 \leq c_2 < n$ が存在し，$\eta_{n'} = \varphi^{c_1}(\eta_{c_2})$ とすると，$\{\eta_n\}_{n \in \mathbb{N}}$ は $a(p)$ の無限簡約列になる．これは M が well-defined

に矛盾する．よって，正しい．

3 ならば 1 は，補題 109 より明らか．よって，推移律より，題意は示された． ■

また，ATT の依存グラフの非巡回性は決定可能であることが知られている [Fül81]．ここから，ATT の well-defined 性は決定可能である．この判定アルゴリズムが，アルゴリズム 1 になる．このアルゴリズムは，依存グラフから同じノード上でのありえる全ての属性の依存関係に対し，依存グラフと依存関係の全てでの組み合わせで巡回するかどうかを確かめる．属性の依存関係は，高々属性の組の個数しか存在せず，属性は有限であるため組も有限である．よって，この確認は有限回で終了する．

定理 111 ([Fül81]). ATT M について，M が well-defined かは決定可能． □

証明. 定理 110 より，$G = \mathrm{depgraph}(M)$ が非巡回かが決定可能であることと，題意は等しい．これはアルゴリズム 1 によって決定できる．この正当性は，[FV98] の観察 5.18 より分かる． ■

以降，特に断りのない限り ATT は well-defined であるものとする． ATT の簡約システムは局所合流性を持つ．

補題 112. ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について，$\Rightarrow_{M,\sharp(t)}$ は局所合流性を持つ． □

証明. [FV98] の補題 5.20 より． ■

また，well-defined 性から ATT の簡約システムは停止性を持つため，合流性を持つ．

系 113. ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について，$\Rightarrow_{M,\sharp(t)}$ は合流性を持つ． □

証明. 補題 109，定理 110，補題 112，補題 56 より． ■

ATT の簡約システムは，合流性と停止性を持つため，補題 57 より一意な正規形を持つ．また，根の初期属性からの正規形は，必ず出力アルファベットのみからなる木になる．

補題 114. ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について，$\mathrm{nf}(\Rightarrow_{M,\sharp(t)}, a_0(\epsilon)) \in \hat{T}_\Delta$ □

証明. [FV98] の補題 5.32 より． ■

よって，ATT の簡約システムにおいて，入力木の根の初期属性の正規形は，必ず一意でかつ出力木になる．この時 ATT に対して，入力木を受け取り，その根の初期属性から ATT の簡約システムで得られる木を出力とする，木変換が考えられる．この木変換を，意味論として導入する．

定義 115 (ATT の意味論). ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，$[\![M]\!] : \hat{T}_\Sigma \to \hat{T}_\Delta$ を以下のように定義する．

$$[\![M]\!] \stackrel{\mathrm{def}}{=} t \mapsto \mathrm{nf}(\Rightarrow_{M,\sharp(t)}, a_0(\epsilon))$$

アルゴリズム 1 ATT の依存グラフの非巡回性検査

Require: ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$G = \mathrm{depgraph}(M)$

$D \leftarrow \emptyset$

loop

$D' \leftarrow \emptyset$

for all $\sigma^{(n)} \in \Sigma_\sharp$，$d_1, \ldots, d_n \in D$ **do**

$g \leftarrow G_\sigma(d_1, \ldots, d_n)$

if $\exists (a(w), a(w)) \in g$ **then**

return 巡回

end if

$d \leftarrow \{(a_1, a_2) \mid a_1, a_2 \in \bigcup_{k \in \mathrm{AS}} A_k, (a_1(\mathrm{w}), a_2(\mathrm{w})) \in g\}$

$D' \leftarrow D' \cup \{d\}$

end for

if $D' \subseteq D$ **then**

return 非巡回

else

$D \leftarrow D' \cup D$

end if

end loop

ただし，$G_\sigma(d_1, \ldots, d_n)$ は，以下のように定義する．

$$G_\sigma(d_1, \ldots, d_n) = \left\{ (a_0(w_0), a_n(w_n)) \;\middle|\; \begin{array}{l} n > 0, \\ a_0(w_0), \ldots, a_n(w_n) \in \mathrm{attr}_{\mathrm{in}}(M)_\sigma \cup \mathrm{attr}_{\mathrm{out}}(M)_\sigma, \\ \forall 0 \leq i < n \,.\, (a_i(w_i), a_{i+1}(w_{i+1})) \in g' \end{array} \right\}$$

$$\left(g' = G_\sigma \cup \bigcup_{i \in [n]} \{(a_1(\mathrm{w}i), a_2(\mathrm{w}i)) \mid (a_1, a_2) \in d_i\} \right)$$

また，木変換のクラス $\{[\![M]\!] \mid M$ は (全域的決定的 well-defined) ATT$\}$ を ATT と表記する．

□

なお，ATT が全域的決定的な場合，ATT の簡約システムを決定的にすることができる．

命題 116 (ATT の決定的な簡約システム). ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_{\Sigma_\sharp}$ について，簡約システム $(\mathrm{SF}_M(t), \Rightarrow^D_{M,t})$ を，$U \stackrel{\mathrm{def}}{=} \mathrm{SF}_M(t)$ として，以下のように構造的帰納法で定義する．

IB1　$a \in A_{\mathrm{syn}}$，$\sigma \in \Sigma_\sharp$，$p \in \mathrm{paths}(t)$，$\mathrm{label}_t(p) = \sigma$，$a(\mathrm{w}) \xrightarrow{\sigma} \eta \in R$ について，

$$a(p) \Rightarrow^D_{M,t} \eta[\mathrm{w} \leftarrow p].$$

IB2　$b \in A_{\mathrm{inh}}$，$\sigma \in \Sigma_\sharp$，$p \in \mathrm{paths}(t)$，$\mathrm{label}_t(p) = \sigma$，$b(\mathrm{w}i) \xrightarrow{\sigma} \eta \in R$ について，

$$b(pi) \Rightarrow^D_{M,t} \eta[\mathrm{w} \leftarrow p].$$

IB3　$\delta \in \Delta^{(0)}$ について，δ は既約

IS　$n \geq 1$，$\delta^{(n)} \in \Delta$，$\eta_1, \ldots, \eta_n \in U$，$i \in [n]$ について，$\eta_1, \ldots, \eta_{i-1}$ が $\Rightarrow^D_{M,t}$ で既約で，$\eta_i \Rightarrow^D_{M,t} \eta'_i$ の時，

$$\delta(\eta_1, \ldots, \eta_i, \ldots, \eta_n) \Rightarrow^D_{M,t} \delta(\eta_1, \ldots, \eta'_i, \ldots, \eta_n).$$

この時，任意の $\eta \in U$ に対して，$\mathrm{nf}(\Rightarrow_{M,t}, \eta) = \mathrm{nf}(\Rightarrow^D_{M,t}, \eta)$ □

証明. 命題 88 と同様に示せる. ■

例 117. 例 99 の ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，

$$t \stackrel{\mathrm{def}}{=} \sharp(\mathrm{B}(\$, \mathrm{B}(\$, \$)))$$

を考える．この時，以下のような簡約例が考えられる．

$$\begin{aligned}
a_0(\epsilon) &\Rightarrow_{M,t} a_0(1) \\
&\Rightarrow_{M,t} a(11) \\
&\Rightarrow_{M,t} \mathrm{S}(b(11)) \\
&\Rightarrow_{M,t} \mathrm{S}(a(12)) \\
&\Rightarrow_{M,t} \mathrm{S}(\mathrm{S}(a(121))) \\
&\Rightarrow_{M,t} \mathrm{S}(\mathrm{S}(\mathrm{S}(b(121)))) \\
&\Rightarrow_{M,t} \mathrm{S}(\mathrm{S}(\mathrm{S}(a(122)))) \\
&\Rightarrow_{M,t} \mathrm{S}(\mathrm{S}(\mathrm{S}(\mathrm{S}(b(122))))) \\
&\Rightarrow_{M,t} \mathrm{S}(\mathrm{S}(\mathrm{S}(\mathrm{S}(b(12))))) \\
&\Rightarrow_{M,t} \mathrm{S}(\mathrm{S}(\mathrm{S}(\mathrm{S}(b(1))))) \\
&\Rightarrow_{M,t} \mathrm{S}(\mathrm{S}(\mathrm{S}(\mathrm{SZ})))
\end{aligned}$$

ここから，意味論による変換は以下のようになる．

$$[\![M]\!] \left(\begin{array}{ccccc} & & \mathrm{B} & & \\ & \swarrow & & \searrow & \\ \$ & & & & \mathrm{B} \\ & & & \swarrow & & \searrow \\ & & \$ & & & \$ \end{array} \right) = \mathrm{S}^4(\mathrm{SZ})$$

□

ところで，合成属性だけの ATT は，TDTT と対応する．よって，TDTT のクラスは，ATT のクラスに含まれる．

定理 118 ([Fül81]). TDTT $\subseteq$ ATT □

証明. TDTT $M = (Q, \Sigma, \Delta, e, R)$ について，ATT $M' = (A', \Sigma, \Delta, \sharp, a_0, R')$ を，$q_0 \in Q$ を適当な状態記号として，以下のように定義する．

$$
\begin{aligned}
A'_{\mathrm{syn}} &\stackrel{\mathrm{def}}{=} Q \\
A'_{\mathrm{inh}} &\stackrel{\mathrm{def}}{=} \emptyset \\
a_0 &\stackrel{\mathrm{def}}{=} q_0 \\
R' &\stackrel{\mathrm{def}}{=} \{a_0(\mathrm{w}) \xrightarrow{\sharp} e[\mathrm{x}_1 \leftarrow \mathrm{w}1]\} \\
&\quad \cup \{q(\mathrm{w}) \xrightarrow{\sigma} \eta[\mathrm{x}_i \leftarrow \mathrm{w}i]_{i\in[n]} \mid \sigma^{(n)} \in \Sigma, q(\sigma(...)) \to \eta \in R\}
\end{aligned}
$$

この時，$[\![M]\!] = [\![M']\!]$ を示す．まず，任意の $t \in \hat{T}_\Sigma$ について，

$$\mathrm{conv}_t(\eta) \stackrel{\mathrm{def}}{=} \eta[\pi \leftarrow \mathrm{subtree}_t(\pi)]_{\pi\in\mathrm{paths}(t)}$$

として，

$$\eta_1 \Rightarrow_{M,t} \eta_2 \text{ implies } \mathrm{conv}_t(\eta_1) \Rightarrow_M \mathrm{conv}_t(\eta_2)$$

を示す．これは η_1 に関する構造的帰納法より容易に示せる．次に，

$$\mathrm{nf}(\Rightarrow_{M,\sharp(t)}, a_0(\epsilon)) = \mathrm{nf}(\Rightarrow_{M,t}, e[\mathrm{x}_1 \leftarrow \epsilon])$$

を示す．これは，e に関する構造的帰納法より容易に示せる．よって，

$$\mathrm{nf}(\Rightarrow_{M,\sharp(t)}, a_0(\epsilon)) = \mathrm{nf}(\Rightarrow_{M,t}, e[\mathrm{x}_1 \leftarrow \epsilon]) = \mathrm{nf}(\Rightarrow_M, e[\mathrm{x}_1 \leftarrow t])$$

より，題意は示された． ■

ATT において継承属性を使用することで，単項の入力木を複製し，その木を繋げて 1 つの木にするような木変換が書ける．

例 119. 単項ランク付きアルファベット Σ について，ATT $\mathrm{InvMRE}_\Sigma = (A, \Sigma, \Delta, \sharp, a_0, R)$ を以下のように定義する．

$$
\begin{aligned}
A_{\mathrm{syn}} &\stackrel{\mathrm{def}}{=} \{a_0, a_1\} \\
A_{\mathrm{inh}} &\stackrel{\mathrm{def}}{=} \{b\} \\
\Delta &\stackrel{\mathrm{def}}{=} \{\mathrm{S}^{(1)}_\sigma \mid \sigma \in \Sigma^{(0)}\} \cup \Sigma \\
R &\stackrel{\mathrm{def}}{=} \{a_0(\mathrm{w}) \xrightarrow{\sharp} a_0(\mathrm{w}1), b(\mathrm{w}1) \xrightarrow{\sharp} a_1(\mathrm{w}1)\}
\end{aligned}
$$

$$
\begin{aligned}
&\cup \{a(\mathrm{w}) \xrightarrow{\sigma} \sigma(a(\mathrm{w1})) \mid a \in A_{\mathrm{syn}}, \sigma \in \Sigma^{(1)}\} \\
&\cup \{b(\mathrm{w1}) \xrightarrow{\sigma} b(\mathrm{w}) \mid \sigma \in \Sigma^{(1)}\} \\
&\cup \{a_0(\mathrm{w}) \xrightarrow{\sigma} \mathrm{S}_\sigma(b(\mathrm{w})) \mid \sigma \in \Sigma^{(0)}\} \\
&\cup \{a_1(\mathrm{w}) \xrightarrow{\sigma} \sigma \mid \sigma \in \Sigma^{(0)}\}
\end{aligned}
$$

□

この木変換も，例 92 のようにある種の木言語の積を表現できる．特に，結合に閉じている木言語クラスの単項の木言語について，その積を逆像で表現できる．

命題 120. 例 119 の InvMRE_Σ について，木言語 $T_1, T_2 \in \mathcal{P}(\hat{T}_\Sigma)$ を考える．この時，以下が成り立つ．

$$
[\![M]\!]^{-1}[\{\mathrm{concat}(t_1, t_2) \mid t_1 \in T_1, t_2 \in T_2\}] = T_1 \cap T_2
$$

ただし，$\mathrm{concat}(t_1(\sigma), t_2) = t_1(\mathrm{S}_\sigma(t_2))$ である． □

証明. $\mathrm{InvMRE}_\Sigma = (A, \Sigma, \Delta, \sharp, a_0, R)$，$A_{\mathrm{syn}} = \{a_0, a_1\}$，$A_{\mathrm{inh}} = \{b\}$ について考える．任意の $t \in \hat{T}_\Sigma$，$p \in \mathrm{paths}(\sharp(t))$ について，

$$
b(p) \Rightarrow^*_{M,\sharp(t)} b(1)
$$

は，t に関する容易な構造的帰納法で示せる．ここから，任意の $t \in \hat{T}_\Sigma$，$p \in \mathrm{paths}(\sharp(t)) \backslash \{\epsilon\}$ について，

- $a_0(p) \Rightarrow^*_{M,\sharp(t)} \mathrm{concat}(\mathrm{subtree}_{\sharp(t)}(p), b(\epsilon))$
- $a_1(p) \Rightarrow^*_{M,\sharp(t)} \mathrm{subtree}_{\sharp(t)}(p)$

は，$\mathrm{subtree}_{\sharp(t)}(p)$ に関する構造的帰納法より容易に示せる．よって，

$$
\begin{aligned}
[\![M]\!](t) &= \mathrm{nf}(\Rightarrow_{M,\sharp(t)}, a(\epsilon)) \\
&= \mathrm{nf}(\Rightarrow_{M,\sharp(t)}, a(1)) \\
&= \mathrm{concat}(t, \mathrm{nf}(\Rightarrow_{M,\sharp(t)}, b(1))) \\
&= \mathrm{concat}(t, \mathrm{nf}(\Rightarrow_{M,\sharp(t)}, a_1(1))) \\
&= \mathrm{concat}(t, t)
\end{aligned}
$$

となる．ここから，

$$
\begin{aligned}
t \in [\![M]\!]^{-1}[\{\mathrm{concat}(t_1, t_2) \mid t_1 \in T_1, t_2 \in T_2\}] &\iff (\exists t_1 \in T_1, t_2 \in T_2 \,.\, t = t_1 = t_2) \\
&\iff t \in T_1 \cap T_2
\end{aligned}
$$

より，題意は示された． ■

TDTT と同様，木変換が ATT で表現できるかを判別する方法のひとつとして，樹高性を使う方法がある．ATT は，出力木の高さが入力木のサイズに対して線形になるという，樹高性を持つ．

定理 121 (ATT の樹高性 [FV98]). $[\![M]\!] : \hat{T}_\Sigma \to \hat{T}_\Delta \in \mathrm{ATT}$ について，

$$\forall t \in \hat{T}_\Sigma \,.\, \mathrm{height}([\![M]\!](t)) \leq c_M \cdot \mathrm{size}(t)$$

を満たす $c_M \in \mathbb{N}$ が存在する． □

証明. ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，

$$\begin{aligned} c_1 &\stackrel{\mathrm{def}}{=} \max\{\mathrm{height}(\eta) \mid a(w) \xrightarrow{\sigma} \eta \in R\} \\ c_2 &\stackrel{\mathrm{def}}{=} \sum_{k \in \mathrm{AS}} |A_k| \\ c_M &\stackrel{\mathrm{def}}{=} 2 \cdot c_1 \cdot c_2 \end{aligned}$$

とおく．また，任意の $t \in \hat{T}_\Sigma$ について，$h_t : \mathrm{attr}(M,t) \times \mathbb{N} \to \mathbb{N}$ を，以下のように，数学的帰納法で定義する．

IB　$h_t(a(p), 0) \stackrel{\mathrm{def}}{=} 0$.

IS　以下のような場合分けを行う．

- $\sigma^{(n)} \in \Sigma_\sharp$，$a(w) \xrightarrow{\sigma} \eta \in R$，$p = w[\mathrm{w} \leftarrow p']$，$\mathrm{label}_t(p') = \sigma$ を満たす η，p' が存在する時，$h_t(a(p), k+1)$ は，以下のように η に関する構造的帰納法で定義された h_η とおく．
  - IB1　$a' \in A_{\mathrm{syn}}$，$i \in [n]$ について，$h_{a'(\mathrm{w}i)} = h_t(a'(p'i), k)$
  - IB2　$b' \in A_{\mathrm{inh}}$ について，$h_{b'(\mathrm{w})} = h_t(b'(p'), k)$
  - IB3　$\delta^{(0)} \in \Delta$ について，$h_\delta = 1$
  - IS　$n \geq 1$，$\delta^{(n)} \in \Delta$ について，$h_{\delta(\eta_1, \ldots, \eta_n)} = 1 + \max\{h_{\eta_i} \mid i \in [n]\}$
- それ以外の時，$h_t(a(p), k+1) \stackrel{\mathrm{def}}{=} 0$.

この時，$h_t(a(p), k) \leq k \cdot c_1$ であることは，k に関する数学的帰納法で容易に示せる．また，任意の $k \in \mathbb{N}$ について $h_t(a(p), k) < h_t(a(p), k')$ となる k' が存在する時，$a(p) \Rightarrow^{k+1}_{\mathrm{depgraph}(M),t} a'(p')$ となる $a'(p')$ が存在することも，k に関する数学的帰納法で容易に示せる．

さて，$\mathrm{height}(\mathrm{nf}(\Rightarrow_{M,\sharp(t)}, a(\epsilon))) > c_1 \cdot c_2 \cdot \mathrm{size}(\sharp(t))$ となる $t \in \hat{T}_\Sigma$ が存在するとする．この時，$h_t(a(p), c_2 \cdot \mathrm{size}(\sharp(t))) < h_t(a(p), k)$ となる k が存在し，補題 108 の $c_{M,t}$ について，$a(p) \Rightarrow^{c_{M,t}+1}_{\mathrm{depgraph}(M),t} a'(p')$ となる $a'(p')$ が存在する．しかしこれは，補題 108 に矛盾する．よって，

$$\mathrm{height}([\![M]\!](t)) = \mathrm{height}(\mathrm{nf}(\Rightarrow_{M,\sharp(t)}, a(\epsilon)))$$

$$
\begin{aligned}
&\leq c_1 \cdot c_2 \cdot \mathrm{size}(\sharp(t)) \\
&= c_1 \cdot c_2 + c_1 \cdot c_2 \cdot \mathrm{size}(t) \\
&\leq 2 \cdot c_1 \cdot c_2 \cdot \mathrm{size}(t) \\
&= c_M \cdot \mathrm{size}(t)
\end{aligned}
$$

となる. ■

なお，$c_M = 1$ になる ATT M として，例 99 がある.

命題 122. 例 99 の $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，以下が成り立つ.

$$\forall t \in \hat{T}_\Sigma \,.\, \mathrm{height}([\![M]\!](t)) = \mathrm{size}(t)$$

□

証明. $A_{\mathrm{syn}} = \{a_0, a\}$，$A_{\mathrm{inh}} = \{b\}$，$t \in \hat{T}_\Sigma$ について，

$$\forall p \in \mathrm{paths}(\sharp(t)) \backslash \{\epsilon\} \,.\, a(p) \Rightarrow^*_{M,\sharp(t)} \mathrm{S}^{\mathrm{size}(\mathrm{subtree}_{\sharp(t)}(p))}(b(p))$$

を，$\mathrm{subtree}_t(p)$ に関する構造的帰納法で示す.

IB1　適用不能

IB2　$\mathrm{subtree}_{\sharp(t)}(p) = \$$ の時，$a(p) \Rightarrow_{M,\sharp(t)} b(p)$ より正しい.

IS　$\mathrm{subtree}_{\sharp(t)}(p) = \mathrm{a}(t_1, t_2)$ の時，

$$
\begin{aligned}
a(p) &\Rightarrow_{M,\sharp(t)} \mathrm{S}(a(p1)) \\
&\Rightarrow^*_{M,\sharp(t)} \mathrm{S}^{1+\mathrm{size}(t_1)}(b(p1)) && (\because \text{i.h.}) \\
&\Rightarrow_{M,\sharp(t)} \mathrm{S}^{1+\mathrm{size}(t_1)}(a(p2)) \\
&\Rightarrow^*_{M,\sharp(t)} \mathrm{S}^{1+\mathrm{size}(t_1)+\mathrm{size}(t_2)}(b(p2)) && (\because \text{i.h.}) \\
&\Rightarrow_{M,\sharp(t)} \mathrm{S}^{1+\mathrm{size}(t_1)+\mathrm{size}(t_2)}(b(p)) \\
&= \mathrm{S}^{\mathrm{size}(\mathrm{a}(t_1,t_2))}(b(p))
\end{aligned}
$$

より正しい.

ここから，

- $t = \$$ の時，$a_0(1) \Rightarrow_{M,\sharp(t)} b(1)$
- $t = \mathrm{a}(t_1, t_2)$ の時，

$$
\begin{aligned}
a_0(1) &\Rightarrow_{M,\sharp(t)} a(11) \\
&\Rightarrow^*_{M,\sharp(t)} \mathrm{S}^{\mathrm{size}(t_1)}(b(11)) \\
&\Rightarrow_{M,\sharp(t)} \mathrm{S}^{\mathrm{size}(t_1)}(a(12))
\end{aligned}
$$

$$\Rightarrow^{*}_{M,\sharp(t)} \mathrm{S}^{\mathrm{size}(t_1)+\mathrm{size}(t_2)}(b(12))$$
$$\Rightarrow_{M,\sharp(t)} \mathrm{S}^{\mathrm{size}(t_1)+\mathrm{size}(t_2)}(b(1))$$

より，

$$\forall p \in \mathrm{paths}(\sharp(t))\backslash\{\epsilon\} \,.\, a_0(p) \Rightarrow^{*}_{M,\sharp(t)} \mathrm{S}^{\mathrm{size}(\mathrm{subtree}_{\sharp(t)}(p))-1}(b(p))$$

が成り立つ．よって，

$$\begin{aligned}
\mathrm{height}([\![M]\!](t)) &= \mathrm{height}(\mathrm{nf}(\Rightarrow_{M,\sharp(t)}, a_0(\epsilon))) \\
&= \mathrm{height}(\mathrm{nf}(\Rightarrow_{M,\sharp(t)}, a_0(1))) \\
&= \mathrm{height}(\mathrm{S}^{\mathrm{size}(t)-1}(\mathrm{nf}(\Rightarrow_{M,\sharp(t)}, b(1)))) \\
&= \mathrm{height}(\mathrm{S}^{\mathrm{size}(t)-1}(\mathrm{SZ})) \\
&= \mathrm{size}(t)
\end{aligned}$$

となる． ■

3.3 マクロ木変換器

マクロ木変換器 (MTT) は，TDTT の拡張であり，再帰の際**コンテキストパラメータ** (*context parameter*) と呼ばれるパラメータを持たせ，それを出力の一部として用いることを許容したモデルになっている．

定義 123 (マクロ木変換器 (MTT)). マクロ木変換器は，以下の要素の組 $M = (Q, \Sigma, \Delta, e, R)$ である．

Q　状態記号のランク付きアルファベットで，$Q = \bigcup_{n\geq 1} Q^{(n)}$ ．
Σ　入力記号のランク付きアルファベット．
Δ　出力記号のランク付きアルファベット．
e　初期式で，$e \in \mathrm{RHS}_M(\mathrm{X}_1, \emptyset)$．
R　書き換え規則の集合で，以下を満たす．

$$\begin{aligned}
R \in \mathcal{P}_{fin}(\{&q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \mathrm{y}_1, \ldots, \mathrm{y}_m) \to \eta \\
&\mid q^{(m+1)} \in Q, \sigma^{(n)} \in \Sigma, \eta \in \mathrm{RHS}_M(\mathrm{X}_n, \mathrm{Y}_m)\})
\end{aligned}$$

ただし，$\mathrm{RHS}_M(X, Y) \subseteq \hat{T}_{\Delta\cup Q}(X \cup Y)$ は，以下のように帰納的に定義される集合 U である．

IB1　$y \in Y$ について，$y \in U$．
IB2　$\delta \in \Delta^{(0)}$ について，$\delta \in U$．
IS1　$q^{(m+1)} \in Q, x \in X, \eta_1, \ldots, \eta_m \in U$ について，$q(x, \eta_1, \ldots, \eta_m) \in U$．

IS2　$n \geq 1, \delta^{(n)} \in \Delta, \eta_1, \ldots, \eta_n \in U$ について，$\delta(\eta_1, \ldots, \eta_n) \in U$.

□

MTT は，現在の状態と入力木の部分木の根のラベルを見て適用する規則を選び，その規則の右辺の通りに出力を構築しながら新たに適用する規則を選んでいく．

例 124. MTT $M = (Q, \Sigma, \Delta, e, R)$ を，以下のように定義する．

$$\begin{aligned}
Q &\stackrel{\text{def}}{=} \{q^{(2)}\} \\
\Sigma &\stackrel{\text{def}}{=} \{\mathrm{a}^{(1)}, \mathrm{b}^{(1)}, \$^{(0)}\} \\
\Delta &\stackrel{\text{def}}{=} \Sigma \\
e &\stackrel{\text{def}}{=} q(\mathrm{x}_1, \$) \\
R &\stackrel{\text{def}}{=} \{q(\mathrm{a}(\mathrm{x}_1), \mathrm{y}_1) \to q(\mathrm{x}_1, \mathrm{a}(\mathrm{y}_1)), \\
&\qquad q(\mathrm{b}(\mathrm{x}_1), \mathrm{y}_1) \to q(\mathrm{x}_1, \mathrm{b}(\mathrm{y}_1)), \\
&\qquad q(\$, \mathrm{y}_1) \to \mathrm{y}_1\}
\end{aligned}$$

この M は，与えられた木を上下反転する MTT で，以下のような変換を定義する．

$$[\![M]\!](\mathrm{a}(\mathrm{b}(\mathrm{b}(\mathrm{a}(\mathrm{a}(\$)))))) = \mathrm{a}(\mathrm{a}(\mathrm{b}(\mathrm{b}(\mathrm{a}(\$)))))$$

なお，M の意味論 $[\![M]\!]$ の形式的な定義は，後ほど導入する． □

MTT も TDTT と同じく，全域的決定的性を定義できる．

定義 125 (全域的決定的 MTT). MTT $M = (Q, \Sigma, \Delta, e, R)$ が

$$\forall q^{(m+1)} \in Q, \sigma^{(n)} \in \Sigma \,.\, \exists! q(\sigma(\vec{\mathrm{x}}), \vec{\mathrm{y}}) \to \eta \in R$$

を満たすとき，全域的決定的であるという． □

以降，特に断りのない限り MTT は全域的決定的であるものとする．MTT について，意味論を決定づける簡約システムを導入する．

定義 126 (MTT の簡約システム). MTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X, Y について，簡約システム $(\mathrm{SF}_M(X, Y), \Rightarrow_{M,X,Y})$ を，以下のように定義する．

$$\mathrm{SF}_M(X, Y) \stackrel{\text{def}}{=} \mathrm{RHS}_M(\hat{T}_\Sigma(X), Y)$$

$$\varphi_1 \Rightarrow_{M,X,Y} \varphi_2 \text{ iff } \left(\begin{array}{l} \exists q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \mathrm{y}_1, \ldots, \mathrm{y}_m) \to \eta \in R, \\ \quad t_1, \ldots, t_n \in \hat{T}_\Sigma, \\ \quad \varphi'_1, \ldots, \varphi'_n \in A, \\ \quad \pi \in \mathrm{paths}(\varphi_1), \mathrm{subtree}_{\varphi_1}(\pi) = q(\sigma(t_1, \ldots, t_n), \varphi'_1, \ldots, \varphi'_m) \,. \\ \varphi_2 = \varphi_1[\pi \leftarrow \eta[\mathrm{x}_i \leftarrow t_i]_{i \in [n]}[\mathrm{y}_i \leftarrow \varphi'_i]_{i \in [m]}] \end{array} \right)$$

この時，$\Rightarrow_{M,\emptyset,\emptyset}$ を $\Rightarrow_M$ と表記する． □

この簡約システムは局所合流性を持つ．

補題 127. MTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X, Y について，$\Rightarrow_{M,X,Y}$ は局所合流性を持つ． □

証明. [FV98] の補題 4.10 より． ■

また，この簡約システムは停止性を持つ．

補題 128. MTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X, Y について，$\Rightarrow_{M,X,Y}$ は停止性を持つ． □

証明. [FV98] の補題 4.11 より． ■

以上から，MTT の簡約システムは合流性を持つ．

系 129. MTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X, Y について，$\Rightarrow_{M,X,Y}$ は合流性を持つ． □

証明. 補題 128，補題 127，補題 56 より． ■

MTT の簡約システムは，合流性と停止性を持つため，補題 57 より一意な正規形を持つ．この正規形は，$\Rightarrow_M$ においては必ず出力アルファベットのみからなる木になる．

補題 130. MTT $M = (Q, \Sigma, \Delta, e, R)$ について，以下が成り立つ．

$$\forall \eta \in \mathrm{SF}_M(\emptyset, \emptyset) \,.\, \mathrm{nf}(\Rightarrow_M, \eta) \in \hat{T}_\Delta$$

□

証明. [FV98] の定理 4.15 より． ■

よって，MTT の簡約システムにおける変数を持たない項の正規形は，必ず一意でかつ出力木になる．この時 MTT に対して，入力木を受け取り，初期式にその木を割り当てたものから MTT の簡約システムで得られる木を出力とする，木変換が考えられる．この木変換を，意味論として導入する．

定義 131 (MTT の意味論). MTT $M = (Q, \Sigma, \Delta, e, R)$ について，$[\![M]\!] : \hat{T}_\Sigma \to \hat{T}_\Delta$ を以下のように定義する．

$$[\![M]\!] \stackrel{\mathrm{def}}{=} t \mapsto \mathrm{nf}(\Rightarrow_M, e[\mathrm{x}_1 \leftarrow t])$$

また，木変換のクラス $\{[\![M]\!] \mid M$ は (全域的決定的) MTT$\}$ を MTT と表記する． □

なお，MTT が全域的決定的な場合，この簡約システムを決定的にすることができる．

命題 132 (MTT の決定的な簡約システム). MTT $M = (Q, \Sigma, \Delta, e, R)$ について，簡約システム $(\mathrm{SF}_M(X,Y), \Rightarrow^D_{M,X,Y})$ を，$U \stackrel{\mathrm{def}}{=} \mathrm{SF}_M(X,Y)$ として，以下のように構造的帰納法で定義する．

IB1　$y \in Y$ について，y は既約．

IB2　$\delta \in \Delta^{(0)}$ について，δ は既約．

IS1　以下のように場合分けを行う．

- $q \in Q$，$\sigma \in \Sigma$，$\sigma(\vec{t}) \in \hat{T}_\Sigma(X)$，$q(\sigma(\vec{\mathrm{x}}), \vec{\mathrm{y}}) \to \eta \in R$ について，

$$q(\sigma(\vec{t}), \vec{\eta_y}) \Rightarrow^D_{M,X,Y} \eta[\vec{\mathrm{x}} \leftarrow \vec{t}][\vec{\mathrm{y}} \leftarrow \vec{\eta_y}].$$

- $q \in Q$，$x \in X$，$\eta_1, \ldots, \eta_m \in U$，$i \in [m]$ について，$\eta_1, \ldots, \eta_{i-1}$ が $\Rightarrow^D_{M,X,Y}$ で既約で，$\eta_i \Rightarrow^D_{M,X,Y} \eta'_i$ の時，

$$q(x, \eta_1, \ldots, \eta_i, \ldots, \eta_m) \Rightarrow^D_{M,X,Y} q(x, \eta_1, \ldots, \eta'_i, \ldots, \eta_m).$$

IS2　$n \geq 1$，$\delta^{(n)} \in \Delta$，$\eta_1, \ldots, \eta_n \in U$，$i \in [n]$ について，$\eta_1, \ldots, \eta_{i-1}$ が $\Rightarrow^D_{M,X,Y}$ で既約で，$\eta_i \Rightarrow^D_{M,X,Y} \eta'_i$ の時，

$$\delta(\eta_1, \ldots, \eta_i, \ldots, \eta_n) \Rightarrow^D_{M,X,Y} \delta(\eta_1, \ldots, \eta'_i, \ldots, \eta_n).$$

この時，任意の $\eta \in U$ に対して，$\mathrm{nf}(\Rightarrow_{M,X,Y}, \eta) = \mathrm{nf}(\Rightarrow^D_{M,X,Y}, \eta)$． □

証明. 命題 88 と同様に示せる． ■

コンテキストパラメータを持たない状態のみからなる MTT は，TDTT に対応する．よって，TDTT $\subseteq$ MTT である．

定理 133 ([Eng80]). TDTT $\subseteq$ MTT □

証明. 定理 118 と同様に確かめられる． ■

木変換が MTT で表現できるかを判別する方法のひとつとして，TDTT と同様，樹高性を使う方法がある．MTT は，出力木の高さが入力木の高さに対して指数関数的になるという，樹高性を持つ．

定理 134 (MTT の樹高性 [FV98]). $[\![M]\!] : \hat{T}_\Sigma \to \hat{T}_\Delta \in \mathrm{MTT}$ について，

$$\forall t \in \hat{T}_\Sigma \,.\, \mathrm{height}([\![M]\!](t)) \leq \exp(c_M \cdot \mathrm{height}(t))$$

を満たす $c_M \in \mathbb{N}$ が存在する． □

証明. MTT $M = (Q, \Sigma, \Delta, e, R)$ について，

$$c_1 \stackrel{\mathrm{def}}{=} \mathrm{height}(e)$$

$$c_2 \stackrel{\text{def}}{=} \max\{\text{height}(\eta) \mid q(...) \to \eta \in R\}$$
$$c_M \stackrel{\text{def}}{=} c_1 \cdot c_2$$

とおく．この時，任意の $q^{(m+1)} \in Q$，$t \in \hat{T}_\Sigma$，$h_1, ..., h_m \in \mathbb{N}$ について，$h_t(q, h_1, ..., h_m)$ を，以下のように t に関する構造的帰納法で示す．

IB1　適用不能．

IB2　$\sigma \in \Sigma^{(0)}$，$q(\sigma, ...) \to \eta \in R$ について，

$$h_\sigma(q, h_1, ..., h_m) \stackrel{\text{def}}{=} \text{count}(\eta, h_1, ..., h_m).$$

IS　$n \geq 1$，$\sigma^{(n)} \in \Sigma$，$t_1, ..., t_n \in \hat{T}_\Sigma$，$q(\sigma(...), ...) \to \eta \in R$ について，

$$h_{\sigma(t_1,...,t_n)}(q, h_1, ..., h_m) = \text{count}(\eta[\text{x}_i \leftarrow t_i]_{i\in[n]}, h_1, ..., h_m).$$

ただし，$U_m \stackrel{\text{def}}{=} \text{SF}_M(\emptyset, \text{Y}_m)$，$\eta \in U_m$ について，$\text{count}(\eta, h_1, ..., h_m)$ は以下のように η に関する構造的帰納法で定義する．

IB1　$i \in [m]$ について，$\text{count}(\text{y}_i, h_1, ..., h_m) \stackrel{\text{def}}{=} h_i$.

IB2　$\delta \in \Delta^{(0)}$ について，$\text{count}(\delta, ...) \stackrel{\text{def}}{=} 1$.

IS1　$q^{(k+1)} \in Q$，$t \in \hat{T}_\Sigma$，$\eta_1, ..., \eta_k \in U_m$ について，

$$\text{count}(q(t, \eta_1, ..., \eta_k)) \stackrel{\text{def}}{=} h_t(q, \text{count}(\eta_1), ..., \text{count}(\eta_k)).$$

IS2　$n \geq 1$，$\delta^{(n)} \in \Delta$，$\eta_1, ..., \eta_n \in U_m$ について，

$$\text{count}(\delta(\eta_1, ..., \eta_n)) \stackrel{\text{def}}{=} 1 + \max\{\text{count}(\eta_i) \mid i \in [n]\}.$$

また，$t \in \hat{T}_\Sigma$，$\eta \in U_m$ について，

$$\text{deriv}_\eta(t) \stackrel{\text{def}}{\iff} \forall q \in Q, \pi \in \text{occ}_\eta(q) \,.\, \exists p \in \text{paths}(t) \,.\, \text{subtree}_\eta(\pi 1) = \text{subtree}_t(p)$$

とおく．この時，任意の $t \in \hat{T}_\Sigma$ について，

- $q^{(m+1)} \in Q$，$\eta_1, ..., \eta_m \in U_0$ について，任意の $i \in [m]$ で $\text{deriv}_{\eta_i}(t)$ の時，$\{h_i\}_{i\in[m]} \stackrel{\text{def}}{=} \{\text{count}(\eta_i)\}_{i\in[m]}$ とすると，
$$\text{height}(\text{nf}(\Rightarrow_M, q(t, \eta_1, ..., \eta_m))) \leq h_t(q, h_1, ..., h_m)$$
- $\eta \in U_0$ について，$\text{deriv}_\eta(t)$ の時，
$$\text{height}(\text{nf}(\Rightarrow_M, \eta)) \leq \text{count}(\eta)$$
- $q^{(m+1)} \in Q$，$h_1, ..., h_m \in \mathbb{N}$ について，
$$h_t(q, h_1, ..., h_m) \leq c_2^{\text{height}(t)} + \max\{h_i \mid i \in [m]\}$$

は，t に関する容易な同時帰納法で示せる．ここから，

$$\mathrm{height}(\mathrm{nf}(\Rightarrow_M, e[\mathrm{x}_1 \leftarrow t])) \leq c_1 \cdot c_2^{\mathrm{height}(t)}$$

は，e に関する容易な構造的帰納法で示せる．よって，

$$\begin{aligned}
\mathrm{height}([\![M]\!](t)) &= \mathrm{height}(\mathrm{nf}(\Rightarrow_M, e[\mathrm{x}_1 \leftarrow t])) \\
&\leq c_1 \cdot c_2^{\mathrm{height}(t)} \\
&\leq \exp(c_1 + c_2 \cdot \mathrm{height}(t)) \\
&\leq \exp(c_1 \cdot c_2 \cdot \mathrm{height}(t)) \\
&= \exp(c_M \cdot \mathrm{height}(t))
\end{aligned}$$

となる． ■

なお，$c_M = 1$ になる MTT M として，以下のものがある．

例 135. MTT $M = (Q, \Sigma, \Delta, e, R)$ を以下のように定義する．

$$\begin{aligned}
Q &\stackrel{\mathrm{def}}{=} \{q^{(2)}\} \\
\Sigma &\stackrel{\mathrm{def}}{=} \{\mathrm{a}^{(1)}, \$^{(0)}\} \\
\Delta &\stackrel{\mathrm{def}}{=} \Sigma \\
e &\stackrel{\mathrm{def}}{=} q(\mathrm{x}_1, \$) \\
R &\stackrel{\mathrm{def}}{=} \{q(\mathrm{a}(\mathrm{x}_1), \mathrm{y}_1) \to \mathrm{a}(q(\mathrm{x}_1, q(\mathrm{x}_1, \mathrm{y}_1))), \\
&\qquad q(\$, \mathrm{y}_1) \to \mathrm{a}(\mathrm{y}_1)\}
\end{aligned}$$

□

命題 136. 例 135 の $M = (Q, \Sigma, \Delta, e, R)$ について，以下が成り立つ．

$$\forall t \in \hat{T}_\Sigma \,.\, \mathrm{height}([\![M]\!](t)) = 2^{\mathrm{height}(t)}$$

□

証明.

$$\forall t \in \hat{T}_\Sigma, t' \in \hat{T}_\Delta \,.\, \mathrm{height}(\mathrm{nf}(\Rightarrow_M, q(t, t'))) = 2^{\mathrm{height}(t)} - 1 + \mathrm{height}(t')$$

を，t に関する構造的帰納法で示す．

IB1　適用不能．

IB2

$$\mathrm{height}(\mathrm{nf}(\Rightarrow_M, q(\$, t'))) = \mathrm{height}(\mathrm{a}(t'))$$

$$= 1 + \mathrm{height}(t')$$
$$= 2^{\mathrm{height}(\$)} - 1 + \mathrm{height}(t')$$

より正しい.

IS

$$\begin{aligned}
\mathrm{height}(\mathrm{nf}(\Rightarrow_M, q(\mathrm{a}(t), t'))) &= \mathrm{height}(\mathrm{a}(\mathrm{nf}(\Rightarrow_M, q(t, \mathrm{nf}(\Rightarrow_M, q(t, t'))))))\\
&= 1 + \mathrm{height}(\mathrm{nf}(\Rightarrow_M, q(t, \mathrm{nf}(\Rightarrow_M, q(t, t')))))\\
&= 1 + 2^{\mathrm{height}(t)} - 1 + \mathrm{height}(\mathrm{nf}(\Rightarrow_M, q(t, t'))) \quad (\because \text{i.h.})\\
&= 1 + 2^{\mathrm{height}(t)} - 1 + 2^{\mathrm{height}(t)} - 1 + \mathrm{height}(t') \quad (\because \text{i.h.})\\
&= 2^{\mathrm{height}(t)+1} + 1 - 2 + \mathrm{height}(t')\\
&= 2^{\mathrm{height}(\mathrm{a}(t))} - 1 + \mathrm{height}(t')
\end{aligned}$$

より正しい.

よって,

$$\begin{aligned}
\mathrm{height}([\![M]\!](t)) &= \mathrm{height}(\mathrm{nf}(\Rightarrow_M, q(t, \$)))\\
&= 2^{\mathrm{height}(t)} - 1 + \mathrm{height}(\$)\\
&= 2^{\mathrm{height}(t)}
\end{aligned}$$

となる. ■

マクロ木変換器に対して, 滞在拡張を施した木変換器を, **滞在付きマクロ木変換器** (*stayMTT*) [EM03a] と呼ぶ. 通常マクロ木変換器は, 部分木に対してしか再帰ができないが, 滞在拡張では現在の入力木に対しても再帰を行うことを許す.

定義 137 (滞在付きマクロ木変換器 (stayMTT)). 滞在付きマクロ木変換器は, 以下の要素の組 $M = (Q, \Sigma, \Delta, e, R)$ である.

Q, Σ, Δ, e MTT と同様.
R 以下のようにする.

$$\begin{aligned}
R \in \mathcal{P}_{fin}(\{q(\mathrm{x} = \sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \mathrm{y}_1, \ldots, \mathrm{y}_m) \to \eta\\
\mid q^{(m+1)} \in Q, \sigma^{(n)} \in \Sigma, \eta \in \mathrm{RHS}_M(\mathrm{X}_n \cup \{\mathrm{x}\}, \mathrm{Y}_m)\})
\end{aligned}$$

ただし, RHS_M は MTT と同様に定義する. □

stayMTT でも, MTT と同じく全域的決定的性を定義できる.

定義 138 (全域的決定的 stayMTT). stayMTT $M = (Q, \Sigma, \Delta, e, R)$ が

$$\forall q^{(m+1)} \in Q, \sigma^{(n)} \in \Sigma \,.\, \exists! q(\mathrm{x} = \sigma(\vec{\mathrm{x}}), \vec{\mathrm{y}}) \to \eta \in R$$

を満たす時，全域的決定的であるという． □

以降，特に断りのない限り stayMTT は全域的決定的であるものとする．stayMTT について，意味論を決定づける簡約システムを，MTT と同様に導入する．

定義 139 (stayMTT の簡約システム). stayMTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X, Y について，簡約システム $(\mathrm{SF}_M(X,Y), \Rightarrow_{M,X,Y})$ を，以下のように定義する．

$$\mathrm{SF}_M(X,Y) \stackrel{\mathrm{def}}{=} \mathrm{RHS}_M(\hat{T}_\Sigma(X), Y)$$

$$\varphi_1 \Rightarrow_{M,X,Y} \varphi_2 \text{ iff } \left(\begin{array}{l} \exists q(\mathrm{x} = \sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \mathrm{y}_1, \ldots, \mathrm{y}_m) \to \eta \in R, \\ \quad t_1, \ldots, t_n \in \hat{T}_\Sigma, t = \sigma(t_1, \ldots, t_n), \\ \quad \varphi'_1, \ldots, \varphi'_n \in A, \\ \quad \pi \in \mathrm{paths}(\varphi_1), \mathrm{subtree}_{\varphi_1}(\pi) = q(\sigma(t_1, \ldots, t_n), \varphi'_1, \ldots, \varphi'_m)\ . \\ \varphi_2 = \varphi_1[\pi \leftarrow \eta[\mathrm{x} \leftarrow t][\mathrm{x}_i \leftarrow t_i]_{i \in [n]}[\mathrm{y}_i \leftarrow \varphi'_i]_{i \in [m]}] \end{array} \right)$$

この時，$\Rightarrow_{M,\emptyset,\emptyset}$ を $\Rightarrow_M$ と表記する． □

stayMTT では，ATT と同様簡約システムは停止性を持たない．これは，stayMTT が全域的決定的であっても意味論が関数にならない場合があることを意味する．そのような場合を除くため，ATT と同様に well-defined の概念を導入する．

定義 140 (stayMTT の well-defined 性). stayMTT $M = (Q, \Sigma, \Delta, e, R)$ について，

$$\forall t \in \hat{T}_\Sigma, q^{(m+1)} \in Q\ .\ q(t, \mathrm{y}_1, \ldots, \mathrm{y}_m) \text{ は } \Rightarrow_{M,\emptyset,\mathrm{Y}_m} \text{ で無限簡約可能でない}$$

を満たす時 M は well-defined という． □

stayMTT の well-defined 性の同値条件に，非巡回性がある．

定義 141 (stayMTT の非巡回性). stayMTT $M = (Q, \Sigma, \Delta, e, R)$ について，

$$\exists \eta, \pi \in \mathrm{occ}_\eta(q), \mathrm{subtree}_\eta(\pi 1) = \sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n)\ .\ q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \mathrm{y}_1, \ldots, \mathrm{y}_m) \Rightarrow^+_{M,\mathrm{X}_n,\mathrm{Y}_m} \eta$$

を満たす $\sigma^{(n)} \in \Sigma$，$q^{(m+1)} \in Q$ が存在する時，M は巡回であるという．M が巡回でない時，非巡回であるという． □

ATT と同様に，滞在に関する依存グラフを作成しそのグラフが非巡回である時，stayMTT は非巡回である．ここから ATT と同様の議論で，stayMTT が非巡回である時，簡約システムは停止性を持つ．

補題 142. stayMTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X, Y について，M が非巡回の時，$\Rightarrow_{M,X,Y}$ は停止性を持つ． □

証明. 補題 109 と同様の議論で示せる． ■

この 2 つの条件が同値であることも，ATT と同様に示せる．

定理 143. stayMTT $M = (Q, \Sigma, \Delta, e, R)$ について，以下は 2 条件は同値である．

- M は well-defined.
- M は非巡回.

□

証明. M が well-defined ならば非巡回であることを示す．M が well-defined である時，M は巡回であると仮定する．この時，

- $\eta_0 = q(\sigma(\mathrm{x}_1, \dots, \mathrm{x}_n), \mathrm{y}_1, \dots, \mathrm{y}_m)$
- $\mathrm{subtree}_{\eta_k}(\pi 1) = \sigma(\mathrm{x}_1, \dots, \mathrm{x}_n)$
- $\eta_0 \Rightarrow_{M,\mathrm{X}_n,\mathrm{Y}_m} \cdots \Rightarrow_{M,\mathrm{X}_n,\mathrm{Y}_m} \eta_k$

を満たす $\sigma^{(n)} \in \Sigma$，$q^{(m+1)} \in Q$，$k > 0$，$\eta_0, \dots, \eta_k$，$\pi \in \mathrm{occ}_{\eta_k}(q)$ が存在する．これらについて，

$$\varphi_1 \stackrel{\mathrm{def}}{=} \eta_k[\pi \leftarrow *]$$
$$\varphi_2 \stackrel{\mathrm{def}}{=} \{\mathrm{subtree}_{\eta_k}(\pi(i+1))\}_{i \in [m]}$$

とおく．$k' > k$ について，$k' = c_1 k + c_2$ と書ける $c_1 \in \mathbb{N}$，$0 \leq c_2 < k$ が存在する．この時，$\eta_{k'} \stackrel{\mathrm{def}}{=} \varphi_1^{c_1}(\eta_{c_2}[\mathrm{y}_i \leftarrow \varphi_{2,i}]_{i \in [m]})$ とおくと，$\{\eta_k\}_{k \in \mathbb{N}}$ は $q(\sigma(\mathrm{x}_1, \dots, \mathrm{x}_n), \mathrm{y}_1, \dots, \mathrm{y}_m)$ の無限簡約列になる．$\mathrm{x}_1, \dots, \mathrm{x}_n$ に適当な入力木を割り当てれば，M が well-defined に矛盾することは直ちに分かる．よって，正しい．逆は，補題 142 より明らか．よって，題意は示された． ■

なお, stayMTT について, ATT と同様に, 滞在に関する依存グラフの非巡回性も well-defined 性と同値になる．ここから，stayMTT の well-defined 性が決定可能であることは，滞在に関する依存グラフの非巡回性が決定可能なことから示せる．なお，ATT の際は依存関係としてあり得る全ての通りに対して検査しなければならなかったが，滞在に関する依存グラフの非巡回性は，単に入力記号ごとの依存グラフに対し，巡回があるか判定するだけで良い．これは，全域的決定的であれば**トポロジカルソート** (*topological sort*) [CLRS09] で状態の数に対し線形時間で判定できる．

定理 144 ([EM03a]). stayMTT M について，M が well-defined かは決定可能． □

証明. 定理 111 と同様の議論で示せる． ■

以降，特に断りのない限り stayMTT は well-defined であるものとする．stayMTT の簡約システムは局所合流性を持つ．

補題 145. stayMTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X, Y について，$\Rightarrow_{M,X,Y}$ は局所合流性を持つ．

□

証明. 補題 127 と同様に示せる． ■

また，well-defined 性から stayMTT の簡約システムは停止性を持つため，合流性を持つ．

系 146. stayMTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X, Y について，$\Rightarrow_{M,X,Y}$ は合流性を持つ． □

証明. 補題 142，補題 145，補題 56 より． ■

stayMTT の簡約システムは，合流性と停止性を持つため，補題 57 より一意な正規形を持つ．この正規形は，$\Rightarrow_M$ においては必ず出力アルファベットのみからなる木になる．

補題 147. stayMTT $M = (Q, \Sigma, \Delta, e, R)$ について，以下が成り立つ．

$$\forall \eta \in \mathrm{SF}_M(\emptyset, \emptyset) \,.\, \mathrm{nf}(\Rightarrow_M, \eta) \in \hat{T}_\Delta$$

□

証明. 補題 130 と同様に示せる． ■

よって，stayMTT の簡約システムにおける変数を持たない項の正規形は，必ず一意でかつ出力木になる．この時 stayMTT に対して，入力木を受け取り，初期式にその木を割り当てたものから stayMTT の簡約システムで得られる木を出力とする，木変換が考えられる．この木変換を，意味論として導入する．

定義 148 (stayMTT の意味論). stayMTT $M = (Q, \Sigma, \Delta, e, R)$ について，$[\![M]\!] : \hat{T}_\Sigma \to \hat{T}_\Delta$ を以下のように定義する．

$$[\![M]\!] \stackrel{\mathrm{def}}{=} t \mapsto \mathrm{nf}(\Rightarrow_M, e[\mathrm{x}_1 \leftarrow t])$$

また，木変換のクラス $\{[\![M]\!] \mid M$ は (全域的決定的 well-defined) stayMTT$\}$ を stayMTT と表記する． □

なお，stayMTT が全域的決定的な場合，この簡約システムを決定的にすることができる．

命題 149 (stayMTT の決定的な簡約システム). stayMTT $M = (Q, \Sigma, \Delta, e, R)$ について，簡約システム $(\mathrm{SF}_M(X, Y), \Rightarrow^D_{M,X,Y})$ を，$U \stackrel{\mathrm{def}}{=} \mathrm{SF}_M(X, Y)$ として，以下のように構造的帰納法で定義する．

IB1　$y \in Y$ について，y は既約．
IB2　$\delta \in \Delta^{(0)}$ について，δ は既約．

IS1　以下のように場合分けを行う．

- $q \in Q$，$\sigma^{(n)} \in \Sigma$，$t = \sigma(t_1, \ldots, t_n) \in \hat{T}_\Sigma(X)$，$q(\mathrm{x} = \sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \vec{\mathrm{y}}) \to \eta \in R$ について，

$$q(t, \vec{\eta_y}) \Rightarrow^D_{M,X,Y} \eta[\mathrm{x} \leftarrow t][\mathrm{x}_i \leftarrow t_i]_{i \in [n]}[\vec{\mathrm{y}} \leftarrow \vec{\eta_y}].$$

- $q \in Q$，$x \in X$，$\eta_1, \ldots, \eta_m \in U$，$i \in [m]$ について，$\eta_1, \ldots, \eta_{i-1}$ が $\Rightarrow^D_{M,X,Y}$ で既約で，$\eta_i \Rightarrow^D_{M,X,Y} \eta'_i$ の時，

$$q(x, \eta_1, \ldots, \eta_i, \ldots, \eta_m) \Rightarrow^D_{M,X,Y} q(x, \eta_1, \ldots, \eta'_i, \ldots, \eta_m).$$

IS2　$n \geq 1$，$\delta^{(n)} \in \Delta$，$\eta_1, \ldots, \eta_n \in U$，$i \in [n]$ について，$\eta_1, \ldots, \eta_{i-1}$ が $\Rightarrow^D_{M,X,Y}$ で既約で，$\eta_i \Rightarrow^D_{M,X,Y} \eta'_i$ の時，

$$\delta(\eta_1, \ldots, \eta_i, \ldots, \eta_n) \Rightarrow^D_{M,X,Y} \delta(\eta_1, \ldots, \eta'_i, \ldots, \eta_n).$$

この時，任意の $\eta \in U$ に対して，$\mathrm{nf}(\Rightarrow_{M,X,Y}, \eta) = \mathrm{nf}(\Rightarrow^D_{M,X,Y}, \eta)$． □

証明. 命題 88 と同様に示せる． ■

滞在拡張を使用しない stayMTT は MTT に対応する．よって，MTT $\subseteq$ stayMTT である．

定理 150 ([EM03a]). MTT $\subseteq$ stayMTT □

証明. MTT $M = (Q, \Sigma, \Delta, e, R)$ について，$M' = (Q, \Sigma, \Delta, e, R')$ を以下のように定義する．

$$R' \stackrel{\mathrm{def}}{=} \{q(\mathrm{x} = \sigma(\vec{\mathrm{x}}), \vec{\mathrm{y}}) \to \eta \mid q(\sigma(\vec{\mathrm{x}}), \vec{\mathrm{y}}) \to \eta \in R\}$$

この時，$[\![M]\!] = [\![M']\!]$ は，定理 118 と同様に示せる． ■

3.4　表現力の比較

TDTT，ATT，MTT はこの順に，木変換クラスの真部分集合関係が成り立つことが，知られている．まず，TDTT と ATT におけるクラスの関係を示す．TDTT $\subseteq$ ATT であることは，定理 118 で示した．その逆の関係は，樹高性により成り立たないことが知られている．

補題 151 ([FV98]). ATT $\not\subseteq$ TDTT □

証明. 例 99 の $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，

$$\forall t \in \hat{T}_\Sigma \,.\, \mathrm{height}([\![M]\!](t)) \leq c_M \cdot \mathrm{height}(t)$$

を満たす $c_M \in \mathbb{N}$ が存在すると仮定する．命題 122 から，

$$\forall t \in \hat{T}_\Sigma \,.\, \mathrm{size}(t) \leq c_M \cdot \mathrm{height}(t)$$

である．ここで，$f : \mathbb{N} \to \hat{T}_\Sigma$ を，以下のように数学的帰納法で定義する．

IB　$f(0) \stackrel{\text{def}}{=} \$.$

IS　$f(n+1) \stackrel{\text{def}}{=} \mathrm{a}(f(n), f(n))$.

この時，$\mathrm{height}(f(n)) = n+1$，$\mathrm{size}(f(n)) = 2^{n+1} - 1$ であることは容易に確かめられる．ここから，任意の $n \in \mathbb{N}$ について，

$$2^{n+1} - 1 = \mathrm{size}(f(n)) \leq c_M \cdot \mathrm{height}(f(n)) = c_M \cdot (n+1)$$

が成り立つ．これを整理すると，

$$2^n \leq \frac{c_M}{2}(n+1) + \frac{1}{2} \leq c_M \cdot n + (c_M + 1)$$

となり，補題 11 に矛盾する．よって，そのような c_M は存在しないため，定理 94 より $[\![M]\!] \notin \mathrm{TDTT}$．よって，題意は示された． ■

以上から，TDTT は ATT より，クラスの部分関係において，真に小さい．

定理 152 ([FV98]). $\mathrm{TDTT} \subsetneq \mathrm{ATT}$ □

証明. 定理 118，補題 151 より． ■

次に，ATT と MTT の関係を示す．全域的決定的な場合，非巡回性を利用して，ATT を意味論同値な MTT に変形できる．この変形は，ATT における合成属性を状態に，継承属性をコンテキストパラメータに変換することで，擬似的に上下の走査を模倣するというものである．しかし，単純にはその変形は停止しない場合がある．それは，ATT の非巡回性が構文的制約としては現れないことに起因する．そこで定理 110 から，ATT において同じノードで同じ属性を 2 回以上続けて参照することがないことを利用し，二回目の参照を強制的に適当な出力木に置換する．これによって，各規則の変形を強制的に合成属性の数に比例したステップ数で停止させる．これにより，$\mathrm{ATT} \subseteq \mathrm{MTT}$ が示された．

補題 153 ([FV98]). $\mathrm{ATT} \subseteq \mathrm{MTT}$ □

証明. ATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，$m \stackrel{\text{def}}{=} |A_{\mathrm{inh}}|$ とおく．また，$A_{\mathrm{inh}} = \{b_1, \ldots, b_m\}$ のように継承属性を $[m]$ で整列させ，$\delta_0 \in \Delta^{(0)}$ を適当におく．この時，MTT $M' = (Q', \Sigma, \Delta, e', R')$ を以下のように定義する．

$$Q \stackrel{\text{def}}{=} \{a^{(m+1)} \mid a \in A_{\mathrm{syn}}\}$$
$$e' \stackrel{\text{def}}{=} \mathrm{conv}_{\sharp,1}(\eta_0, \emptyset)$$
$$R' \stackrel{\text{def}}{=} \{a(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \mathrm{y}_1, \ldots, \mathrm{y}_m) \to \mathrm{conv}_{\sigma,n}(\eta, \emptyset) \mid \sigma^{(n)} \in \Sigma, a(\mathrm{w}) \xrightarrow{\sigma} \eta \in R\}$$

ただし，η_0 は，$a_0(\mathrm{w}) \overset{\sharp}{\to} \eta_0 \in R$ を満たすものとしておき，$\mathrm{conv}_{\sigma,n} : \mathrm{RHS}_M(\{\mathrm{w}\},[n]) \times \mathcal{P}(A_{\mathrm{syn}} \times [n]) \to \mathrm{RHS}_{M'}(\mathrm{X}_n, \mathrm{Y}_m)$ は，$U \overset{\mathrm{def}}{=} \mathrm{RHS}_M(\{\mathrm{w}\},[n])$ として，以下のように $\eta \in U$ に関する構造的帰納法で定義する．

IB1　$a \in A_{\mathrm{syn}}$，$i \in [n]$，$C \in \mathcal{P}(A_{\mathrm{syn}} \times [n])$ について，$C' = C \cup \{(a,i)\}$ として，

$$\mathrm{conv}_{\sigma,n}(a(\mathrm{w}i), C) \overset{\mathrm{def}}{=} \begin{cases} \delta_0 & ((a,i) \in C) \\ a(\mathrm{x}_i, \mathit{iconv}_{\sigma,n}(i,1,C'), \ldots, \mathit{iconv}_{\sigma,n}(i,m,C')) & (\text{otherwise}) \end{cases}.$$

IB2　$j \in [m]$，$C \in \mathcal{P}(A_{\mathrm{syn}} \times [n])$ について，$\mathrm{conv}_{\sigma,n}(b_j(\mathrm{w}), C) \overset{\mathrm{def}}{=} \mathrm{y}_j$．

IB3　$\delta \in \Delta^{(0)}$，$C \in \mathcal{P}(A_{\mathrm{syn}} \times [n])$ について，$\mathrm{conv}_{\sigma,n}(\delta, C) \overset{\mathrm{def}}{=} \delta$．

IS　$k \geq 1$，$\delta^{(k)} \in \Delta$，$\eta_1, \ldots, \eta_k \in U$，$C \in \mathcal{P}(A_{\mathrm{syn}} \times [n])$ について，

$$\mathrm{conv}_{\sigma,n}(\delta(\eta_1, \ldots, \eta_k), C) \overset{\mathrm{def}}{=} \delta(\mathrm{conv}_{\sigma,n}(\eta_1, C), \ldots, \mathrm{conv}_{\sigma,n}(\eta_k, C)).$$

また，

$$\mathrm{iconv}_{\sigma,n}(i,j,C) \overset{\mathrm{def}}{=} \mathrm{conv}_{\sigma,n}(\eta, C) \qquad (b_j(\mathrm{w}i) \overset{\sigma}{\to} \eta \in R)$$

と定義する．この時，$[\![M]\!] = [\![M']\!]$ であることは，[FV98] の補題 6.1 から分かる． ■

また，樹高性により逆の関係は成り立たないことも知られている．

補題 154 ([FV98]). $\mathrm{MTT} \not\subseteq \mathrm{ATT}$ □

証明. 例 135 の $M = (Q, \Sigma, \Delta, e, R)$ について，

$$\forall t \in \hat{T}_\Sigma \,.\, \mathrm{height}([\![M]\!](t)) \leq c_M \cdot \mathrm{size}(t)$$

を満たす $c_M \in \mathbb{N}$ が存在すると仮定する．命題 136，命題 37 から，

$$\forall t \in \hat{T}_\Sigma \,.\, 2^{\mathrm{height}(t)} \leq c_M \cdot \mathrm{height}(t)$$

である．ここで，$f : \mathbb{N} \to \hat{T}_\Sigma$ を，以下のように数学的帰納法で定義する．

IB　$f(0) \overset{\mathrm{def}}{=} \$$.

IS　$f(n+1) \overset{\mathrm{def}}{=} \mathrm{a}(f(n))$.

この時，$\mathrm{height}(f(n)) = n+1$ であることは容易に確かめられる．ここから，任意の $n \in \mathbb{N}$ について，

$$2^{n+1} = 2^{\mathrm{height}(f(n))} \leq c_M \cdot \mathrm{height}(f(n)) = c_M \cdot (n+1)$$

が成り立つ．これは整理すると，

$$2^n \leq \frac{c_M}{2}(n+1) \leq c_M \cdot n + c_M$$

となるが，補題 11 に矛盾する．よって，そのような c_M は存在しないため，定理 121 より $[\![M]\!] \notin \mathrm{ATT}$．よって，題意は示された． ■

以上から，ATT は MTT より，クラスの部分関係において真に小さい．

定理 155 ([FV98]). $\mathrm{ATT} \subsetneq \mathrm{MTT}$ □

証明. 補題 153，補題 154 より． ■

TDTT は MTT より，クラスの部分関係において真に小さいことも，推移律からただちに導かれる．

系 156. $\mathrm{TDTT} \subsetneq \mathrm{MTT}$ □

証明. 定理 152，定理 155 より． ■

また，stayMTT と MTT のクラスは一致することが知られている．これは，滞在付きの拡張が除去できることを示している．MTT が stayMTT に含まれていることは，定理 150 で示した．逆の関係も成り立つことが示せる．これは，定理 143 を利用して，stayMTT において滞在部分を入力木によらず簡約することができることに起因する．この性質により，滞在部分を除去した規則を作ることで，stayMTT は意味論同値な MTT に変形できる．

補題 157 ([EM03a]). $\mathrm{stayMTT} \subseteq \mathrm{MTT}$ □

証明. stayMTT $M = (Q, \Sigma, \Delta, e, R)$ について，$\delta_0 \in \Delta^{(0)}$ を適当におく．この時，MTT $M' = (Q, \Sigma, \Delta, e, R')$ を以下のように定義する．

$$R' \stackrel{\mathrm{def}}{=} \{q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \vec{\mathrm{y}}) \to \mathrm{conv}_\sigma(\eta, \{q\}) \mid q(\mathrm{x} = \sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \vec{\mathrm{y}}) \in R\}$$

ただし，$\mathrm{conv}_\sigma : \mathrm{RHS}_M(\{\mathrm{x}\} \cup \mathrm{X}_n, \mathrm{Y}_m) \times \mathcal{P}(Q) \to \mathrm{RHS}_{M'}(\mathrm{X}_n, \mathrm{Y}_m)$ は，$U \stackrel{\mathrm{def}}{=} \mathrm{RHS}_M(\{\mathrm{x}\} \cup \mathrm{X}_n, \mathrm{Y}_m)$ として，以下のように $\eta \in U$ に関する構造的帰納法で定義する．

IB1　$j \in [m]$，$C \in \mathcal{P}(Q)$ について，$\mathrm{conv}_\sigma(\mathrm{y}_j, C) \stackrel{\mathrm{def}}{=} \mathrm{y}_j$．

IB2　$\delta \in \Delta^{(0)}$，$C \in \mathcal{P}(Q)$ について，$\mathrm{conv}_\sigma(\delta, C) \stackrel{\mathrm{def}}{=} \delta$．

IS1　$q^{(k+1)} \in Q$，$x \in \{\mathrm{x}\} \cup \mathrm{X}_n$，$\eta_1, \ldots, \eta_k \in U$，$C \in \mathcal{P}(Q)$ について，

$$\mathrm{conv}_\sigma(q(x, \eta_1, \ldots, \eta_k), C) \stackrel{\mathrm{def}}{=} \begin{cases} q(x, \mathrm{conv}_\sigma(\eta_1, C), \ldots, \mathrm{conv}_\sigma(\eta_k, C)) & (x \in \mathrm{X}_n) \\ \delta_0 & (q \in C) \\ \mathrm{conv}_\sigma(\eta[\mathrm{y}_i \leftarrow \eta_i]_{i \in [k]}, C \cup \{q\}) & \\ & (q(\mathrm{x} = \sigma(\ldots), \mathrm{y}_1, \ldots, \mathrm{y}_k) \to \eta \in R) \end{cases}.$$

IS2　$n \geq 1$，$\delta^{(k)} \in \Delta$，$\eta_1, \ldots, \eta_k \in U$，$C \in \mathcal{P}(Q)$ について，

$$\mathrm{conv}_\sigma(\delta(\eta_1, \ldots, \eta_k), C) \stackrel{\mathrm{def}}{=} \delta(\mathrm{conv}_\sigma(\eta_1, C), \ldots, \mathrm{conv}_\sigma(\eta_k, C)).$$

この時，$[\![M']\!] = [\![M]\!]$ は，[EM03a] の補題 27 から分かる． ■

以上から，stayMTT と MTT はクラスが一致する．

定理 158 ([EM03a]). stayMTT = MTT □

証明. 定理 150，補題 157 より． ■

また，ATT は stayMTT よりクラスの部分関係において真に小さいことも，推移律よりただちに導かれる．

系 159. ATT $\subsetneq$ stayMTT □

証明. 定理 155，定理 158 より． ■

また，RTL は，MTT の逆像において閉じていることが知られている．

定理 160 ([EV85]). $[\![M]\!] : \hat{T}_\Sigma \to \hat{T}_\Delta \in \mathrm{MTT}$ について，以下が成り立つ．

$$\forall L \in \mathcal{P}(\hat{T}_\Delta) \cap \mathrm{RTL} \,.\, [\![M]\!]^{-1}[L] \in \mathrm{RTL}$$

□

証明. [EV85] の定理 7.4 より． ■

MTT に含まれる TDTT，ATT，stayMTT も同様である．しかし，CFTL は逆像によって閉じていない．これは，TDTT が CFTL の逆像で CFTL の積を表現でき，CFTL が積で閉じていないことに由来する．

定理 161. 以下を満たす $[\![M]\!] : \hat{T}_\Sigma \to \hat{T}_\Delta \in \mathrm{TDTT}$ が存在する．

$$\exists L \in \mathcal{P}(\hat{T}_\Delta) \cap \mathrm{CFTL} \,.\, [\![M]\!]^{-1}[L] \notin \mathrm{CFTL}$$

□

証明. 命題 77 の文脈自由木文法 $G_1 = (N_1, \Sigma, S_1, R_1)$，$G_2 = (N_2, \Sigma, S_2, R_2)$ について，文脈自由木文法 $G = (N, \Sigma, S, R)$ を，以下のように定義する．

$$N \stackrel{\mathrm{def}}{=} \{S^{(0)}\} \uplus N_1 \uplus N_2$$
$$R \stackrel{\mathrm{def}}{=} \{S \to \mathrm{S}(S_1, S_2)\} \cup R_1 \cup R_2$$

この時，$[\![G]\!] = \{\mathrm{S}(t_1, t_2) \mid t_1 \in [\![G_1]\!], t_2 \in [\![G_2]\!]\}$ は容易に確かめられる．また，$[\![G]\!] \in \mathrm{CFTL}$ である．ところで，例 92 の M について，命題 93，命題 77 より，

$$[\![M]\!]^{-1}[[\![G]\!]] = [\![G_1]\!] \cap [\![G_2]\!] \notin \mathrm{CFTL}$$

である．よって，題意は示された． ■

TDTT において，CFTL の逆像が CFTL になるクラスとして，木から文字列への変換がある．

定理 162. TDTT $M = (Q, \Sigma, \Delta, e, R)$ について，Δ が単項ならば，以下が成り立つ．

$$\forall L \in \mathcal{P}(\hat{T}_\Delta) \cap \mathrm{CFTL} \,.\, [\![M]\!]^{-1}[L] \in \mathrm{CFTL}$$

□

証明. $L \in \mathcal{P}(\hat{T}_\Delta) \cap \mathrm{CFTL}$ について，$[\![M_L]\!] = L$ を満たす PDTA $M_L = (Q_L, \Delta, \Gamma, q_{L,0}, u_0, R_L)$ が存在する．この時，PDTA $M'_L = (Q'_L, \Sigma, \Gamma, q'_0, u_0, R'_L)$ を以下のように定義する．

$$\begin{aligned}
Q'_L &\stackrel{\mathrm{def}}{=} (Q \times \Sigma_+ \times Q_L) \cup Q_{R,e} \cup \{q_{ac}\} \\
q'_0 &\stackrel{\mathrm{def}}{=} \langle \sharp, 1, q_{L,0}, e \rangle \\
R'_L &\stackrel{\mathrm{def}}{=} \{\langle \sigma, i, q_L, \delta(\eta)\rangle(\mathrm{x}, \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \to \langle \sigma, i, q'_L, \eta\rangle(\mathrm{x}, u) \\
&\qquad \mid q_L(\delta(\mathrm{x}_1), \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \to \delta(q'_L(\mathrm{x}_1, u)) \in R_L, \langle \sigma, i, q_L, \delta(\eta)\rangle \in Q_{R,e}\} \\
&\cup \{\langle \sigma, i, q_L, \delta\rangle(\mathrm{x}, \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \to q_{ac}(\mathrm{x}, u_0) \\
&\qquad \mid q_L(\delta, \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \to \delta \in R_L, \langle \sigma, i, q_L, \delta\rangle \in Q_{R,e}\} \\
&\cup \{\langle \sigma, i, q_L, \eta\rangle(\mathrm{x}, \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \to \langle \sigma, q'_L, \eta\rangle(\mathrm{x}, u) \\
&\qquad \mid q_L(\mathrm{x}, \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \to q'_L(\mathrm{x}, u) \in R_L, \langle \sigma, i, q_L, \eta\rangle \in Q_{R,e}\} \\
&\cup \{\langle \sigma, i, q_L, q(\mathrm{x}_i)\rangle(\mathrm{x}, \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \to \langle q, \sigma, q_L\rangle(\mathrm{x}, \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \\
&\qquad \mid \langle \sigma, i, q_L, q(\mathrm{x}_i)\rangle \in Q_{R,e}, \gamma^{(m)} \in \Gamma\} \\
&\cup \{\langle \sigma, i, q_L, q(\mathrm{x}_j)\rangle(\mathrm{x}, \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \to q_{ac}(\mathrm{x}, u_0) \\
&\qquad \mid \langle \sigma, i, q_L, q(\mathrm{x}_j)\rangle \in Q_{R,e}, i \neq j, \gamma^{(m)} \in \Gamma\} \\
&\cup \{\langle q, \sharp, q_L\rangle(\mathrm{x}, \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \to \langle \sigma, i, q_L, \eta\rangle(\mathrm{x}, \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \\
&\qquad \mid q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n)) \to \eta \in R, i \in [n], \gamma^{(m)} \in \Gamma\} \\
&\cup \{\langle q, \sigma, q_L\rangle(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \to \\
&\qquad\quad \sigma(\langle \sigma', 1, q_L, \eta\rangle(\mathrm{x}_1, \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)), \ldots, \langle \sigma', n, q_L, \eta\rangle(\mathrm{x}_n, \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m))) \\
&\qquad \mid q(\sigma'(\ldots)) \to \eta \in R, \gamma^{(m)} \in \Gamma\} \\
&\cup \{q_{ac}(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \gamma(\mathrm{z}_1, \ldots, \mathrm{z}_m)) \to \sigma(q_{ac}(\mathrm{x}_1, u_0), \ldots, q_{ac}(\mathrm{x}_n, u_0)) \\
&\qquad \mid \sigma^{(n)} \in \Sigma, \gamma^{(m)} \in \Gamma\}
\end{aligned}$$

ただし，

$$\begin{aligned}
\Sigma_+ &\stackrel{\mathrm{def}}{=} \Sigma \uplus \{\sharp^{(1)}\} \\
Q_{R,e} &\stackrel{\mathrm{def}}{=} Q_{rhs}(\{\eta \mid q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n)) \to \eta \in R\} \cup \{e\}) \\
Q_{rhs}(U) &\stackrel{\mathrm{def}}{=} \{\langle \sigma, q_L, \eta\rangle \\
&\qquad \mid \sigma^{(n)} \in \Sigma_+, i \in [n], q_L \in Q_L, \eta' \in U, \pi \in \mathrm{paths}(\eta'), \eta = \mathrm{subtree}_{\eta'}(\pi)\}
\end{aligned}$$

とおく．この時，$[\![M'_L]\!] = [\![M]\!]^{-1}[L]$ であることは，容易に確認できる． ■

しかし，ATT では例 119 について定理 161 と同様に考えると，木から文字列の変換でも CFTL の逆像は CFTL では閉じていない．また，命題 97 から，定理 162 の拡張として以下の問題が考えられる．

予想 163. 線形 TDTT $M = (Q, \Sigma, \Delta, e, R)$ について，以下が成り立つ．

$$\forall L \in \mathcal{P}(\hat{T}_\Delta) \cap \mathrm{CFTL} \,.\, [\![M]\!]^{-1}[L] \in \mathrm{CFTL}$$

□

しかし，この問題は未解決である．

第 4 章　スタック木変換器

中野は，ATT に対して 5 つの操作を用いて拡張した**スタック属性付き木変換器** (*stack-attributed tree transducer*) を提唱した [Nak09]．また，スタック属性付き木変換器が定める木変換クラスは，属性付き木変換器による木変換クラスより真に広いことを示した．この章では，中野が導入したスタック属性付き木変換器について述べ，さらに本研究で新たに導入した，スタック属性付き木変換器の拡張を TDTT，MTT に適用した**スタックトップダウン木変換器**，**スタックマクロ木変換器**についても述べる．これらを総じてスタック木変換器と呼ぶ．

スタック木変換器は，木変換器に対し，

- スタックを空にする
- スタックに木を積む
- スタックの先頭から木を取り出す
- スタックの先頭を除く
- スタックの参照エラーを起こす

という 5 つの原始的な操作を組み込む．この操作は，スタックシステムとして定義 164，定義 165 で形式的に導入している．このスタックシステムによって，トップダウン木変換器，属性付き木変換器，マクロ木変換器の規則の右辺を拡張した木変換器が，スタックトップダウン木変換器，スタック属性付き木変換器，スタックマクロ木変換器であり，定義 188，定義 213，定義 254 でそれぞれ導入している．中野は，スタック属性付き木変換器を導入した際，属性付き木変換器と同じように簡約システムに対する停止性が一般に保証されないことに触れ，停止性が保証されるような意味論的制約として well-defined 性を導入した [Nak09]．これは，簡約が最外簡約で行われる時，スタックの要素 1 つ 1 つに着目した簡約が停止するという制約であり，この時必ず停止性が保証される．本研究では，スタックシステムの最外戦略以外での簡約について議論した上で，中野が導入した well-defined 性と同値になる，非螺旋性条件を定義 227 で導入し，同値であることの正当性を定理 240 で示した．非螺旋性は属性付き木変換器の非巡回性と対応する条件ではあるが，単純な拡張ではない．これはスタック属性付き木変換器に対する非巡回性が別に定義できることから分かる．スタック属性付き木変換器の非螺旋性と非巡回性の関係については，第 5 章で議論する．スタック属性付き木変換器の簡約システムは，最外戦略に限定しないと well-defined 性を保証できないが，スタックトップダウン木変換器，スタックマクロ木変換器では戦略を限定しなくても問題が起きない．この違いは，表現力について大きな影響を及ぼすと本研究では予想している．この予想に対する考察も第 5 章で行う．

4.1 スタックシステムとスタック評価器

スタック木変換器の定義に入る前に，その定義に必要な**スタックシステム** (*stack system*) を導入する．

定義 164 (スタックシステム). $\mathrm{SS} = \{\mathrm{stk}, \mathrm{elm}\}$ として，SS-型付きアルファベット STK を，以下のように定義する．

$$\begin{aligned}\mathrm{STK} \stackrel{\mathrm{def}}{=} \{ \quad & \mathrm{Empty} : () \twoheadrightarrow \mathrm{stk}, \\ & \mathrm{Cons} : (\mathrm{elm}, \mathrm{stk}) \twoheadrightarrow \mathrm{stk}, \\ & \mathrm{Head} : \mathrm{stk} \twoheadrightarrow \mathrm{elm}, \\ & \mathrm{Tail} : \mathrm{stk} \twoheadrightarrow \mathrm{stk}, \\ & \bot : () \twoheadrightarrow \mathrm{elm}\}\end{aligned}$$

SS-型付きアルファベット Σ，SS-型付き集合 X について，スタックシステム $\mathcal{S}_\Sigma(X)$ を $T_{\mathrm{STK}\cup\Sigma}(X)$ で定義する．また，$\hat{\mathcal{S}}_\Sigma(X) \stackrel{\mathrm{def}}{=} \hat{T}_{\mathrm{STK}\cup\Sigma}(X)$ と定義する．

ランク付きアルファベット Σ に対して，$\dot{\Sigma}$ を集合 Σ と $\mathrm{arity}_{\dot{\Sigma}} = \sigma^{(n)} \mapsto ((\underbrace{\mathrm{elm}, \ldots, \mathrm{elm}}_{n\text{ 項}}), \mathrm{elm})$ による型付きアルファベットと定義する．また，$\Sigma \cup \{\bot^{(0)}\}$ を $\Sigma_\bot$ と表記する．そして，$\ddot{\Sigma}$ を集合 Σ と $\mathrm{arity}_{\ddot{\Sigma}} = \sigma^{(n+1)} \mapsto ((\mathrm{elm}, \underbrace{\mathrm{stk}, \ldots, \mathrm{stk}}_{n\text{ 項}}), \mathrm{stk})$ による型付きアルファベットと定義する．

□

スタックシステムは，スタック木変換器において，ランク付き木に代わる構造として用いられる．スタックシステムは**要素型** elm と**スタック型** stk の 2 つの型を持ち，5 つの原始的な操作を含むことができる．それぞれ以下の操作を表す．

$\bot$　　空のスタックへの参照結果を表す．
Empty　空のスタックを表す．
$\mathrm{Cons}(e, \eta)$　要素 e をスタック η に付け足したスタックを表す．
$\mathrm{Head}(\eta)$　スタック η の先頭の要素への参照を表す．
$\mathrm{Tail}(\eta)$　スタック η から先頭の要素を除去したスタックを表す．

それぞれのスタック操作は，以下のように形式的に定義される．

定義 165 (スタックシステム評価器). 関数の族 $\mathrm{stkeval} : \prod_{k\in\mathrm{SS}} \mathcal{S}_\Sigma(X)_k \to \mathcal{S}_\Sigma(X)_k$ を，以下のように構造的帰納法で定義する．

IB1　$k \in \mathrm{SS}$，$x \in X_k$ について，$\mathrm{stkeval}_k(x) \stackrel{\mathrm{def}}{=} x$

IB2　$k \in \mathrm{SS}$，$\sigma : () \to_{\Sigma \cup \{\bot, \mathrm{Empty}\}} k$ について，$\mathrm{stkeval}_k(\sigma) \stackrel{\mathrm{def}}{=} \sigma$

IS　以下のように場合分けを行う．

- 任意の $n \geq 1$，$k_1, \ldots, k_n, k \in \mathrm{SS}$，$\sigma : (k_1, \ldots, k_n) \to_{\Sigma \cup \{\mathrm{Cons}\}} k$，$t_1 \in \mathcal{S}_\Sigma(X)_{k_1}, \ldots, t_n \in \mathcal{S}_\Sigma(X)_{k_n}$ について，

$$\mathrm{stkeval}_k(\sigma(t_1, \ldots, t_n)) \stackrel{\mathrm{def}}{=} \sigma(\mathrm{stkeval}_{k_1}(t_1), \ldots, \mathrm{stkeval}_{k_n}(t_n))$$

- 任意の $t \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}$ について，$t' \stackrel{\mathrm{def}}{=} \mathrm{stkeval}_{\mathrm{stk}}(t)$ とした時，

$$\mathrm{stkeval}_{\mathrm{elm}}(\mathrm{Head}(t)) \stackrel{\mathrm{def}}{=} \begin{cases} \eta & (t' = \mathrm{Cons}(\eta, \ldots)) \\ \bot & (t' = \mathrm{Empty}) \\ \mathrm{Head}(t') & (\text{otherwise}) \end{cases}$$

- 任意の $t \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}$ について，$t' \stackrel{\mathrm{def}}{=} \mathrm{stkeval}_{\mathrm{stk}}(t)$ とした時，

$$\mathrm{stkeval}_{\mathrm{stk}}(\mathrm{Tail}(t)) \stackrel{\mathrm{def}}{=} \begin{cases} \eta & (t' = \mathrm{Cons}(\ldots, \eta)) \\ \mathrm{Empty} & (t' = \mathrm{Empty}) \\ \mathrm{Tail}(t') & (\text{otherwise}) \end{cases}$$

この時，$k \in \mathrm{SS}$，$\eta \in \mathcal{S}_\Sigma(X)_k$ について，$\mathrm{stkeval}_k(\eta)$ を η の標準形と呼ぶ． □

スタックシステムにおける**標準形**とは，スタック評価による正規形のことを指す．

例 166. $\Sigma = \{\mathrm{A}^{(2)}, \mathrm{B}^{(0)}\}$ について，$t \in \hat{\mathcal{S}}_\Sigma$ を以下のように定義する．

$$t \stackrel{\mathrm{def}}{=} \mathrm{Cons}(\mathrm{A}(\mathrm{Head}(\mathrm{Empty}), \mathrm{Head}(\mathrm{Cons}(\mathrm{B}, \mathrm{Empty}))), \mathrm{Tail}^2(\mathrm{Cons}(\mathrm{B}, \mathrm{Empty})))$$

この時，

$$\mathrm{stkeval}_{\mathrm{stk}}(t) = \mathrm{Cons}(\mathrm{A}(\bot, \mathrm{B}), \mathrm{Empty})$$

となる． □

一般に標準形は，以下の形に制限される．

定理 167 (スタックシステムの標準形). SS-型付きアルファベット Σ，SS-型付き集合 X について，集合の族 $\{A_k\}_{k \in \mathrm{SS}}$ を，以下のように帰納的に定義する．

IB1　$\mathrm{Empty} \in A_{\mathrm{stk}}$

IB2　$m \in \mathbb{N}$，$x \in X_{\mathrm{stk}} \cup (() \to_\Sigma \mathrm{stk})$ について，$\mathrm{Tail}^m(x) \in A_{\mathrm{stk}}$

IB3　$m \in \mathbb{N}$，$x \in X_{\mathrm{stk}} \cup (() \to_\Sigma \mathrm{stk})$ について，$\mathrm{Head}(\mathrm{Tail}^m(x)) \in A_{\mathrm{elm}}$

IB4　$x \in X_{\mathrm{elm}} \cup (() \to_\Sigma \mathrm{elm}) \cup \{\bot\}$ について，$x \in A_{\mathrm{elm}}$

IS1　$\eta_1 \in A_{\mathrm{elm}}$，$\eta_2 \in A_{\mathrm{stk}}$ について，$\mathrm{Cons}(\eta_1, \eta_2) \in A_{\mathrm{stk}}$

IS2　$n \geq 1$，$m \in \mathbb{N}$，$k_1, \ldots, k_n \in \mathrm{SS}$，$\sigma : (k_1, \ldots, k_n) \to_\Sigma \mathrm{stk}$，$\eta_1 \in A_{k_1}, \ldots, \eta_n \in A_{k_n}$ について，$\mathrm{Tail}^m(\sigma(\eta_1, \ldots, \eta_n)) \in A_{\mathrm{stk}}$

IS3　$n \geq 1$，$m \in \mathbb{N}$，$k_1, \ldots, k_n \in \mathrm{SS}$，$\sigma : (k_1, \ldots, k_n) \twoheadrightarrow_\Sigma \mathrm{stk}$，$\eta_1 \in A_{k_1}, \ldots, \eta_n \in A_{k_n}$ について，$\mathrm{Head}(\mathrm{Tail}^m(\sigma(\eta_1, \ldots, \eta_n))) \in A_{\mathrm{elm}}$

IS4　$n \geq 1$，$k_1, \ldots, k_n \in \mathrm{SS}$，$\sigma : (k_1, \ldots, k_n) \twoheadrightarrow_\Sigma \mathrm{elm}$，$\eta_1 \in A_{k_1}, \ldots, \eta_n \in A_{k_n}$ について，$\sigma(\eta_1, \ldots, \eta_n) \in A_{\mathrm{elm}}$

この時，任意の $k \in \mathrm{SS}$ について，$A_k = \{\mathrm{stkeval}_k(\eta) \mid \eta \in \mathcal{S}_\Sigma(X)_k\}$　□

証明. まず，任意の $k \in \mathrm{SS}$ について，$\{\mathrm{stkeval}_k(\eta) \mid \eta \in \mathcal{S}_\Sigma(X)_k\} \subseteq A_k$，つまり，

$$\forall k \in \mathrm{SS}, \eta \in \mathcal{S}_\Sigma(X)_k \,.\, \mathrm{stkeval}_k(\eta) \in A_k$$

を η に関する構造的帰納法で示す.

IB1　以下のように場合分けを行う.

- $x \in X_{\mathrm{elm}}$ について，$\mathrm{stkeval}_{\mathrm{elm}}(x) = x \in A_{\mathrm{elm}}$ より正しい.
- $x \in X_{\mathrm{stk}}$ について，$\mathrm{stkeval}_{\mathrm{stk}}(x) = x = \mathrm{Tail}^0(x) \in A_{\mathrm{stk}}$ より正しい.

IB2　以下のように場合分けを行う.

- $\sigma : () \twoheadrightarrow_{\Sigma \cup \{\bot\}} \mathrm{elm}$ について，$\mathrm{stkeval}_{\mathrm{elm}}(\sigma) = \sigma \in A_{\mathrm{elm}}$ より正しい.
- $\sigma : () \twoheadrightarrow_\Sigma \mathrm{stk}$ について，$\mathrm{stkeval}_{\mathrm{stk}}(\sigma) = \sigma = \mathrm{Tail}^0(\sigma) \in A_{\mathrm{stk}}$ より正しい.

IS　以下のように場合分けを行う.

- 任意の $n \geq 1$，$k_1, \ldots, k_n \in \mathrm{SS}$，$\sigma : (k_1, \ldots, k_n) \twoheadrightarrow_\Sigma \mathrm{elm}$，
  $t_1 \in \mathcal{S}_\Sigma(X)_{k_1}, \ldots, t_n \in \mathcal{S}_\Sigma(X)_{k_n}$ について，

$$\begin{aligned}\mathrm{stkeval}_{\mathrm{elm}}(\sigma(t_1, \ldots, t_n)) &= \sigma(\mathrm{stkeval}_{k_1}(t_1), \ldots, \mathrm{stkeval}_{k_n}(t_n)) \\ &\in A_{\mathrm{elm}} \\ &(\because \text{i.h. より } \forall i \in [n] \,.\, \mathrm{stkeval}_{k_i}(t_i) \in A_{k_i})\end{aligned}$$

  より正しい.

- 任意の $n \geq 1$，$k_1, \ldots, k_n \in \mathrm{SS}$，$\sigma : (k_1, \ldots, k_n) \twoheadrightarrow_\Sigma \mathrm{stk}$，
  $t_1 \in \mathcal{S}_\Sigma(X)_{k_1}, \ldots, t_n \in \mathcal{S}_\Sigma(X)_{k_n}$ について，

$$\begin{aligned}\mathrm{stkeval}_{\mathrm{stk}}(\sigma(t_1, \ldots, t_n)) &= \sigma(\mathrm{stkeval}_{k_1}(t_1), \ldots, \mathrm{stkeval}_{k_n}(t_n)) \\ &= \mathrm{Tail}^0(\sigma(\mathrm{stkeval}_{k_1}(t_1), \ldots, \mathrm{stkeval}_{k_n}(t_n))) \\ &\in A_{\mathrm{stk}} \\ &(\because \text{i.h. より } \forall i \in [n] \,.\, \mathrm{stkeval}_{k_i}(t_i) \in A_{k_i})\end{aligned}$$

  より正しい.

- 任意の $t \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}$ について，$\mathrm{stkeval}_{\mathrm{stk}}(t) = \mathrm{Cons}(t', \ldots)$ の時，

$$\mathrm{stkeval}_{\mathrm{elm}}(\mathrm{Head}(t)) = t'$$

$$\in A_{\mathrm{elm}} \qquad (\because \text{i.h. より } \mathrm{stkeval}_{\mathrm{stk}}(t) \in A_{\mathrm{stk}})$$

より正しい.

- 任意の $t \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}$ について，$\mathrm{stkeval}_{\mathrm{stk}}(t) = \mathrm{Empty}$ の時，

$$\mathrm{stkeval}_{\mathrm{elm}}(\mathrm{Head}(t)) = \bot \in A_{\mathrm{elm}}$$

より正しい.

- 任意の $t \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}$ について，$t' = \mathrm{stkeval}_{\mathrm{stk}}(t)$ とする． i.h. より，$t' \in A_{\mathrm{stk}}$ である．ここから A_{stk} の定義より，$t' = \mathrm{Cons}(...)$ でも $t' = \mathrm{Empty}$ でもない場合，ある $m \in \mathbb{N}$ について，$t' = \mathrm{Tail}^m(\eta)$ と書け，さらに $\mathrm{Head}(\mathrm{Tail}^m(\eta)) \in A_{\mathrm{elm}}$ である．よって，

$$\mathrm{stkeval}_{\mathrm{elm}}(\mathrm{Head}(t)) = \mathrm{Head}(t') = \mathrm{Head}(\mathrm{Tail}^m(\eta)) \in A_{\mathrm{elm}}$$

より正しい.

- 任意の $t \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}$ について，$\mathrm{stkeval}_{\mathrm{stk}}(t) = \mathrm{Cons}(\ldots, t')$ の時，

$$\begin{aligned} \mathrm{stkeval}_{\mathrm{stk}}(\mathrm{Tail}(t)) &= t' \\ &\in A_{\mathrm{stk}} \qquad (\because \text{i.h. より } \mathrm{stkeval}_{\mathrm{stk}}(t) \in A_{\mathrm{stk}}) \end{aligned}$$

より正しい.

- 任意の $t \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}$ について，$\mathrm{stkeval}_{\mathrm{stk}}(t) = \mathrm{Empty}$ の時，

$$\mathrm{stkeval}_{\mathrm{stk}}(\mathrm{Tail}(t)) = \mathrm{Empty} \in A_{\mathrm{stk}}$$

より正しい.

- 任意の $t \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}$ について，$t' = \mathrm{stkeval}_{\mathrm{stk}}(t)$ とする． i.h. より，$t' \in A_{\mathrm{stk}}$ である．ここから A_{stk} の定義より，$t' = \mathrm{Cons}(...)$ でも $t' = \mathrm{Empty}$ でない場合，ある $m \in \mathbb{N}$ について，$t' = \mathrm{Tail}^m(\eta)$ と書け，さらに $\mathrm{Tail}(\mathrm{Tail}^m(\eta)) = \mathrm{Tail}^{m+1}(\eta) \in A_{\mathrm{stk}}$ である．よって，

$$\mathrm{stkeval}_{\mathrm{stk}}(\mathrm{Tail}(t)) = \mathrm{Tail}(t') = \mathrm{Tail}^{m+1}(\eta) \in A_{\mathrm{stk}}$$

より正しい.

また，任意の $k \in \mathrm{SS}$ について，$A_k \subseteq \{\mathrm{stkeval}_k(\eta) \mid \eta \in \mathcal{S}_\Sigma(X)_k\}$ を示す．これは，

$$\forall k \in \mathrm{SS} \,.\, \forall \eta \in A_k \,.\, \eta = \mathrm{stkeval}_k(\eta)$$

から導かれる．よって，これを η に関する構造的帰納法で示す.

IB1　$\mathrm{stkeval}_{\mathrm{stk}}(\mathrm{Empty}) = \mathrm{Empty}$ より正しい.

IB2　$m \in \mathbb{N}$，$x \in X_{\mathrm{stk}} \cup (() \rightarrow\!\!\triangleright_{\Sigma} \mathrm{stk})$ について，$\mathrm{stkeval}_{\mathrm{stk}}(\mathrm{Tail}^m(x)) = \mathrm{Tail}^m(x)$ となることは，数学的帰納法で容易に示せる．よって正しい．

IB3　IB2 と同様．

IB4　$x \in X_{\mathrm{elm}} \cup (() \rightarrow\!\!\triangleright_{\Sigma} \mathrm{elm}) \cup \{\bot\}$ について，$\mathrm{stkeval}_{\mathrm{elm}}(x) = x$ より正しい．

IS1　$\eta_1 \in A_{\mathrm{elm}}$，$\eta_2 \in A_{\mathrm{stk}}$ について，

$$\begin{aligned}\mathrm{stkeval}_{\mathrm{stk}}(\mathrm{Cons}(\eta_1, \eta_2)) &= \mathrm{Cons}(\mathrm{stkeval}_{\mathrm{elm}}(\eta_1), \mathrm{stkeval}_{\mathrm{stk}}(\eta_2)) \\ &= \mathrm{Cons}(\eta_1, \eta_2) \qquad (\because \text{i.h.})\end{aligned}$$

より正しい．

IS2　$n \geq 1$，$m \in \mathbb{N}$，$k_1, \ldots, k_n \in \mathrm{SS}$，$\sigma : (k_1, \ldots, k_n) \rightarrow\!\!\triangleright_{\Sigma} \mathrm{stk}$，$\eta_1 \in A_{k_1}, \ldots, \eta_n \in A_{k_n}$ について，

$$\begin{aligned}\mathrm{stkeval}_{\mathrm{stk}}(\mathrm{Tail}^m(\sigma(\eta_1, \ldots, \eta_n))) &= \mathrm{Tail}^m(\sigma(\mathrm{stkeval}_{k_1}(\eta_1), \ldots, \mathrm{stkeval}_{k_n}(\eta_n))) \\ &\qquad (\because \text{IB2 と同様に数学的帰納法で示せる}) \\ &= \mathrm{Tail}^m(\sigma(\eta_1, \ldots, \eta_n)) \\ &\qquad (\because \text{i.h.})\end{aligned}$$

より正しい．

IS3　IS2 と同様．

IS4　IS1 と同様．

よって，題意は示された． ■

また，変数を含まず，要素型だけのアルファベットから構築されるスタックシステムの標準形は，ランク付き木に $\bot$ を追加したものに一致する．

定理 168. $\eta \in \mathcal{S}_{\dot{\Sigma}}(\emptyset)_{\mathrm{elm}}$ について，$\mathrm{stkeval}_{\mathrm{elm}}(\eta) \in \hat{T}_{\Sigma_\bot}$ □

証明. 定理 167 より，以下のように，スタックシステムの標準形に対する構造的帰納法で示す．

IB1, IB2, IB3　適用不能

IB4　$\sigma \in \Sigma^{(0)} \cup \{\bot\}$ について，$\sigma \in \hat{T}_{\Sigma_\bot}$ より正しい．

IS1, IS2, IS3　適用不能

IS4　$n \geq 1$，$\sigma^{(n)} \in \Sigma$，$\mathrm{stkeval}_{\mathrm{elm}}(\eta_1), \ldots, \mathrm{stkeval}_{\mathrm{elm}}(\eta_n) \in \mathcal{S}_{\dot{\Sigma}}(\emptyset)_{\mathrm{elm}}$ について，i.h. より $\mathrm{stkeval}_{\mathrm{elm}}(\eta_1), \ldots, \mathrm{stkeval}_{\mathrm{elm}}(\eta_n) \in \hat{T}_{\Sigma_\bot}$．よって，
$\sigma(\mathrm{stkeval}_{\mathrm{elm}}(\eta_1), \ldots, \mathrm{stkeval}_{\mathrm{elm}}(\eta_n))) \in \hat{T}_{\Sigma_\bot}$ より正しい．

■

ところで，スタックシステム評価器はその評価を**最内戦略** (*inside-out strategy*) で行う．しかし，一般にスタックシステムの評価は合流性を持つ戦略を持たない簡約システムで定義できる．

定義 169 (スタックシステム簡約器). 簡約システム $(\hat{\mathcal{S}}_\Sigma(X), \Rightarrow_{\mathrm{stkeval}})$ を，以下のように定義する．

$$\eta_1 \Rightarrow_{\mathrm{stkeval}} \eta_2 \text{ iff } \left(\begin{array}{l} \exists \pi \in \mathrm{paths}(\eta_1), \\ \quad \eta_1' = \mathrm{subtree}_{\eta_1}(\pi), \\ \quad \eta_2', \eta_2 = \eta_1[\pi \leftarrow \eta_2'] \ . \\ (\eta_1' \to \eta_2') \in R \end{array} \right)$$

ただし，$R \subseteq \bigcup_{k \in \mathrm{SS}} \{\eta_1 \to \eta_2 \mid \eta_1, \eta_2 \in \mathcal{S}_\Sigma(X)_k\}$ は以下のように定義する．

$$\begin{aligned} R \stackrel{\mathrm{def}}{=} & \{\mathrm{Head}(\mathrm{Empty}) \to \bot, \mathrm{Tail}(\mathrm{Empty}) \to \mathrm{Empty}\} \\ & \cup \{\mathrm{Head}(\mathrm{Cons}(\eta_1, \eta_2)) \to \eta_1 \mid \eta_1 \in \mathcal{S}_\Sigma(X)_{\mathrm{elm}}, \eta_2 \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}\} \\ & \cup \{\mathrm{Tail}(\mathrm{Cons}(\eta_1, \eta_2)) \to \eta_2 \mid \eta_1 \in \mathcal{S}_\Sigma(X)_{\mathrm{elm}}, \eta_2 \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}\} \end{aligned}$$

□

この簡約システムは，型を保存する．

系 170. $s \in \mathrm{SS}$，η_1, η_2 について，$\eta_1 \Rightarrow^*_{\mathrm{stkeval}} \eta_2$ で，$\eta_1 \in \mathcal{S}_\Sigma(X)_s$ の時，$\eta_2 \in \mathcal{S}_\Sigma(X)_s$ □

証明. 簡約の長さに関する数学的帰納法で，容易に示せる． ■

簡約システムが，合流性を持つことは補題 56 から，局所合流性と停止性を示すことで得られる．よって，これらを示せば良い．

補題 171. $\Rightarrow_{\mathrm{stkeval}}$ は局所合流性を持つ． □

証明.

$$\begin{aligned} D \stackrel{\mathrm{def}}{=} & \{\mathrm{Head}(\mathrm{Empty})\} \cup \{\mathrm{Head}(\mathrm{Cons}(\eta_1, \eta_2)) \mid \eta_1 \in \mathcal{S}_\Sigma(X)_{\mathrm{elm}}, \mathcal{S}_\Sigma(X)_{\mathrm{stk}}\} \\ & \cup \{\mathrm{Tail}(\mathrm{Empty})\} \cup \{\mathrm{Tail}(\mathrm{Cons}(\eta_1, \eta_2)) \mid \eta_1 \in \mathcal{S}_\Sigma(X)_{\mathrm{elm}}, \mathcal{S}_\Sigma(X)_{\mathrm{stk}}\} \end{aligned}$$

とする．任意の $t \Rightarrow_{\mathrm{stkeval}} t_1$，$t \Rightarrow_{\mathrm{stkeval}} t_2$ について，以下を満たす $\pi_1, \pi_2 \in \mathrm{paths}(t)$ が存在する．

- $\mathrm{subtree}_t(\pi_1), \mathrm{subtree}_t(\pi_2) \in D$
- $t[\pi_1 \leftarrow \mathrm{subtree}_{t_1}(\pi_1)] = t_1$
- $t[\pi_2 \leftarrow \mathrm{subtree}_{t_2}(\pi_2)] = t_2$

この時，以下のように場合分けを行う．

- $\pi_1 = \pi_2$ の時，定義より $t_1 = t_2$ は明らかより，正しい．
- $\pi_1 \sqsubseteq \pi_2 \wedge \pi_1 \neq \pi_2$ の時，以下のように場合分けを行う．
  - $\mathrm{subtree}_t(\pi_1) \in \{\mathrm{Head}(\mathrm{Empty}), \mathrm{Tail}(\mathrm{Empty})\}$ の場合，条件を満たす π_2 は存在しない．

- $\mathrm{subtree}_t(\pi_1) = \mathrm{Head}(\mathrm{Cons}(\eta_1, \eta_2))$ の場合，$i \in [2]$ について，$\pi_2 = \pi_1 1 i \pi_2'$ と書ける．
  - $i = 1$ の時，$\mathrm{subtree}_{t_2}(\pi_1) = \mathrm{Head}(\mathrm{Cons}(\eta_1[\pi_2' \leftarrow \mathrm{subtree}_{t_2}(\pi_2)], \eta_2))$ と書ける．よって，t_2 に対し π_1 の部分を簡約でき，$t' = t[\pi_1 \leftarrow \eta_1[\pi_2' \leftarrow \mathrm{subtree}_{t_2}(\pi_2)]]$ とすると，$t_2 \Rightarrow_{\mathrm{stkeval}} t'$ である．また，t_1 に対しても $\pi_1\pi_2'$ の部分を簡約でき $t_1 \Rightarrow_{\mathrm{stkeval}} t_1[\pi_1\pi_2' \leftarrow \mathrm{subtree}_{t_2}(\pi_2)] = t'$ である．よって正しい．
  - $i = 2$ の時，$\mathrm{subtree}_{t_2}(\pi_1) = \mathrm{Head}(\mathrm{Cons}(\eta_1, ...))$ と書ける．よって，t_2 に対し π_1 の部分を簡約でき，$t_2 \Rightarrow_{\mathrm{stkeval}} t[\pi_1 \leftarrow \eta_1] = t_1$ より正しい．

  $\mathrm{subtree}_t(\pi_1) = \mathrm{Tail}(\mathrm{Cons}(\eta_1, \eta_2))$ の場合も同様.
- $\pi_2 \sqsubset \pi_1 \wedge \pi_1 \neq \pi_2$ の時，$\pi_1 \sqsubset \pi_2$ の時と同様.
- $\pi_1 \not\sqsubseteq \pi_2 \wedge \pi_2 \not\sqsubseteq \pi_1$ の時，独立性より $t' = t[\pi_1 \leftarrow \mathrm{subtree}_{t_1}(\pi_1), \pi_2 \leftarrow \mathrm{subtree}_{t_2}(\pi_2)]$ とした時，$t_1 \Rightarrow_{\mathrm{stkeval}} t'$，$t_2 \Rightarrow_{\mathrm{stkeval}} t'$ より正しい．

■

補題 172. $\Rightarrow_{\mathrm{stkeval}}$ は停止性を持つ． □

証明. $f \stackrel{\mathrm{def}}{=} \eta \mapsto |\mathrm{occ}_{\eta_1}(\mathrm{Head}) \cup \mathrm{occ}_{\eta_1}(\mathrm{Tail})|$ とする．この時，任意の $\eta_1 \Rightarrow_{\mathrm{stkeval}} \eta_2$ に対して，$f(\eta_1) > f(\eta_2)$ を示す．補題 171 の D について，以下を満たす $\pi \in \mathrm{paths}(\eta_1)$ が存在する．

- $\mathrm{subtree}_{\eta_1}(\pi) \in D$
- $\eta_1[\pi \leftarrow \mathrm{subtree}_{\eta_2}(\pi)] = \eta_2$

この時，定義より $f(\mathrm{subtree}_{\eta_1}(\pi)) = 1 + n$ となる $n \in \mathbb{N}$ が存在し，$f(\mathrm{subtree}_{\eta_2}(\pi)) \leq n$ である．よって，$f(\eta_2) < f(\eta_1)$．さて，$\Rightarrow_{\mathrm{stkeval}}$ で無限簡約可能な η が存在した時，$\eta \Rightarrow_{\mathrm{stkeval}}^{f(\eta)+1} \eta'$ となる η' が存在する．この時 $f(\eta') \leq f(\eta) - (f(\eta) + 1) < 0$ だが，これは $f(\eta') \geq 0$ に矛盾する．よって，題意は示された． ■

以上から，スタックシステム簡約器が合流性を持つことが示された．

系 173. $\Rightarrow_{\mathrm{stkeval}}$ は合流性を持つ． □

証明. 補題 171，補題 172，補題 56 より． ■

これは，戦略に関係なくその評価結果が一致することを示している．よって，最内戦略によるスタックシステム評価器の評価結果は，スタックシステム簡約器の結果と一致する．

定理 174. 任意の $\eta \in \mathcal{S}_\Sigma(X)_k$ について，$\mathrm{stkeval}_k(\eta) = \mathrm{nf}(\Rightarrow_{\mathrm{stkeval}}, \eta)$ □

証明.

$$\eta \Rightarrow_{\mathrm{stkeval}}^* \mathrm{stkeval}_k(\eta) = \mathrm{nf}(\Rightarrow_{\mathrm{stkeval}}, \mathrm{stkeval}_k(\eta))$$

を示せば，$\mathrm{nf}(\Rightarrow_{\mathrm{stkeval}}, \eta)$ の一意性から，題意は導かれる．これは，容易な構造的帰納法で示せる． ■

なお，スタックシステム簡約器は**最外戦略** (*outside-in strategy*) によっても，同等の決定的な簡約システムを構築できる．

定義 175 (決定的スタックシステム簡約器). 簡約システム $(\hat{\mathcal{S}}_\Sigma(X), \Rightarrow^D_{\mathrm{stkeval}})$ を，以下のように構造的帰納法で定義する．

IB1　$k \in \mathrm{SS}$，$x \in X_k$ について，x は既約

IB2　$k \in \mathrm{SS}$，$\sigma : () \rightarrowtriangle_{\Sigma\cup\{\bot,\mathrm{Empty}\}} k$ について，σ は既約

IS　以下のように場合分けを行う．

- $\mathrm{Head}(\mathrm{Empty}) \Rightarrow^D_{\mathrm{stkeval}} \bot$
- $\mathrm{Tail}(\mathrm{Empty}) \Rightarrow^D_{\mathrm{stkeval}} \mathrm{Empty}$
- 任意の $t_1 \in \mathcal{S}_\Sigma(X)_{\mathrm{elm}}$，$t_2 \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}$ について，$\mathrm{Head}(\mathrm{Cons}(t_1, t_2)) \Rightarrow^D_{\mathrm{stkeval}} t_1$
- 任意の $t_1 \in \mathcal{S}_\Sigma(X)_{\mathrm{elm}}$，$t_2 \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}$ について，$\mathrm{Tail}(\mathrm{Cons}(t_1, t_2)) \Rightarrow^D_{\mathrm{stkeval}} t_2$
- それ以外の場合の $n \geq 1$，$k_1, \ldots, k_n, k \in \mathrm{SS}$，$\sigma : (k_1, \ldots, k_n) \rightarrowtriangle_{\Sigma\cup\{\mathrm{Cons},\mathrm{Head},\mathrm{Tail}\}} k$，$t_1 \in \mathcal{S}_\Sigma(X)_{k_1}, \ldots, t_i \in \mathcal{S}_\Sigma(X)_{k_i}, \ldots, t_n \in \mathcal{S}_\Sigma(X)_{k_n}, t'_i \in \mathcal{S}_\Sigma(X)_{k_i}$ について，$t_1, \ldots, t_{i-1}$ が $\Rightarrow^D_{\mathrm{stkeval}}$ で既約で，$t_i \Rightarrow^D_{\mathrm{stkeval}} t'_i$ の時，$\sigma(t_1, \ldots, t_i, \ldots, t_n) \Rightarrow^D_{\mathrm{stkeval}} \sigma(t_1, \ldots, t'_i, \ldots, t_n)$

□

定理 176. 任意の $\eta \in \hat{\mathcal{S}}_\Sigma(X)$ について，$\mathrm{nf}(\Rightarrow_{\mathrm{stkeval}}, \eta) = \mathrm{nf}(\Rightarrow^D_{\mathrm{stkeval}}, \eta)$ □

証明.

$$\eta_1 \Rightarrow^D_{\mathrm{stkeval}} \eta_2 \text{ implies } \eta_1 \Rightarrow_{\mathrm{stkeval}} \eta_2$$

は容易な構造的帰納法で確かめられる．ここから，$\eta \Rightarrow^*_{\mathrm{stkeval}} \mathrm{nf}(\Rightarrow^D_{\mathrm{stkeval}}, \eta)$ である．これが既約であれば，$\mathrm{nf}(\Rightarrow_{\mathrm{stkeval}}, \eta)$ の一意性から題意は導かれる．これも，容易な構造的帰納法で確かめられる． ■

以上より，最内戦略か最外戦略かは結果に影響しないことも示せる．

系 177. 任意の $\eta \in \mathcal{S}_\Sigma(X)_k$ について，$\mathrm{stkeval}_s(\eta) = \mathrm{nf}(\Rightarrow^D_{\mathrm{stkeval}}, \eta) = \mathrm{nf}(\Rightarrow_{\mathrm{stkeval}}, \eta)$ □

証明. 定理 174，定理 176 より． ■

最外戦略の評価システムは，属性付き木変換器に対するスタック木変換器への拡張を考える際，重要になる．また，最外戦略のスタック評価システムに対する議論を明確にするため，ス

タックシステムに対する**弱標準形**の概念を導入する．弱標準形は，スタックシステムに対し，出力の一部が露出するまで最外戦略で評価を進めた場合の，正規形のことである．

定義 178 (スタックシステム弱評価器). 関数の族 $\mathrm{stkwkeval} : \prod_{k\in \mathrm{SS}} \mathcal{S}_\Sigma(X)_k \to \mathcal{S}_\Sigma(X)_k$ を，以下のように構造的帰納法で定義する．

IB1　$k \in \mathrm{SS}$，$x \in X_k$ について，$\mathrm{stkwkeval}_k(x) \stackrel{\mathrm{def}}{=} x$

IB2　$k \in \mathrm{SS}$，$\sigma : () \to_{\Sigma\cup\{\bot,\mathrm{Empty}\}} k$ について，$\mathrm{stkwkeval}_k(\sigma) \stackrel{\mathrm{def}}{=} \sigma$

IS　以下のように場合分けを行う．

- 任意の $n \geq 1$，$k_1, \ldots, k_n, k \in \mathrm{SS}$，$\sigma : (k_1, \ldots, k_n) \to_{\Sigma\cup\{\mathrm{Cons}\}} k$，$t_1 \in \mathcal{S}_\Sigma(X)_{k_1}, \ldots, t_n \in \mathcal{S}_\Sigma(X)_{k_n}$ について，

$$\mathrm{stkwkeval}_k(\sigma(t_1, \ldots, t_n)) \stackrel{\mathrm{def}}{=} \sigma(t_1, \ldots, t_n)$$

- 任意の $t \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}$ について，$t' \stackrel{\mathrm{def}}{=} \mathrm{stkwkeval}_{\mathrm{stk}}(t)$ とした時，

$$\mathrm{stkwkeval}_{\mathrm{elm}}(\mathrm{Head}(t)) \stackrel{\mathrm{def}}{=} \begin{cases} \mathrm{stkwkeval}_{\mathrm{elm}}(\eta) & (t' = \mathrm{Cons}(\eta, \ldots)) \\ \bot & (t' = \mathrm{Empty}) \\ \mathrm{Head}(t') & (\mathrm{otherwise}) \end{cases}$$

- 任意の $t \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}$ について，$t' \stackrel{\mathrm{def}}{=} \mathrm{stkwkeval}_{\mathrm{stk}}(t)$ とした時，

$$\mathrm{stkwkeval}_{\mathrm{elm}}(\mathrm{Tail}(t)) \stackrel{\mathrm{def}}{=} \begin{cases} \mathrm{stkwkeval}_{\mathrm{elm}}(\eta) & (t' = \mathrm{Cons}(\ldots, \eta)) \\ \mathrm{Empty} & (t' = \mathrm{Empty}) \\ \mathrm{Tail}(t') & (\mathrm{otherwise}) \end{cases}$$

この時，$k \in \mathrm{SS}$，$\eta \in \mathcal{S}_\Sigma(X)_k$ について，$\mathrm{stkwkeval}_k(\eta)$ を η の弱標準形と呼ぶ． □

例 179. $\Sigma = \{\mathrm{A}^{(2)}, \mathrm{B}^{(0)}\}$ について，$t \in \hat{\mathcal{S}}_\Sigma$ を以下のように定義する．

$$t \stackrel{\mathrm{def}}{=} \mathrm{Head}(\mathrm{Tail}(\mathrm{Cons}(\mathrm{B}, \mathrm{Cons}(\mathrm{A}(\mathrm{Head}(\mathrm{Empty}), \mathrm{Head}(\mathrm{Cons}(\mathrm{B}, \mathrm{Empty}))), \mathrm{Empty}))))$$

この時，

$$\mathrm{stkwkeval}_{\mathrm{stk}}(t) = \mathrm{A}(\mathrm{Head}(\mathrm{Empty}), \mathrm{Head}(\mathrm{Cons}(\mathrm{B}, \mathrm{Empty})))$$

となる． □

一般に弱標準形は，以下の形に制限される．

定理 180 (スタックシステムの弱標準形). SS-型付きアルファベット Σ，SS-型付き集合 X について，集合の族 $\{A_k\}_{k\in \mathrm{SS}}$ を，以下を満たす最小の集合で定義する．

C1　$m \in \mathbb{N}$，$x \in X_{\mathrm{stk}}$ について，$\mathrm{Tail}^m(x) \in A_{\mathrm{stk}}$

C2　$m \in \mathbb{N}$，$x \in X_{\mathrm{stk}}$ について，$\mathrm{Head}(\mathrm{Tail}^m(x)) \in A_{\mathrm{elm}}$

C3　$x \in X_{\mathrm{elm}}$ について，$x \in A_{\mathrm{elm}}$

C4　$\sigma : (k_1, \ldots, k_n) \to_{\{\mathrm{Cons},\mathrm{Empty}\}} \mathrm{stk}$，$\eta_1 \in \mathcal{S}_\Sigma(X)_{k_1}, \ldots, \eta_n \in \mathcal{S}_\Sigma(X)_{k_n}$ について，$\sigma(\eta_1, \ldots, \eta_n) \in A_{\mathrm{stk}}$

C5　$m \in \mathbb{N}$，$\sigma : (k_1, \ldots, k_n) \to_\Sigma \mathrm{stk}$，$\eta_1 \in \mathcal{S}_\Sigma(X)_{k_1}, \ldots, \eta_n \in \mathcal{S}_\Sigma(X)_{k_n}$ について，$\mathrm{Tail}^m(\sigma(\eta_1, \ldots, \eta_n)) \in A_{\mathrm{stk}}$

C6　$m \in \mathbb{N}$，$\sigma : (k_1, \ldots, k_n) \to_\Sigma \mathrm{stk}$，$\eta_1 \in \mathcal{S}_\Sigma(X)_{k_1}, \ldots, \eta_n \in \mathcal{S}_\Sigma(X)_{k_n}$ について，$\mathrm{Head}(\mathrm{Tail}^m(\sigma(\eta_1, \ldots, \eta_n))) \in A_{\mathrm{elm}}$

C7　$\sigma : (k_1, \ldots, k_n) \to_{\Sigma\cup\{\bot\}} \mathrm{elm}, \eta_1 \in \mathcal{S}_\Sigma(X)_{k_1}, \ldots, \eta_n \in \mathcal{S}_\Sigma(X)_{k_n}$ について，$\sigma(\eta_1, \ldots, \eta_n) \in A_{\mathrm{elm}}$

この時，任意の $k \in \mathrm{SS}$ について，$A_k = \{\mathrm{stkwkeval}_k(\eta) \mid \eta \in \mathcal{S}_\Sigma(X)_k\}$ □

証明. 定理 167 と同様に示せる． ■

スタックシステムは決定的スタックシステム簡約器で，弱標準形へ簡約することができる．

定理 181. 任意の $\eta \in \mathcal{S}_\Sigma(X)_k$ について，$\eta \Rightarrow_{\mathrm{stkeval}}^{D\ *} \mathrm{stkwkeval}_k(\eta)$ □

証明. 構造的帰納法で示す．

IB1　$k \in \mathrm{SS}$，$x \in X_k$ について，$x = \mathrm{stkwkeval}_k(x)$ より正しい．

IB2　$k \in \mathrm{SS}$，$\sigma : () \to_{\Sigma\cup\{\bot,\mathrm{Empty}\}} k$ について，$\sigma = \mathrm{stkwkeval}_k(\sigma)$ より正しい．

IS　以下のように場合分けを行う．

- $k \in \mathrm{SS}$，$\sigma : \mathrm{stk} \to_{\{\mathrm{Head},\mathrm{Tail}\}} k$，$t \in \mathcal{S}_\Sigma(X)_{\mathrm{stk}}$ について，i.h. から $\sigma(t) \Rightarrow_{\mathrm{stkeval}}^{D\ *} \sigma(\mathrm{stkwkeval}_{\mathrm{stk}}(t))$ であることは，簡約の長さに関する数学的帰納法で容易に示せる．この時，$t' \stackrel{\mathrm{def}}{=} \mathrm{stkwkeval}_{\mathrm{stk}}(t)$ とすると，
  - $\sigma = \mathrm{Head}$，$t' = \mathrm{Empty}$ の時，

$$\mathrm{Head}(\mathrm{Empty}) \Rightarrow_{\mathrm{stkeval}}^{D} \bot = \mathrm{stkwkeval}_k(\mathrm{Head}(t))$$

    より正しい．
  - $\sigma = \mathrm{Head}$，$t' = \mathrm{Cons}(\eta, \ldots)$ の時，

$$\begin{aligned}\mathrm{Head}(\mathrm{Cons}(\eta, \ldots)) &\Rightarrow_{\mathrm{stkeval}}^{D} \eta \\ &\Rightarrow_{\mathrm{stkeval}}^{D\ *} \mathrm{stkwkeval}_{\mathrm{elm}}(\eta) \qquad (\because \text{i.h.}) \\ &= \mathrm{stkwkeval}_{\mathrm{elm}}(\mathrm{Head}(t))\end{aligned}$$

    より正しい．
  - $\sigma = \mathrm{Head}$ でそれ以外の場合，$\mathrm{Head}(t') = \mathrm{stkwkeval}_{\mathrm{stk}}(\mathrm{Head}(t))$ より正しい．
  - $\sigma = \mathrm{Tail}$ の時，$\sigma = \mathrm{Head}$ の時と同様．

- $n \geq 1$，$k_1, \ldots, k_n, k \in \mathrm{SS}$，$\sigma : (k_1, \ldots, k_n) \rightarrowtriangle_{\Sigma \cup \{\mathrm{Cons}\}} k$，
  $t_1 \in \mathcal{S}_\Sigma(X)_{k_1}, \ldots, t_n \in \mathcal{S}_\Sigma(X)_{k_n}$ について，

$$\sigma(t_1, \ldots, t_n) = \mathrm{stkwkeval}_k(\sigma(t_1, \ldots, t_n))$$

  より正しい．

■

この時，弱標準形になるまでの簡約回数を，**スタックシステム弱評価回数**として導入する．

定義 182 (スタックシステム弱評価回数). $\mathrm{stkwevstep} : \hat{\mathcal{S}}_\Sigma(X) \to \mathbb{N}$ を，

$$\forall k \in \mathrm{SS}, \eta \in \mathcal{S}_\Sigma(X)_k \,.\, \eta \Rightarrow_{\mathrm{stkeval}}^{D\ n} \mathrm{stkwkeval}_k(\eta) \text{ implies } \mathrm{stkwevstep}(\eta) = n$$

を満たすように定義する． □

なお，型無しの木をスタックシステムと見なすことができる．この場合，次の射影によりスタックシステムへの射影を行う．射影は，

- 要素をスタック型とみなしたい場合，要素を 1 つ持ったスタックの値とする．
- スタックを要素型とみなしたい場合，スタックを先頭の要素を表す要素型の値とする．

として，その変換を型に不整合がある部分で再帰的に適用していくことで，行われる．

定義 183 (スタックシステムへの射影). SS-型付きアルファベット Σ，SS-型付き集合 X について，関数の族 $\mathrm{stkproj} : \prod_{k \in \mathrm{SS}} \hat{T}_{\Sigma \cup \bar{\mathrm{S}}\mathrm{TK}}(\bar{X}) \to \mathcal{S}_\Sigma(X)_k$ を，以下のように定義する．

$$\mathrm{stkproj}_s \stackrel{\mathrm{def}}{=} t \mapsto \mathrm{rankproj}_{p,s}(t)$$

ただし，$p_{s_1,s_2} : \mathcal{S}_\Sigma(X)_{s_1} \to \mathcal{S}_\Sigma(X)_{s_2}$ は以下のように定義する．

$$p_{s_1,s_2} \stackrel{\mathrm{def}}{=} t \mapsto \begin{cases} t & (s_1 = s_2) \\ \mathrm{Cons}(t, \mathrm{Empty}) & (s_1 = \mathrm{elm}, s_2 = \mathrm{stk}) \\ \mathrm{Head}(t) & (s_1 = \mathrm{stk}, s_2 = \mathrm{elm}) \end{cases}$$

□

例 184. $\Sigma = \{\mathrm{A}^{(2)}, \mathrm{B}^{(0)}\}$ について，$t \in \hat{T}_{\Sigma \cup \mathrm{S}\bar{\mathrm{T}}\mathrm{K}}$ を以下のように定義する．

$$t \stackrel{\mathrm{def}}{=} \mathrm{Head}(\mathrm{Head}(\mathrm{A}(\mathrm{Tail}(\mathrm{Empty}), \mathrm{Cons}(\mathrm{Empty}, \mathrm{B}))))$$

この時，

$$\mathrm{stkproj}_{\mathrm{stk}}(t) = \mathrm{Cons}(\mathrm{Head}(\mathrm{Cons}(\mathrm{Head}(\mathrm{Cons}(\\ \mathrm{A}(\mathrm{Head}(\mathrm{Tail}(\mathrm{Empty})), \mathrm{Head}(\mathrm{Cons}(\mathrm{Head}(\mathrm{Empty}), \mathrm{Cons}(\mathrm{B}, \mathrm{Empty})))),$$

$$\mathrm{Empty})), \mathrm{Empty})), \mathrm{Empty})$$

となる．□

この射影は，後ほど，スタック木変換器と木変換器の表現力の比較の際，使用する．スタック木変換器は，木からスタックへの関数，スタック木変換を意味論に持つ構文構造の総称である．なお，スタックの意味論は，$\bot$ を含む木の無限列である．

定義 185 (スタック木変換). ランク付きアルファベット Σ, Δ について，$f : \hat{T}_\Sigma \to (\mathbb{N} \to \hat{T}_{\Delta_\bot})$ をスタック木変換と呼ぶ．

また，以下の表記を用いる．

- スタック木変換 $f : \hat{T}_\Sigma \to (\mathbb{N} \to \hat{T}_{\Delta_\bot})$，$n \in \mathbb{N}$ について，$f(-)(n) : \hat{T}_\Sigma \to \hat{T}_{\Delta_\bot}$ を以下のように定義する．

$$f(-)(n) \stackrel{\mathrm{def}}{=} x \mapsto f(x)(n)$$

- 木変換 $f : \hat{T}_\Sigma \to \hat{T}_\Delta$ について，$f_\omega : \hat{T}_\Sigma \to (\mathbb{N} \to \hat{T}_{\Delta_\bot})$ を以下のように定義する．

$$f_\omega \stackrel{\mathrm{def}}{=} x \mapsto n \mapsto \begin{cases} f(x) & (n = 0) \\ \bot & (n > 1) \end{cases}$$

- 木変換のクラス F について，スタック木変換のクラス $\{f_\omega \mid f \in F\}$ を F_ω と表記する．

□

スタック木変換器の木変換による意味論は，スタック木変換の意味論 f に対して，$f(-)(0)$，つまり出力の木の列の先頭の要素を出力木として定める．なお，このスタック木変換から定まる木変換において，スタック木変換のクラスの部分関係は保存される．

定理 186. スタック木変換クラス F_ω, G_ω について，$F \stackrel{\mathrm{def}}{=} \{f_\omega(-)(0) \mid f_\omega \in F_\omega\}, G \stackrel{\mathrm{def}}{=} \{g_\omega(-)(0) \mid g_\omega \in G_\omega\}$ とする．この時，以下が成り立つ．

$$F_\omega \subseteq G_\omega \implies F \subseteq G$$

□

証明. $F_\omega \subseteq G_\omega$ とする．この時，$f_\omega(-)(0) \in F$ について，$f_\omega \in G_\omega$ より $f_\omega(-)(0) \in G_\omega$．よって，題意は示された．■

また，その対偶も成り立つ．

系 187. スタック木変換クラス F_ω, G_ω において，$F \stackrel{\mathrm{def}}{=} \{f_\omega(-)(0) \mid f_\omega \in F_\omega\}, G \stackrel{\mathrm{def}}{=} \{g_\omega(-)(0) \mid g_\omega \in G_\omega\}$ とする．この時，以下が成り立つ．

$$F \not\subseteq G \implies F_\omega \not\subseteq G_\omega$$

□

証明. 定理 186 より. ■

4.2 スタックトップダウン木変換器

以上のスタックシステムを基に，TDTT を拡張したものを，スタックトップダウン木変換器 (StkTT) と呼ぶ． StkTT は，規則の右辺を，ランク付き木ではなく，スタックシステムで構成させる．

定義 188 (スタックトップダウン木変換器 (StkTT)). スタックトップダウン木変換器は，以下の要素の組 $M = (Q, \Sigma, \Delta, e, R)$ である．

Q　状態記号のランク付きアルファベットで，$Q = Q^{(1)}$.
Σ　入力記号のランク付きアルファベット.
Δ　出力記号のランク付きアルファベットで，$\bot \notin \Delta$.
e　初期式で，$e \in \mathrm{RHS}_M(\mathrm{X}_1)_{\mathrm{stk}}$.
R　書き換え規則の集合で，以下を満たす.

$$R \in \mathcal{P}_{fin}(\{q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n)) \to \eta \mid q \in Q, \sigma^{(n)} \in \Sigma, \eta \in \mathrm{RHS}_M(\mathrm{X}_n)_{\mathrm{stk}}\})$$

ただし，$\mathrm{RHS}_M(X) \in \prod_{k \in \mathrm{SS}} \mathcal{P}(\mathcal{S}_{\dot{\Delta} \cup \ddot{Q}}(\{\mathrm{elm} \mapsto X\})_k)$ は，以下のように帰納的に定義される集合の族 $\{U_k\}_{k \in \mathrm{SS}}$ で定義する.

IB1　$q : \mathrm{elm} \twoheadrightarrow_{\ddot{Q}} k$，$x \in X$ について，$q(x) \in U_k$.
IB2　$\delta : () \twoheadrightarrow_{\dot{\Delta} \cup \mathrm{STK}} k$ について，$\delta \in U_k$.
IS　$n \geq 1$，$\delta : (k_1, \ldots, k_n) \twoheadrightarrow_{\dot{\Delta} \cup \mathrm{STK}} k$，$\eta_1 \in U_{k_1}, \ldots, \eta_n \in U_{k_n}$ について，$\delta(\eta_1, \ldots, \eta_n) \in U_k$.

□

StkTT は，TDTT と同じように現在の状態と入力の木のラベルを見て，適用する規則を選びながら出力を構築していく．ただ，TDTT と異なるのは，規則の適用の他にスタック評価も行われる点である.

例 189. StkTT $M = (Q, \Sigma, \Delta, e, R)$ を，以下のように定義する.

$$Q \stackrel{\mathrm{def}}{=} \{q_1, q_2\}$$
$$\Sigma \stackrel{\mathrm{def}}{=} \{\mathrm{a}^{(1)}, \mathrm{b}^{(1)}, \$^{(0)}\}$$
$$\Delta \stackrel{\mathrm{def}}{=} \{\top^{(0)}\}$$
$$e \stackrel{\mathrm{def}}{=} q_1(\mathrm{x}_1)$$

$$
\begin{aligned}
R \stackrel{\text{def}}{=} \{ & q_1(\mathrm{a}(\mathrm{x}_1)) \to \mathrm{Tail}(q_1(\mathrm{x}_1)), \\
& q_1(\mathrm{b}(\mathrm{x}_1)) \to \mathrm{Cons}(\bot, q_2(\mathrm{x}_1)), \\
& q_1(\$) \to \mathrm{Cons}(\top, \mathrm{Empty}), \\
& q_2(\mathrm{a}(\mathrm{x}_1)) \to \mathrm{Empty}, \\
& q_2(\mathrm{b}(\mathrm{x}_1)) \to \mathrm{Cons}(\bot, q_2(\mathrm{x}_1)), \\
& q_2(\$) \to \mathrm{Cons}(\top, \mathrm{Empty})\}
\end{aligned}
$$

この M は，与えられた木が，ある $n \in \mathbb{N}$ において $\mathrm{a}^n(\mathrm{b}^n(\$))$ の形になっているかを判定する StkTT である．$\top$ が積まれたスタックに対して，a の数だけ Tail 操作を，b の数だけ $\bot$ をスタックに積む操作を行う．Tail 操作の数と $\bot$ を積む操作が相殺されれば，$\top$ だけが積まれたスタックに評価されるが，両者の数が合っていない場合，積まれた $\bot$ が残るか，Empty になるかになる．そこから，以下のような木変換を定義する．

$$
\begin{aligned}
[\![M]\!](\mathrm{a}^2(\mathrm{b}^2(\$))) &= \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{Head}(\mathrm{Tail}^2(\mathrm{Cons}(\bot, \mathrm{Cons}(\bot, \mathrm{Cons}(\top, \mathrm{Empty})))))) = \top \\
[\![M]\!](\mathrm{a}^5(\mathrm{b}^2(\$))) &= \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{Head}(\mathrm{Tail}^5(\mathrm{Cons}(\bot, \mathrm{Cons}(\bot, \mathrm{Cons}(\top, \mathrm{Empty})))))) = \bot
\end{aligned}
$$

なお，M の意味論 $[\![M]\!]$ の形式的な定義は，後ほど導入する． □

木変換器と同様，スタック木変換器でも全域的決定的性を定義できる．

定義 190 (全域的決定的 StkTT). StkTT $M = (Q, \Sigma, \Delta, e, R)$ が

$$
\forall q \in Q, \sigma^{(n)} \in \Sigma \,.\, \exists! q(\sigma(\vec{\mathrm{x}})) \to \eta \in R
$$

を満たすとき，全域的決定的であるという． □

例 189 は全域的決定的の例になっている．以降，特に断りのない限り StkTT は全域的決定的であるものとする．StkTT について，意味論を決定づける簡約システムを導入する．

定義 191 (StkTT の簡約システム). StkTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X について，簡約システム $(\bigcup_{k \in \mathrm{SS}} \mathrm{SF}_M(X)_k, \Rightarrow_{M,X})$ を，以下のように定義する．

$$
\mathrm{SF}_M(X) \stackrel{\text{def}}{=} \mathrm{RHS}_M(\hat{T}_\Sigma(X))
$$

$$
\eta_1 \Rightarrow_{\mathrm{stkeval}} \eta_2 \text{ iff } \left(\begin{array}{l} \exists \pi \in \mathrm{paths}(\eta_1), \\ \quad \eta_1' = \mathrm{subtree}_{\eta_1}(\pi), \\ \quad \eta_2', \eta_2 = \eta_1[\pi \leftarrow \eta_2'] \,. \\ (\eta_1' \to \eta_2') \in R \end{array} \right)
$$

ただし，$R \subseteq \bigcup_{k \in \mathrm{SS}} \{\eta_1 \to \eta_2 \mid \eta_1, \eta_2 \in \mathrm{SF}_M(X)_k\}$ は以下のように定義する．

$$
R \stackrel{\text{def}}{=} \{\mathrm{Head}(\mathrm{Empty}) \to \bot, \mathrm{Tail}(\mathrm{Empty}) \to \mathrm{Empty}\}
$$

$$\cup \{\mathrm{Head}(\mathrm{Cons}(\eta_1, \eta_2)) \to \eta_1 \mid \eta_1 \in \mathrm{SF}_M(X)_{\mathrm{elm}}, \eta_2 \in \mathrm{SF}_M(X)_{\mathrm{stk}}\}$$
$$\cup \{\mathrm{Tail}(\mathrm{Cons}(\eta_1, \eta_2)) \to \eta_2 \mid \eta_1 \in \mathrm{SF}_M(X)_{\mathrm{elm}}, \eta_2 \in \mathrm{SF}_M(X)_{\mathrm{stk}}\}$$
$$\cup \{q(\sigma(t_1, \ldots, t_n)) \to \eta[\mathrm{x}_i \leftarrow t_i]_{i \in [n]} \mid q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n)) \to \eta \in R, t_1, \ldots, t_n \in \hat{T}_\Sigma\}$$

この時，$\Rightarrow_{M,\emptyset}$ を $\Rightarrow_M$ と表記する． □

この簡約システムは，型を保存する．

系 192. StkTT $M = (Q, \Sigma, \Delta, e, R)$，$\eta_1 \Rightarrow^*_{M,X} \eta_2$ について，以下が成り立つ．

$$\eta_1 \in \mathrm{SF}_M(X)_k \implies \eta_2 \in \mathrm{SF}_M(X)_k$$

□

証明. 定義より明らか． ■

また，この簡約システムにおいて，スタックシステムとしての簡約ができる．

系 193. StkTT $M = (Q, \Sigma, \Delta, e, R)$，$\eta_1, \eta_2 \in \mathrm{SF}_M(X)_k$ について，以下が成り立つ．

$$\eta_1 \Rightarrow^*_{\mathrm{stkeval}} \eta_2 \text{ implies } \eta_1 \Rightarrow^*_{M,X} \eta_2$$

□

証明. 定義より明らか． ■

さらに，この簡約システムは局所合流性を持つ．

補題 194. StkTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X について，$\Rightarrow_{M,X}$ は局所合流性を持つ． □

証明. 補題 83，補題 171 と同様に示せる． ■

また，この簡約システムは停止性を持つ．

補題 195. StkTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X について，$\Rightarrow_{M,X}$ は停止性を持つ． □

証明. 任意の $q \in Q$，$t \in \hat{T}_\Sigma(X)$ について，$l(q, t) \in \mathbb{N}$ を，以下のように t に関する構造的帰納法で定義する．

IB1　$x \in X$ について，$l(q, x) \stackrel{\mathrm{def}}{=} 0$.

IB2　$\sigma \in \Sigma^{(0)}$，$q(\sigma) \to \eta \in R$ について，$l(q, \sigma) \stackrel{\mathrm{def}}{=} 1 + \mathrm{count}(\eta)$.

IS　$n \geq 1$，$\sigma^{(n)} \in \Sigma$，$t_1, \ldots, t_n \in \hat{T}_\Sigma(X)$，$q(\sigma) \to \eta \in R$ について，

$$l(q, \sigma(t_1, \ldots, t_n)) \stackrel{\mathrm{def}}{=} 1 + \mathrm{count}(\eta[\mathrm{x}_i \leftarrow t_i]_{i \in [n]}).$$

ただし，$\eta \in \mathrm{SF}_M(X)_k$ について，$\mathrm{count}(\eta)$ は以下のように構造的帰納法で定義する．

IB1　$q \in Q$，$t \in \hat{T}_\Sigma(X)$ について，$\mathrm{count}(q(t)) \stackrel{\mathrm{def}}{=} l(q,t)$.

IB2　$\delta \in \Delta^{(0)} \cup \{\bot, \mathrm{Empty}\}$ について，$\mathrm{count}(\delta) \stackrel{\mathrm{def}}{=} 0$.

IS　以下のように場合分けを行う．

- $n \geq 1$，$\delta^{(n)} \in \Delta \cup \{\mathrm{Cons}\}$ について，

$$\mathrm{count}(\delta(\eta_1, \ldots, \eta_n)) \stackrel{\mathrm{def}}{=} \sum_{i \in [n]} \mathrm{count}(\eta_i).$$

- $\delta \in \{\mathrm{Head}, \mathrm{Tail}\}$ について，

$$\mathrm{count}(\delta(\eta_1)) \stackrel{\mathrm{def}}{=} 1 + \mathrm{count}(\eta_1).$$

また，$t \in \hat{T}_\Sigma(X)$，$\eta \in \mathrm{SF}_M(X)$ について，

$$\mathrm{deriv}_\eta(t) \stackrel{\mathrm{def}}{\iff} \forall q \in Q, \pi \in \mathrm{occ}_\eta(q) \,.\, \exists p \in \mathrm{paths}(t) \,.\, \mathrm{subtree}_\eta(\pi 1) = \mathrm{subtree}_t(p)$$

とおく．この時，任意の $t \in \hat{T}_\Sigma$ について，

- $q \in Q$ について，$q(t)$ の簡約が $l(q,t)$ 回以下で終わること
- $\eta \in \mathrm{SF}_M(X)_k$ について，$\mathrm{deriv}_\eta(t)$ の時，η の簡約が $\mathrm{count}(\eta)$ 回以下で終わること

は，t に関する容易な同時帰納法で示せる．ここから，任意の $\eta \in \mathrm{SF}_M(X)$ について，η の簡約が $\mathrm{count}(\eta)$ 回以下で終わることも，η に関する容易な構造的帰納法で示せる．よって，題意は示された． ■

以上から，StkTT の簡約システムは合流性を持つ．

系 196. StkTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X について，$\Rightarrow_{M,X}$ は合流性を持つ． □

証明. 補題 195，補題 194，補題 56 より． ■

StkTT の簡約システムは，合流性と停止性を持つため，補題 57 より一意な正規形を持つ．要素型の正規形は，$\Rightarrow_M$ においては必ず出力アルファベットのみからなる木になる．

補題 197. StkTT $M = (Q, \Sigma, \Delta, e, R)$ について，以下が成り立つ．

$$\forall \eta \in \mathrm{SF}_M(\emptyset)_{\mathrm{elm}} \,.\, \mathrm{nf}(\Rightarrow_M, \eta) \in \hat{T}_{\Delta_\bot}$$

□

証明. 補題 86 と同様に，$\mathrm{nf}(\Rightarrow_M, \eta) \in \hat{\mathcal{S}}_\Delta(\emptyset)_{\mathrm{elm}}$ であることが示せる．ここから，定理 168 より，$\mathrm{nf}(\Rightarrow_M, \eta) \in \hat{T}_{\Delta_\bot}$ も示せる． ■

よって，StkTT の簡約システムにおける変数を持たない項の正規形は，必ず一意でかつ出力木になる．この時 StkTT に対して，入力木を受け取り，初期式にその木を割り当てたものから

StkTT の簡約システムで得られる木を出力とする，木変換が考えられる．この木変換を，意味論として導入する．

定義 198 (StkTT のスタック意味論). StkTT $M = (Q, \Sigma, \Delta, e, R)$ について，$[\![M]\!]_\omega : \hat{T}_\Sigma \to (\mathbb{N} \to \hat{T}_{\Delta_\perp})$ を以下のように定義する．

$$[\![M]\!]_\omega \stackrel{\text{def}}{=} t \mapsto n \mapsto \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^n(e[\mathrm{x}_1 \leftarrow t])))$$

また，スタック木変換のクラス $\{[\![M]\!]_\omega \mid M$ は (全域的決定的) StkTT$\}$ を StkTT_ω と表記する．□

なお，StkTT が全域的決定的な場合，StkTT の簡約システムを決定的にすることができる．

命題 199 (StkTT の決定的な簡約システム). StkTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X について，$U \stackrel{\text{def}}{=} \mathrm{SF}_M(X)$ として，簡約システム $(\bigcup_{k \in \mathrm{SS}} U_k, \Rightarrow^D_{M,X})$ を，以下のように構造的帰納法で定義する．

IB1　$q : \mathrm{elm} \rightarrowtriangle_{\dot{Q}} s$，$\sigma \in \Sigma$，$\sigma(\vec{t}) \in \hat{T}_\Sigma(X)$，$q(\sigma(\vec{\mathrm{x}})) \to \eta \in R$ について，

$$q(\sigma(\vec{t})) \Rightarrow_{M,X} \eta[\vec{\mathrm{x}} \leftarrow \vec{t}].$$

IB2　$\delta : () \rightarrowtriangle_{\dot{\Delta} \cup \mathrm{STK}} s$ について，δ は既約．

IS　以下のように場合分けを行う．

- $\eta_1 \in U_{\mathrm{elm}}$，$\eta_2 \in U_{\mathrm{stk}}$ について，$\mathrm{Head}(\mathrm{Cons}(\eta_1, \eta_2)) \Rightarrow_{M,X} \eta_1$．
- $\eta_1 \in U_{\mathrm{elm}}$，$\eta_2 \in U_{\mathrm{stk}}$ について，$\mathrm{Tail}(\mathrm{Cons}(\eta_1, \eta_2)) \Rightarrow_{M,X} \eta_2$．
- $\mathrm{Head}(\mathrm{Empty}) \Rightarrow_{M,X} \perp$．
- $\mathrm{Tail}(\mathrm{Empty}) \Rightarrow_{M,X} \mathrm{Empty}$．
- それ以外の場合，$n \geq 1$，$\delta : (s_1, \ldots, s_n) \rightarrowtriangle_{\dot{\Delta} \cup \mathrm{STK}} s$，$\eta_1 \in U_{s_1}, \ldots, \eta_n \in U_{s_n}$，$\eta_1, \ldots, \eta_{i-1}$ が $\Rightarrow_{M,X}$ で既約で，$\eta_i \Rightarrow_{M,X} \eta_i'$ の時，

$$\delta(\eta_1, \ldots, \eta_i, \ldots, \eta_n) \Rightarrow_{M,X} \delta(\eta_1, \ldots, \eta_i', \ldots, \eta_n).$$

この時，任意の $\eta \in U_k$ に対して，$\mathrm{nf}(\Rightarrow_{M,X}, \eta) = \mathrm{nf}(\Rightarrow^D_{M,X}, \eta)$．□

証明. 命題 88 と同様に示せる．■

StkTT のスタック木変換の意味論から，木変換の意味論を導入できる．

定義 200 (StkTT の意味論). StkTT M について，$[\![M]\!]$ を $[\![M]\!]_\omega(-)(0)$ で定義する．また，木変換のクラス $\{[\![M]\!] \mid M$ は StkTT$\}$ を StkTT と表記する．□

なお，スタックのどの位置で木変換を構築しても，それは StkTT の木変換のクラスに含まれる．

定理 201. StkTT M，$n \in \mathbb{N}$ について，以下が成り立つ．

$$[\![M]\!]_\omega(-)(n) \in \mathrm{StkTT}$$

□

証明. StkTT $M = (Q, \Sigma, \Delta, e, R)$ について，StkTT $M_n = (Q, \Sigma, \Delta, \mathrm{Tail}^n(e), R)$ とすると，

$$[\![M]\!]_\omega(-)(n) = [\![M_n]\!]$$

を満たすことは，容易に確かめられる． ■

TDTT は，深さが常にひとつのスタックを使用する StkTT と見なすことができる．ここから，$\mathrm{TDTT}_\omega \subseteq \mathrm{StkTT}_\omega$ となる．これは，定理 186 から，つまり TDTT $\subseteq$ StkTT ということである．

定理 202. $\mathrm{TDTT}_\omega \subseteq \mathrm{StkTT}_\omega$ □

証明. TDTT $M = (Q, \Sigma, \Delta, e, R)$ について，StkTT $M' = (Q, \Sigma, \Delta, e', R')$ を以下のように定義する．

$$\begin{aligned} e' &\stackrel{\mathrm{def}}{=} \mathrm{stkproj}_{\mathrm{stk}}(e) \\ R' &\stackrel{\mathrm{def}}{=} \{q(\sigma(\vec{\mathrm{x}})) \to \mathrm{stkproj}_{\mathrm{stk}}(\eta) \mid q(\sigma(\vec{\mathrm{x}})) \to \eta \in R\} \end{aligned}$$

この時，$[\![M]\!]_\omega = [\![M']\!]_\omega$ を示す．任意の $q \in Q$，$t \in \hat{T}_\Sigma$ について，

$$\begin{aligned} q(t) \Rightarrow_M \eta \text{ implies } \mathrm{Head}(q(t)) &\Rightarrow_{M'} \mathrm{Head}(\mathrm{stkproj}_{\mathrm{stk}}(\eta)) \\ &\Rightarrow^*_{M'} \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{Head}(\mathrm{stkproj}_{\mathrm{stk}}(\eta))) \\ &= \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(\eta)) \end{aligned}$$

であり，また，$U = \{\mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(\eta)) \mid \eta \in \mathrm{SF}_M(\emptyset)\}$ は以下のような帰納的構造を持つことが容易に示せる．

IB1　$q \in Q$，$t \in \hat{T}_\Sigma$ について，$\mathrm{Head}(q(t)) \in U$
IB2　$\delta \in \Delta^{(0)}$ について，$\delta \in U$
IS　$n \geq 1$，$\delta \in \Delta^{(n)}$，$\eta_1, \ldots, \eta_n \in U$ について，$\delta(\eta_1, \ldots, \eta_n)$

よって，

$$\eta_1 \Rightarrow_M \eta_2 \text{ implies } \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(\eta_1)) \Rightarrow^*_{M'} \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(\eta_2))$$

となることも容易に示せる．さらに，

$$\mathrm{nf}(\Rightarrow_M, \eta) = \mathrm{nf}(\Rightarrow_{M'}, \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(\eta)))$$

となることも容易に示せるため，

$$\begin{aligned}
[\![M]\!]_\omega(t)(0) &= \mathrm{nf}(\Rightarrow_M, e[\mathrm{x}_1 \leftarrow t]) \\
&= \mathrm{nf}(\Rightarrow_{M'}, \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(e[\mathrm{x}_1 \leftarrow t]))) \\
&= \mathrm{nf}(\Rightarrow_{M'}, \mathrm{stkproj}_{\mathrm{elm}}(e[\mathrm{x}_1 \leftarrow t])) \\
&= \mathrm{nf}(\Rightarrow_{M'}, \mathrm{Head}(\mathrm{stkproj}_{\mathrm{stk}}(e[\mathrm{x}_1 \leftarrow t]))) \\
&= \mathrm{nf}(\Rightarrow_{M'}, \mathrm{Head}(e'[\mathrm{x}_1 \leftarrow t])) \\
&= [\![M']\!]_\omega(t)(0)
\end{aligned}$$

となる．また，$n > 0$ について，$[\![M]\!]_\omega(t)(n) = \bot = [\![M']\!]_\omega(t)(n)$ は容易に示せる．よって，題意は示された． ■

StkTT で表現できる木変換として，入力木が $\mathrm{a}^n(\mathrm{b}^n(\mathrm{c}^n(\$)))$ の形をしているかを判定するものがある．これは独立に例 189 のように 2 つの文字の数が一致するかを判定し，その結果を合わせることで実現される．

例 203. StkTT $M = (Q, \Sigma, \Delta, e, R)$ を，以下のように定義する．

$$\begin{aligned}
Q &\stackrel{\mathrm{def}}{=} \{q_1, q_2, q_1', q_2', q_3'\} \\
\Sigma &\stackrel{\mathrm{def}}{=} \{\mathrm{a}^{(1)}, \mathrm{b}^{(1)}, \mathrm{c}^{(1)}, \$^{(0)}\} \\
\Delta &\stackrel{\mathrm{def}}{=} \{\wedge^{(2)}, \top^{(0)}\} \\
e &\stackrel{\mathrm{def}}{=} \mathrm{Cons}(\wedge(\mathrm{Head}(q_1(\mathrm{x}_1)), \mathrm{Head}(q_1'(\mathrm{x}_1))), \mathrm{Empty}) \\
R &\stackrel{\mathrm{def}}{=} \{q_1(\mathrm{a}(\mathrm{x}_1)) \to \mathrm{Tail}(q_1(\mathrm{x}_1)), \\
&\qquad q_1(\mathrm{b}(\mathrm{x}_1)) \to \mathrm{Cons}(\mathrm{F}, q_2(\mathrm{x}_1)), \\
&\qquad q_1(\$) \to \mathrm{Cons}(\mathrm{T}, \mathrm{Empty}), \\
&\qquad q_2(\mathrm{b}(\mathrm{x}_1)) \to \mathrm{Cons}(\mathrm{F}, q_2(\mathrm{x}_1)), \\
&\qquad q_2(\mathrm{c}(\mathrm{x}_1)) \to \mathrm{Cons}(\mathrm{T}, \mathrm{Empty}), \\
&\qquad q_1'(\mathrm{a}(\mathrm{x}_1)) \to q_1'(\mathrm{x}_1), \\
&\qquad q_1'(\mathrm{b}(\mathrm{x}_1)) \to \mathrm{Tail}(q_2'(\mathrm{x}_1)), \\
&\qquad q_2'(\mathrm{b}(\mathrm{x}_1)) \to \mathrm{Tail}(q_2'(\mathrm{x}_1)), \\
&\qquad q_2'(\mathrm{c}(\mathrm{x}_1)) \to \mathrm{Cons}(\mathrm{F}, q_3'(\mathrm{x}_1)), \\
&\qquad q_2'(\$) \to \mathrm{Cons}(\mathrm{T}, \mathrm{Empty}), \\
&\qquad q_3'(\mathrm{c}(\mathrm{x}_1)) \to \mathrm{Cons}(\mathrm{F}, q_3'(\mathrm{x}_1)), \\
&\qquad q_3'(\$) \to \mathrm{Cons}(\mathrm{T}, \mathrm{Empty}), \\
&\qquad q_1(\mathrm{c}(\mathrm{x}_1)) \to \mathrm{Empty},
\end{aligned}$$

$$q_2(\mathrm{a}(\mathrm{x}_1)) \to \mathrm{Empty},$$
$$q_2(\$) \to \mathrm{Empty},$$
$$q_1'(\mathrm{c}(\mathrm{x}_1)) \to \mathrm{Empty},$$
$$q_1'(\$) \to \mathrm{Empty},$$
$$q_2'(\mathrm{a}(\mathrm{x}_1)) \to \mathrm{Empty},$$
$$q_3'(\mathrm{a}(\mathrm{x}_1)) \to \mathrm{Empty},$$
$$q_3'(\mathrm{b}(\mathrm{x}_1)) \to \mathrm{Empty}\}$$

□

この木変換は，RTL の逆像が CFTL にならない例である．

命題 204. 例 203 の M について，以下が成り立つ．

$$[\![M]\!]^{-1}[\{\wedge(\top,\top)\}] = \{\mathrm{a}^n(\mathrm{b}^n(\mathrm{c}^n(\$))) \mid n \in \mathbb{N}\}$$

□

証明. $M = (Q, \Sigma, \Delta, e, R)$，$Q = \{q_1, q_2, q_1', q_2', q_3'\}$ について，

$$\forall m \in \mathbb{N}, t \in \hat{T}_\Sigma, \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^m(q_2(t)))) = \top \,.\, \exists t' \in \hat{T}_\Sigma \,.\, t = \mathrm{b}^m(\mathrm{c}(t'))$$

は，t に関する構造的帰納法で容易に示せる．ここから，

$$\forall t \in \hat{T}_\Sigma, \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(q_1(t))) = \top \,.\, \exists n \in \mathbb{N}, t' \in \hat{T}_\Sigma \,.\, t = \begin{cases} \$ & (n=0) \\ \mathrm{a}^n(\mathrm{b}^n(\mathrm{c}(t'))) & (n>0) \end{cases}$$

は，t に関する構造的帰納法で容易に示せる．逆に任意の $n > 0$，$t' \in \hat{T}_\Sigma$ について，

$$\begin{aligned} \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(q_1(\mathrm{a}^n(\mathrm{b}^n(\mathrm{c}(t')))))) &= \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^n(q_2(\mathrm{b}^n(\mathrm{c}(t')))))) \\ &= \top \end{aligned}$$

であることも，$n-1$ に関する容易な数学的帰納法から示せる．

$$\forall t \in \hat{T}_\Sigma, \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(q_1'(t))) = \top \,.\, \exists n, m \in \mathbb{N} \,.\, t = \mathrm{a}^m(\mathrm{b}^n(\mathrm{c}^n(\$)))$$

であること，また，任意の $n, m \in \mathbb{N}$ について，

$$\begin{aligned} \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(q_1'(\mathrm{a}^m(\mathrm{b}^n(\mathrm{c}^n(\$)))))) &= \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(q_2'(\mathrm{b}^n(\mathrm{c}^n(\$))))) \\ &= \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^n(q_3'(\mathrm{c}^n(\$))))) \\ &= \top \end{aligned}$$

も同様に示せる．よって，

$$[\![M]\!]^{-1}[\{\wedge(\top,\top)\}] = \{t \in \hat{T}_\Sigma \mid \mathrm{nf}(\Rightarrow_M, \wedge(q_1(t), q_1'(t))) = \wedge(\top,\top)\}$$

$$
\begin{aligned}
&= \{t \in \hat{T}_\Sigma \mid \mathrm{nf}(\Rightarrow_M, q_1(t)) = \mathrm{nf}(\Rightarrow_M, q'_1(t)) = \top\} \\
&= (\{\mathrm{a}^n(\mathrm{b}^n(\mathrm{c}(t'))) \mid n > 0, t' \in \hat{T}_\Sigma\} \cup \{\$\}) \cap \{\mathrm{a}^m(\mathrm{b}^n(\mathrm{c}^n(\$))) \mid m, n \in \mathbb{N}\} \\
&= \{\mathrm{a}^n(\mathrm{b}^n(\mathrm{c}^n(\$))) \mid n \in \mathbb{N}\}
\end{aligned}
$$

である．■

StkTT の性質として，入力木を固定すると，深さが有限のスタックしか構築できないというものがある．これは，後ほどスタック木変換器の表現力の比較を行う際，重要になる．

定理 205. $[\![M]\!]_\omega : \hat{T}_\Sigma \to (\mathbb{N} \to \hat{T}_{\Delta_\perp}) \in \mathrm{StkTT}_\omega$ について，以下が成り立つ．

$$\forall t \in \hat{T}_\Sigma \,.\, \exists c_{M,t} \in \mathbb{N} \,.\, \forall n \geq c_{M,t} \,.\, [\![M]\!]_\omega(t)(n) = \perp$$

□

証明. StkTT $M = (Q, \Sigma, \Delta, e, R)$，$t \in \hat{T}_\Sigma$ について，$\eta = \mathrm{nf}(\Rightarrow_M, e[\mathrm{x}_1 \leftarrow t])$ とおくと，

$$[\![M]\!]_\omega(t)(n) = \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^n(e[\mathrm{x}_1 \leftarrow t]))) = \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^n(\eta)))$$

である．また，系 193 より $\eta = \mathrm{stkeval}_{\mathrm{stk}}(\eta)$ である．この時，

$$\forall n \geq \mathrm{avstk}(\eta) \,.\, \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^n(\eta))) = \perp$$

を満たす，$\mathrm{avstk}(\eta)$ を，η に関するスタックシステムの標準形としての構造的帰納法で，以下のように定義する．

IB1　$\mathrm{avstk}(\mathrm{Empty}) \stackrel{\mathrm{def}}{=} 0$ とおく．この時，条件を満たすことは，数学的帰納法で容易に確認できる．

IB2, IB3, IB4　適用不能.

IS1　$\mathrm{avstk}(\mathrm{Cons}(\eta_1, \eta_2)) \stackrel{\mathrm{def}}{=} 1 + \mathrm{avstk}(\eta_2)$ とおく．この時，$n \geq 1 + \mathrm{avstk}(\eta_2)$ について，$\mathrm{Head}(\mathrm{Tail}^n(\mathrm{Cons}(\eta_1, \eta_2))) = \mathrm{Head}(\mathrm{Tail}^{n-1}(\eta_2))$ より，i.h. から条件を満たす．

IS2　$m \in \mathbb{N}$，$q \in Q$ について，$\eta = \mathrm{Tail}^m(q(t))$ の時 η は可約だが，これは η が正規形であることに矛盾する．よって，この場合は適用不能である．

IS3, IS4　適用不能.

ここから，$c_{M,t} \stackrel{\mathrm{def}}{=} \mathrm{avstk}(\eta)$ とおくことで，題意は示された．■

StkTT の簡約において，スタックシステム評価の部分だけを切り離したものを，Stk として導入する． Stk は，後ほどスタック木変換器と木変換器の比較を行う際，重要になる．

定義 206 (スタックシステム評価器 (Stk)). ランク付きアルファベット Σ について，StkTT $\mathrm{Stk}_\Sigma = (Q, \Sigma \cup \bar{\mathrm{STK}}, \Sigma, e, R)$ を以下のように定義する．

$$Q \stackrel{\mathrm{def}}{=} \{q^{(1)}\}$$

$$e \stackrel{\text{def}}{=} q(\mathrm{x}_1)$$
$$R \stackrel{\text{def}}{=} \{q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n)) \to \mathrm{stkproj}_{\mathrm{stk}}(\sigma(q(\mathrm{x}_1), \ldots, q(\mathrm{x}_n))) \mid \sigma^{(n)} \in \Sigma \cup \mathrm{S\bar{T}K}\}$$

この時，スタック木変換のクラス $\{[\![\mathrm{Stk}_\Sigma]\!]_\omega \mid$ ランク付きアルファベット $\Sigma\}$ を Stk_ω，木変換のクラス $\{[\![\mathrm{Stk}_\Sigma]\!] \mid$ ランク付きアルファベット $\Sigma\}$ を Stk と表記する． □

Stk は単なる StkTT であるため，そのクラスは StkTT のクラスに含まれる．

系 207. $\mathrm{Stk}_\omega \subseteq \mathrm{StkTT}_\omega$ □

証明. 定義より明らか. ■

また，Stk は，stkeval と stkproj を用いて記述できる．これを示すため，まず以下の補題を示す．

補題 208. $\forall t \in \hat{T}_{\Sigma \cup \mathrm{S\bar{T}K}}$. $\mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(\mathrm{Head}(t))) = \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(t))$ □

証明. 以下のように構造的帰納法で示す．なお，スペースの都合上 stkeval を f，$\mathrm{stkproj}_{\mathrm{elm}}$ を p_1，$\mathrm{stkproj}_{\mathrm{stk}}$ を p_2 と略記する．

IB1　適用不能

IB2　以下のように場合分けを行う．

- 任意の $\sigma \in \Sigma^{(0)} \cup \{\bot\}$ について，

$$\begin{aligned} f(p_1(\mathrm{Head}(\sigma))) &= f(\mathrm{Head}(\mathrm{Cons}(\sigma, \mathrm{Empty}))) \\ &= f(\sigma) \\ &= f(p_1(\sigma)) \end{aligned}$$

より正しい．

- $t = \mathrm{Empty}$ の場合，

$$\begin{aligned} f(p_1(\mathrm{Head}(\mathrm{Empty}))) &= f(\mathrm{Head}(\mathrm{Empty})) \\ &= f(p_1(\mathrm{Empty})) \end{aligned}$$

より正しい．

IS　以下のように場合分けを行う．

- 任意の $n \geq 1$，$\sigma^{(n)} \in \Sigma$，$t_1, \ldots, t_n \in \hat{T}_{\Sigma \cup \mathrm{S\bar{T}K}}$ について，

$$\begin{aligned} f(p_1(\mathrm{Head}(\sigma(t_1, \ldots, t_n)))) &= f(\mathrm{Head}(\mathrm{Cons}(\sigma(p_1(t_1), \ldots, p_1(t_n)), \mathrm{Empty}))) \\ &= f(\sigma(p_1(t_1), \ldots, p_1(t_n))) \\ &= f(p_1(\sigma(t_1, \ldots, t_n))) \end{aligned}$$

より正しい．

- $t_1, t_2 \in \hat{T}_{\Sigma \cup \mathrm{S\bar{T}K}}$ について，

$$\begin{aligned} f(p_1(\mathrm{Head}(\mathrm{Cons}(t_1, t_2)))) &= f(\mathrm{Head}(\mathrm{Cons}(p_1(t_1), p_2(t_2)))) \\ &= f(p_1(\mathrm{Cons}(t_1, t_2))) \end{aligned}$$

より正しい．

- $t_1 \in \hat{T}_{\Sigma \cup \mathrm{S\bar{T}K}}$ について，

$$\begin{aligned} f(p_1(\mathrm{Head}(\mathrm{Head}(t_1)))) &= f(\mathrm{Head}(\mathrm{Cons}(\mathrm{Head}(p_2(t_1)), \mathrm{Empty}))) \\ &= f(\mathrm{Head}(p_2(t_1))) \\ &= f(p_1(\mathrm{Head}(t_1))) \end{aligned}$$

- $t_1 \in \hat{T}_{\Sigma \cup \mathrm{S\bar{T}K}}$ について，

$$\begin{aligned} f(p_1(\mathrm{Head}(\mathrm{Tail}(t_1)))) &= f(\mathrm{Head}(\mathrm{Tail}(p_2(t_1)))) \\ &= f(p_1(\mathrm{Tail}(t_1))) \end{aligned}$$

■

ここから，Stk は stkproj で射影した後 stkeval で評価するのと同じ意味論を持つ．

定理 209. $[\![M]\!] : \hat{T}_{\Sigma \cup \mathrm{S\bar{T}K}} \to \hat{T}_{\Sigma_\perp} \in \mathrm{Stk}$ について，以下が成り立つ．

$$\forall t \in \hat{T}_{\Sigma \cup \mathrm{S\bar{T}K}} \,.\, [\![M]\!](t) = \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(t))$$

□

証明. $\mathrm{Stk}_\Sigma = M = (Q, \Sigma_+ = \Sigma \cup \mathrm{S\bar{T}K}, \Sigma, e, R)$，$Q = \{q\}$，$t \in \hat{T}_{\Sigma_+}$，$m \in \mathbb{N}$ について，$\mathrm{ev} = \mathrm{stkeval}_{\mathrm{elm}} \circ \mathrm{stkproj}_{\mathrm{elm}}$ として，

$$\mathrm{Head}(\mathrm{Tail}^m(q(t))) \Rightarrow_M^* \mathrm{ev}(\mathrm{Head}(\mathrm{Tail}^m(t)))$$

を，t に関する構造的帰納法で示す．なお，スペースの都合上 $\mathrm{stkeval}_{\mathrm{elm}}$ を f，$\mathrm{stkproj}_{\mathrm{elm}}$ を p_1，$\mathrm{stkproj}_{\mathrm{stk}}$ を p_2，$t \mapsto \mathrm{Head}(\mathrm{Tail}^m(t))$ を g_m と略記する．

IB1　適用不能．

IB2　以下のように場合分けを行う．

- $\sigma \in \Sigma^{(0)} \cup \{\perp\}$ について，

$$\begin{aligned} g_m(q(\sigma)) &\Rightarrow_M g_m(\mathrm{stkproj}_{\mathrm{stk}}(\sigma)) \\ &= g_m(\mathrm{Cons}(\sigma, \mathrm{Empty})) \\ &= p_1(g_m(\sigma)) \\ &\Rightarrow_M^* \mathrm{ev}(g_m(\sigma)) \end{aligned}$$

より正しい．

- $\sigma = \mathrm{Empty}$ について，

$$\begin{aligned} g_m(q(\mathrm{Empty})) &\Rightarrow_M g_m(\mathrm{stkproj}_{\mathrm{stk}}(\mathrm{Empty})) \\ &= g_m(\mathrm{Empty}) \\ &= p_1(g_m(\mathrm{Empty})) \\ &\Rightarrow_M^* \mathrm{ev}(g_m(\mathrm{Empty})) \end{aligned}$$

より正しい．

IS　以下のように場合分けを行う．

- $n \geq 1$，$\sigma^{(n)} \in \Sigma$，$t_1, \ldots, t_n \in \hat{T}_{\Sigma_+}$ について，

$$\begin{aligned} g_m(q(\sigma(t_1, \ldots, t_n))) &\Rightarrow_M g_m(\mathrm{stkproj}_{\mathrm{stk}}(\sigma(q(t_1), \ldots, q(t_n)))) \\ &= g_m(\mathrm{Cons}(\sigma(g_0(q(t_1)), \ldots, g_0(q(t_n))), \mathrm{Empty})) \\ &\Rightarrow_M^* g_m(\mathrm{Cons}(\sigma(\mathrm{ev}(g_0(t_1)), \ldots, \mathrm{ev}(g_0(t_n))), \mathrm{Empty})) \\ &\quad (\because \text{i.h.}) \\ &= g_m(\mathrm{Cons}(\sigma(\mathrm{ev}(t_1), \ldots, \mathrm{ev}(t_n)), \mathrm{Empty})) \\ &= p_1(g_m(\sigma(\mathrm{ev}(t_1), \ldots, \mathrm{ev}(t_n)))) \\ &\Rightarrow_M^* \mathrm{ev}(g_m(\sigma(\mathrm{ev}(t_1), \ldots, \mathrm{ev}(t_n)))) \\ &= \mathrm{ev}(g_m(\sigma(t_1, \ldots, t_n))) \end{aligned}$$

より正しい．

- $\sigma = \mathrm{Head}$，$t_1 \in \hat{T}_{\Sigma_+}$ について，

$$\begin{aligned} g_m(q(\mathrm{Head}(t_1))) &\Rightarrow_M g_m(\mathrm{stkproj}_{\mathrm{stk}}(\mathrm{Head}(q(t_1)))) \\ &= g_m(\mathrm{Cons}(\mathrm{Head}(q(t_1)), \mathrm{Empty})) \\ &\Rightarrow_M \begin{cases} g_0(q(t_1)) & (m = 0) \\ g_{m-1}(\mathrm{Empty}) & (m > 0) \end{cases} \\ &\Rightarrow_M^* \begin{cases} \mathrm{ev}(g_0(t_1)) & (m = 0) \\ \mathrm{Empty} & (m > 0) \end{cases} && (\because \text{i.h.}) \\ &= \begin{cases} \mathrm{ev}(g_0(\mathrm{Head}(t_1))) & (m = 0) \\ \mathrm{ev}(g_m(\mathrm{Head}(t_1))) & (m > 0) \end{cases} && (\because \text{補題 208}) \\ &= \mathrm{ev}(g_m(\mathrm{Head}(t_1))) \end{aligned}$$

より正しい．

- $\sigma = \mathrm{Tail}$，$t_1 \in \hat{T}_{\Sigma_+}$ について，

$$\begin{aligned} g_m(q(\mathrm{Tail}(t_1))) &\Rightarrow_M g_m(\mathrm{stkproj}_{\mathrm{stk}}(\mathrm{Tail}(q(t_1)))) \\ &= g_m(\mathrm{Tail}(q(t_1))) \\ &= g_{m+1}(q(t_1)) \\ &\Rightarrow_M^* \mathrm{ev}(g_{m+1}(t_1)) && (\because \text{i.h.}) \\ &= \mathrm{ev}(g_m(\mathrm{Tail}(t_1))) \end{aligned}$$

より正しい.

- $\sigma = \mathrm{Cons}$，$t_1, t_2 \in \hat{T}_{\Sigma_+}$ について，

$$\begin{aligned} g_m(q(\mathrm{Cons}(t_1,t_2))) &\Rightarrow_M g_m(\mathrm{stkproj}_{\mathrm{stk}}(\mathrm{Cons}(q(t_1),q(t_2)))) \\ &= g_m(\mathrm{Cons}(\mathrm{Head}(q(t_1)),q(t_2))) \\ &\Rightarrow_M \begin{cases} g_0(q(t_1)) & (m=0) \\ g_{m-1}(q(t_2)) & (m>0) \end{cases} \\ &\Rightarrow_M^* \begin{cases} \mathrm{ev}(g_0(t_1)) & (m=0) \\ \mathrm{ev}(g_{m-1}(t_2)) & (m>0) \end{cases} \qquad (\because \text{i.h.}) \\ &= \mathrm{ev}(g_m(\mathrm{Cons}(t_1,t_2))) \end{aligned}$$

より正しい.

また，$\mathrm{nf}(\Rightarrow_M, \mathrm{ev}(g_m(t))) = \mathrm{ev}(g_m(t))$ は，容易に確かめられる．よって，

$$\begin{aligned} [\![M]\!](t) &= \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(q(t))) \\ &= \mathrm{nf}(\Rightarrow_M, \mathrm{ev}(\mathrm{Head}(t))) \\ &= \mathrm{ev}(\mathrm{Head}(t)) \\ &= \mathrm{ev}(t) \qquad (\because \text{補題 208}) \end{aligned}$$

となる． ■

なお，補題 208 からただちに Stk の評価に Head は影響しないことが分かる．

系 210. $\forall t \in \hat{T}_{\Sigma \cup \mathrm{S\bar{T}K}} \,.\, [\![\mathrm{Stk}_\Sigma]\!](\mathrm{Head}(t)) = [\![\mathrm{Stk}_\Sigma]\!](t)$ □

証明. 補題 208，定理 209 より． ■

線形 TDTT をスタック木変換器に拡張したものとして，線形 StkTT を導入する．

定義 211 (線形 StkTT). StkTT $M = (Q, \Sigma, \Delta, e, R)$ が線形とは，以下を満たすことを言う．

- $|\mathrm{occ}_e(\mathrm{x}_1)| \leq 1$
- $\forall q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n)) \to \eta \in R, i \in [n] \,.\, |\mathrm{occ}_\eta(\mathrm{x}_i)| \leq 1$

この時，スタック木変換のクラス $\{[\![M]\!]_\omega \mid M$ は線形 StkTT$\}$ を $\mathrm{StkTT}_{\mathrm{lu},\omega}$，木変換のクラス $\{[\![M]\!] \mid M$ は線形 StkTT$\}$ を $\mathrm{StkTT}_{\mathrm{lu}}$ と表記する． □

例 189 は，線形 StkTT の例になっている．また，例 189 は RTL の逆像が RTL に閉じていない例になっている．

命題 212. 例 189 の M について，以下が成り立つ．

$$[\![M]\!]^{-1}[\{\top\}] = \{\mathrm{a}^n(\mathrm{b}^n(\$)) \mid n \in \mathbb{N}\}$$

□

証明. 命題 204 と同様に示せる. ■

4.3 スタック属性付き木変換器

ATT を拡張したスタック木変換器を，スタック属性付き木変換器 (StkATT) と呼ぶ．StkATT も StkTT と同じく，規則の右辺を，ランク付き木ではなく，スタックシステムで構成させる．

定義 213 (スタック属性付き木変換器 (StkATT)). スタック属性付き木変換器は，以下の要素の組 $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ である．

$A = \{A_k\}_{k \in \mathrm{AS}}$　互いに素な属性記号のランク付きアルファベットの族で，$A = \{A_k^{(1)}\}_{k \in \mathrm{AS}}$．
　　なお，$a \in A_{\mathrm{syn}}$ を合成属性，$b \in A_{\mathrm{inh}}$ を継承属性と呼ぶ．

Σ　　入力記号のランク付きアルファベット．

Δ　　出力記号のランク付きアルファベットで，$\bot \notin \Delta$ ．

$\sharp \notin \Sigma$　ルート記号のランク付き記号．$\Sigma_\sharp = \Sigma \cup \{\sharp^{(1)}\}$ と表記する．

$a_0 \in A_{\mathrm{syn}}$　初期合成属性．

R　　書き換え規則の集合で，以下を満たす．

$$R \in \mathcal{P}_{fin}(\{\varphi \xrightarrow{\sigma} \eta \mid \sigma^{(n)} \in \Sigma_\sharp, \varphi \in \mathrm{GLHS}_{M,\sigma}([n]), \eta \in \mathrm{GRHS}_{M,\sigma}(\{\mathrm{w}\}, [n])\})$$

ただし，

$$\mathrm{GLHS}_{M,\sigma}(I) \stackrel{\mathrm{def}}{=} \{\varphi \in \mathrm{LHS}_M(I) \mid \sigma \neq \sharp \vee \forall a \in A_{\mathrm{syn}} \setminus \{a_0\} \,.\, \mathrm{occ}_\varphi(a) = \emptyset\}$$
$$\mathrm{GRHS}_{M,\sigma}(P, I) \stackrel{\mathrm{def}}{=} \{\eta \in \mathrm{RHS}_M(P, I)_{\mathrm{stk}} \mid \sigma \neq \sharp \vee \forall b \in A_{\mathrm{inh}} \,.\, \mathrm{occ}_\eta(b) = \emptyset\}$$
$$\mathrm{LHS}_M(I) \stackrel{\mathrm{def}}{=} \{a(\mathrm{w}) \mid a \in A_{\mathrm{syn}}\} \cup \{b(\mathrm{w}i) \mid b \in A_{\mathrm{inh}}, i \in I\}$$

で，$\mathrm{RHS}_M(P, I) \in \prod_{k \in \mathrm{SS}} \mathcal{P}(\mathcal{S}_{\dot{\Delta} \cup \bigcup_{k \in \mathrm{AS}} \ddot{A_k}}(\{\mathrm{elm} \mapsto P \cup \{\pi i \mid \pi \in P, i \in I\}\})_k)$ は，以下のように帰納的に定義される集合の族 $\{U_k\}_{k \in \mathrm{SS}}$ で定義する．

IB1　$a : \mathrm{elm} \twoheadrightarrow_{\ddot{A}_{\mathrm{syn}}} k$，$\pi \in P$，$i \in I$ について，$a(\pi i) \in U_k$．

IB2　$b : \mathrm{elm} \twoheadrightarrow_{\ddot{A}_{\mathrm{inh}}} k$，$\pi \in P$ について，$b(\pi) \in U_k$．

IB3　$\delta : () \twoheadrightarrow_{\dot{\Delta} \cup \mathrm{STK}} k$ について，$\delta \in U_k$．

IS　$n \geq 1$，$\delta : (k_1, \ldots, k_n) \twoheadrightarrow_{\dot{\Delta} \cup \mathrm{STK}} k$，$\eta_1 \in U_{k_1}, \ldots, \eta_n \in U_{k_n}$ について，$\delta(\eta_1, \ldots, \eta_n) \in U_k$．

□

ATT と同様，StkATT でも全域的決定的性を定義できる．

定義 214 (全域的決定的 StkATT). StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ が

$$\forall \sigma^{(n)} \in \Sigma_\sharp, \varphi \in \mathrm{GLHS}_{M,\sigma}([n])\ .\ \exists! \varphi \xrightarrow{\sigma} \eta \in R$$

を満たすとき，全域的決定的であるという．□

以降，特に断りのない限り StkATT は全域的決定的であるものとする．StkATT について，意味論を決定づける簡約システムを導入する．この簡約システムは，木変換器や他のスタック木変換器と異なり，最外戦略によって定義される．これは，StkATT に戦略によらない自由な簡約を許すと，その簡約システムが停止性を持たなくなるからである．この性質は，StkATT に対し，他のスタック木変換器と異なる特殊な性質をもたらす．この性質についての詳細は，第 5 章で述べる．

定義 215 (StkATT の簡約システム). StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_{\Sigma_\sharp}$ について，$\mathrm{SF}_M(t) \stackrel{\mathrm{def}}{=} \mathrm{RHS}_M(\mathrm{paths}(t), \mathbb{N})$ とする．この時，簡約システム $(\bigcup_{k \in \mathrm{SS}} \mathrm{SF}_M(t)_k, \Rightarrow_{M,t})$ を，$U \stackrel{\mathrm{def}}{=} \mathrm{SF}_M(t)$ として，以下のように構造的帰納法で定義する．

IB1　$a \in A_{\mathrm{syn}}$，$\sigma \in \Sigma_\sharp$，$p \in \mathrm{paths}(t)$，$\mathrm{label}_t(p) = \sigma$，$a(\mathrm{w}) \xrightarrow{\sigma} \eta \in R$ について，

$$a(p) \Rightarrow_{M,t} \eta[\mathrm{w} \leftarrow p].$$

IB2　$b \in A_{\mathrm{inh}}$，$\sigma \in \Sigma_\sharp$，$p \in \mathrm{paths}(t)$，$\mathrm{label}_t(p) = \sigma$，$b(\mathrm{w}i) \xrightarrow{\sigma} \eta \in R$ について，

$$b(pi) \Rightarrow_{M,t} \eta[\mathrm{w} \leftarrow p].$$

IB3　$\delta : () \rightarrowtriangle_{\dot{\Delta} \cup \mathrm{STK}} s$ について，δ は既約．

IS　以下のように場合分けを行う．

- $\eta_1 \in U_{\mathrm{elm}}$，$\eta_2 \in U_{\mathrm{stk}}$ について，$\mathrm{Head}(\mathrm{Cons}(\eta_1, \eta_2)) \Rightarrow_{M,t} \eta_1$．
- $\eta_1 \in U_{\mathrm{elm}}$，$\eta_2 \in U_{\mathrm{stk}}$ について，$\mathrm{Tail}(\mathrm{Cons}(\eta_1, \eta_2)) \Rightarrow_{M,t} \eta_2$．
- $\mathrm{Head}(\mathrm{Empty}) \Rightarrow_{M,t} \bot$．
- $\mathrm{Tail}(\mathrm{Empty}) \Rightarrow_{M,t} \mathrm{Empty}$．
- それ以外の場合，$n \geq 1$，$\delta : (k_1, \ldots, k_n) \rightarrowtriangle_{\dot{\Delta} \cup \mathrm{STK}} k$，$\eta_1 \in U_{k_1}, \ldots, \eta_n \in U_{k_n}$ について，$\eta_i \Rightarrow_{M,t} \eta_i'$ の時，

$$\delta(\eta_1, \ldots, \eta_i, \ldots, \eta_n) \Rightarrow_{M,t} \delta(\eta_1, \ldots, \eta_i', \ldots, \eta_n).$$

□

この簡約システムは，型を保存する．

系 216. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_{\Sigma_\sharp}$，$\eta_1 \Rightarrow_{M,t}^* \eta_2$ について，以下が成り立つ．

$$\eta_1 \in \mathrm{SF}_M(t)_k \implies \eta_2 \in \mathrm{SF}_M(t)_k$$

□

証明. 定義より明らか. ■

また，この簡約システムは，最外戦略によるスタック評価の簡約ができる.

補題 217. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_{\Sigma_\sharp}$ について，以下が成り立つ.

$$\forall \eta_1, \eta_2 \in \mathrm{SF}_M(t)_k \,.\, \eta_1 {\Rightarrow^D_{\mathrm{stkeval}}}^* \eta_2 \text{ implies } \eta_1 \Rightarrow^*_{M,t} \eta_2$$

□

証明. $\eta_1 \Rightarrow^D_{\mathrm{stkeval}} \eta_2$ implies $\eta_1 \Rightarrow_{M,t} \eta_2$ は構造的帰納法から容易に示せる．よって，題意は簡約の長さに関する数学的帰納法より容易に示せる. ■

ここから，スタックシステムの弱標準形，標準形への簡約ができる.

系 218. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_{\Sigma_\sharp}$，$k \in \mathrm{SS}$ について，以下が成り立つ.

- $\forall \eta \in \mathrm{SF}_M(t)_k \,.\, \eta \Rightarrow^*_{M,t} \mathrm{stkwkeval}_k(\eta)$
- $\forall \eta \in \mathrm{SF}_M(t)_k \,.\, \eta \Rightarrow^*_{M,t} \mathrm{stkeval}_k(\eta)$

□

証明. 補題 217，系 177，定理 181 より. ■

さらに，弱標準形までの簡約は決定的であり，最外戦略のスタックシステムの簡約に対応する.

補題 219. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_{\Sigma_\sharp}$，$k \in \mathrm{SS}$，$\eta \in \mathrm{SF}_M(t)_k$ について，以下が成り立つ.

$$\forall n \leq \mathrm{stkwevstep}(\eta), \eta' \in \mathrm{SF}_M(t)_k \,.\, \eta {\Rightarrow^D_{\mathrm{stkeval}}}^n \eta' \text{ iff } \eta \Rightarrow^n_{M,t} \eta'$$

□

証明. $\eta \neq \mathrm{stkwkeval}_k(\eta)$ の時，以下が構造的帰納法から容易に示せる.

$$\forall \eta' \,.\, \eta \Rightarrow^D_{\mathrm{stkeval}} \eta' \text{ iff } \eta \Rightarrow_{M,t} \eta'$$

ここから，題意は n に関する数学的帰納法により容易に示せる. ■

特に，弱標準形までの簡約回数は，スタックシステム弱評価回数と一致する.

系 220. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_{\Sigma_\sharp}$，$k \in \mathrm{SS}$，$\eta \in \mathrm{SF}_M(t)_k$ について，以下が成り立つ.

$$\forall \eta' \in \mathrm{SF}_M(t)_k \,.\, \eta' = \mathrm{stkwkeval}_k(\eta) \text{ iff } \eta \Rightarrow^{\mathrm{stkwevstep}(\eta)}_{M,t} \eta'$$

□

証明. 補題 219 について，$n = \mathrm{stkwevstep}(\eta)$ の時を考える． ■

また，利便性のため，いくつかの表記を導入する．まず，attr を ATT と同様に導入する．また，$\mathrm{Head}(\mathrm{Tail}^m(x))$ の形の要素に対する集合を表すため，htform を導入する．さらに，スタックシステムにおいて，スタック型の要素 η の標準形は $\mathrm{Cons}(\eta_1, \cdots \mathrm{Cons}(\eta_n, \mathrm{Tail}^m(x)) \cdots)$ という形を取る．この Cons の数を $\mathrm{avstk}(\eta)$ で表す．StkATT の右辺に対して avstk を計算した値の最大値を maxavstk で表す．この時，StkATT M は，規則の右辺が全て $\mathrm{Cons}(\eta_1, \cdots \mathrm{Cons}(\eta_{\mathrm{maxavstk}(M)-1}, \mathrm{Tail}^m(x)) \cdots)$ という形になるよう意味論同値な変換を行うことができる．この変換を行った時に，規則の右辺で $\mathrm{Head}(\mathrm{Tail}^m(x))$ という形の出現で最大の m を maxrefstk，右辺 $\mathrm{Cons}(\eta_1, \cdots \mathrm{Cons}(\eta_{\mathrm{maxavstk}(M)-1}, \mathrm{Tail}^m(x)) \cdots)$ において最大の m を maxtailstk で表す．

定義 221. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について，

$$\mathrm{attr}(M,t) \stackrel{\mathrm{def}}{=} \{a(p) \mid a \in \bigcup_{k \in \mathrm{AS}} A_k, p \in \mathrm{paths}(\sharp(t))\}$$

$$\mathrm{htform}(X) \stackrel{\mathrm{def}}{=} X \times \mathbb{N}$$

$$\mathrm{maxavstk}(M) \stackrel{\mathrm{def}}{=} \max\{\mathrm{avstk}(\eta) \mid a(w) \xrightarrow{\sigma} \eta \in R\}$$

$$\mathrm{maxrefstk}(M) \stackrel{\mathrm{def}}{=} \max \left\{ m \in \mathbb{N} \;\middle|\; \begin{array}{l} a(w) \xrightarrow{\sigma} \eta \in R, \\ n \in [\mathrm{maxavstk}(M)], \\ \eta_n = \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{Head}(\mathrm{Tail}^{n-1}(\eta))), \\ a' \in \bigcup_{k \in \mathrm{AS}} A_k, \\ \exists w' \,.\, \mathrm{occ}_{\eta_n}(\mathrm{Head}(\mathrm{Tail}^m(a'(w')))) \neq \emptyset \end{array} \right\}$$

$$\mathrm{maxtailstk}(M) \stackrel{\mathrm{def}}{=} \max \left\{ m \in \mathbb{N} \;\middle|\; \begin{array}{l} a(w) \xrightarrow{\sigma} \eta \in R, \\ n = \mathrm{maxavstk}(M), \\ a' \in \bigcup_{k \in \mathrm{AS}} A_k, \\ \exists w' \,.\, \mathrm{Tail}^m(a'(w')) = \mathrm{stkeval}_{\mathrm{stk}}(\mathrm{Tail}^n(\eta)) \end{array} \right\}$$

とおく．ただし，$\mathrm{avstk}(\eta)$ は，以下のように η の標準形に関する構造的帰納法で定義する．

IB1　$\mathrm{stkeval}_{\mathrm{stk}}(\eta) = \mathrm{Empty}$ の時，$\mathrm{avstk}(\eta) \stackrel{\mathrm{def}}{=} 0$.
IB2, IB3, IB4　適用不能.
IS1　$\mathrm{stkeval}_{\mathrm{stk}}(\eta) = \mathrm{Cons}(\eta_1, \eta_2)$ の時，$\mathrm{avstk}(\eta) \stackrel{\mathrm{def}}{=} 1 + \mathrm{avstk}(\eta_2)$.
IS2　$m \in \mathbb{N}$，$a \in \bigcup_{k \in \mathrm{AS}} A_k$ について，$\mathrm{stkeval}_{\mathrm{stk}}(\eta) = \mathrm{Tail}^m(a(w))$ の時，$\mathrm{avstk}(\eta) \stackrel{\mathrm{def}}{=} 0$.
IS3, IS4　適用不能.

また，$(x, m) \in \mathrm{htform}(X)$ を分かりやすさのため $\mathrm{Head}(\mathrm{Tail}^m(x))$ と表記する． □

一般の StkATT において，簡約システムは停止性を持たない．特に，要素型に限定した場合の簡約システムも停止性を持たない．これは，StkATT が全域的決定的であっても意味論が関数に

ならない場合があることを意味する．そのような場合を除くため，ATT と同様に well-defined の概念を導入する．

定義 222 (StkATT の well-defined 性). StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ が

$$\forall t \in \hat{T}_\Sigma, \mathrm{Head}(\mathrm{Tail}^n(a(p)) \in \mathrm{htform}(\mathrm{attr}(M, t)) \,.\\ \mathrm{Head}(\mathrm{Tail}^n(a(p))) \text{ は } \Rightarrow_{M,\sharp(t)} \text{ で無限簡約可能でない}$$

を満たす時，M は well-defined であるという． □

この well-defined 性は，要素型のみに対し無限簡約可能でないことを要求しており，スタック全体として無限簡約可能でないことは要求していない． well-defined だが，属性のスタック全体については無限簡約可能な StkATT として，次の例が考えられる．

例 223. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ を以下のように定義する．

$$\begin{aligned} A_{\mathrm{syn}} &\stackrel{\mathrm{def}}{=} \{a\} \\ A_{\mathrm{inh}} &\stackrel{\mathrm{def}}{=} \{b\} \\ \Sigma &\stackrel{\mathrm{def}}{=} \{\$^{(0)}\} \\ \Delta &\stackrel{\mathrm{def}}{=} \Sigma \\ a_0 &\stackrel{\mathrm{def}}{=} a \\ R &\stackrel{\mathrm{def}}{=} \{a(\mathrm{w}) \xrightarrow{\sharp} a(\mathrm{w1}), a(\mathrm{w}) \xrightarrow{\$} \mathrm{Cons}(\$, b(\mathrm{w})), b(\mathrm{w1}) \xrightarrow{\sharp} a(\mathrm{w1})\} \end{aligned}$$

□

この例は，全ての位置に $ が積まれたスタックを返すスタック木変換を定義する．スタックのどの位置に対しても簡約は唯一の正規形を持つためこの例は well-defined だが，スタック全体に対しては無限簡約可能である．

命題 224. 例 223 の $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，以下が成り立つ．

$$\forall \mathrm{Head}(\mathrm{Tail}^n(a(p)) \in \mathrm{htform}(\mathrm{attr}(M, \$)) \,.\, \exists \eta \in \{\$\} \cup \{b(\epsilon) \mid b \in A_{\mathrm{inh}}\}\\ \eta = \mathrm{nf}(\Rightarrow_{M,\sharp(\$)}, \mathrm{Head}(\mathrm{Tail}^n(a(p))))$$

□

証明. n に関する容易な数学的帰納法から示せる． ■

StkATT の特徴的な簡約として，Tail-**簡約**がある．Tail-簡約は，その簡約において，Tail が必ず露出しているか，属性が露出しているかのいずれかの状態であり続ける簡約のことである．ある簡約の最初と最後が Tail か属性が露出している状態の場合，その簡約は Tail-簡約である．

補題 225. StkATT $M = (A, \Sigma, \Delta, \sharp, \alpha_0, R)$，$t \in \hat{T}_\Sigma$ について，

- ある $m_0 \in \mathbb{N}$，$a_0(p_0) \in \mathrm{attr}(M, t)$ について，$\eta_0 = \mathrm{Tail}^{m_0}(a_0(p_0))$
- ある $m_n \in \mathbb{N}$，$a_n(p_n) \in \mathrm{attr}(M, t)$ について，$\eta_n = \mathrm{Tail}^{m_n}(a_n(p_n))$
- $\eta_0 \Rightarrow_{M,\sharp(t)} \cdots \Rightarrow_{M,\sharp(t)} \eta_n$

を満たす $\eta_0, \ldots, \eta_n$ を考える．この時，以下が成り立つ．

$$\forall 0 \leq i \leq n \,.\, \exists m_i \in \mathbb{N}, a_i(p_i) \in \mathrm{attr}(M, t) \,.\, \mathrm{stkwkeval}_{\mathrm{stk}}(\eta_i) = \mathrm{Tail}^{m_i}(a_i(p_i))$$

□

証明. $\eta \in \mathrm{SF}_M(t)_{\mathrm{stk}}$ について，

$$\forall m \in \mathbb{N}, a(p) \in \mathrm{attr}(M, t) \,.\, \mathrm{stkwkeval}_{\mathrm{stk}}(\eta) \neq \mathrm{Tail}^{m}(a(p))$$

の時，η は η_n に簡約できないことを，定理 180 より，$\mathrm{stkwkeval}_{\mathrm{stk}}(\eta)$ の場合分けで示す．

C1, C2, C3　適用不能

C4　$\sigma^{(m)} \in \{\mathrm{Cons}, \mathrm{Empty}\}$ について，$\eta = \sigma(\varphi_1, \ldots, \varphi_k)$ となる時，η_n への簡約が存在したとすると，$\eta_n = \sigma(\ldots)$ となることは簡約の長さに関する帰納法で容易に示せる．これは，η_n の条件に矛盾する．よって，正しい．

C5　仮定に矛盾するため，適用不能

C6, C7　適用不能

ここから，$\mathrm{stkwkeval}_{\mathrm{stk}}(\eta_i) = \mathrm{Tail}^{m_i}(a_i(p_i))$ とならない η_i が存在すると仮定すると矛盾．よって，題意は示された．■

また，Tail-簡約において，その簡約列に全て同じ数 Tail を付与しても，それは Tail-簡約の列になる．

補題 226. StkATT $M = (A, \Sigma, \Delta, \sharp, \alpha_0, R)$，$t \in \hat{T}_\Sigma$ について，

$$\mathrm{Tail}^{m_0}(a_0(p_0)) \Rightarrow^{+}_{M,\sharp(t)} \mathrm{Tail}^{m_1}(a_1(p_1)) \Rightarrow^{+}_{M,\sharp(t)} \cdots \Rightarrow^{+}_{M,\sharp(t)} \mathrm{Tail}^{m_n}(a_n(p_n))$$

を満たす $\{\mathrm{Tail}^{m_i}(a_i(p_i))\}_{0 \leq i \leq n}$ を考える．この時，以下が成り立つ．

$$\forall k \in \mathbb{N} \,.\, \mathrm{Tail}^{m_0+k}(a_0(p_0)) \Rightarrow^{+}_{M,\sharp(t)} \mathrm{Tail}^{m_1+k}(a_1(p_1)) \Rightarrow^{+}_{M,\sharp(t)} \cdots \Rightarrow^{+}_{M,\sharp(t)} \mathrm{Tail}^{m_n+k}(a_n(p_n))$$

□

証明. 補題 225 から，容易に示せる．■

StkATT の well-defined 性の同値条件に，**非螺旋性** (*non-spiral*) がある．非螺旋 StkATT は，全てのノードの全ての属性のスタックの全ての位置が簡約結果が，同じノードの同じ属性の

スタックの要素の簡約結果に依存する場合，異なる位置の結果に必ず依存し，しかも位置が上昇していくような Tail-簡約を持たないことを意味する．これは，簡約がスタックのある位置以下で閉じていることを意味する．

定義 227 (StkATT の螺旋性). StkATT $M = (A, \Sigma, \Delta, \sharp, \alpha_0, R)$ について，

- $\exists \eta = \mathrm{stkeval}_{\mathrm{elm}}(\eta), \mathrm{occ}_\eta(\mathrm{Head}(\mathrm{Tail}^m(a(p)))) \neq \emptyset \,.\, \mathrm{Head}(\mathrm{Tail}^m(a(p))) \Rightarrow^+_{M,\sharp(t)} \eta$
- $\exists m' \geq m \,.\, \mathrm{Tail}^m(a(p)) \Rightarrow^+_{M,\sharp(t)} \mathrm{Tail}^{m'}(a(p))$

のいずれかを満たす $t \in \hat{T}_\Sigma$，$a(p) \in \mathrm{attr}(M,t)$，$m \in \mathbb{N}$ が存在する時，M は螺旋であるという．M が螺旋でない時，M は非螺旋であるという． □

これらが同値条件であることを示すために，ATT の依存グラフにスタックの位置も考慮するよう拡張を加えた，StkATT の依存グラフの概念を導入する．

定義 228 (StkATT の依存グラフ). StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，集合の族 $\{\mathrm{attr}_{\mathrm{in}}(M)_\sigma\}_{\sigma \in \Sigma_\sharp}$，$\{\mathrm{attr}_{\mathrm{out}}(M)_\sigma\}_{\sigma \in \Sigma_\sharp}$ を以下のように定義する．

$$\mathrm{attr}_{\mathrm{in}}(M)_{\sigma^{(n)}} \stackrel{\mathrm{def}}{=} \{a(\mathrm{w}) \mid a \in A_{\mathrm{syn}}\} \cup \{b(\mathrm{w}i) \mid b \in A_{\mathrm{inh}}, i \in [n]\}$$
$$\mathrm{attr}_{\mathrm{out}}(M)_{\sigma^{(n)}} \stackrel{\mathrm{def}}{=} \{a(\mathrm{w}i) \mid a \in A_{\mathrm{syn}}, i \in [n]\} \cup \{b(\mathrm{w}) \mid b \in A_{\mathrm{inh}}\}$$

この時，M の依存グラフ $\mathrm{depgraph}(M)$ とは，次のように定義される関係の族 $G \in \prod_{\sigma \in \Sigma} \mathrm{htform}(\mathrm{attr}_{\mathrm{in}}(M)_\sigma) \times \mathrm{htform}(\mathrm{attr}_{\mathrm{out}}(M)_\sigma)$ のことである．

$$\begin{aligned} G_\sigma \stackrel{\mathrm{def}}{=} \{&\langle \mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(w_1))), \mathrm{Head}(\mathrm{Tail}^{m_2}(a_2(w_2)))\rangle \\ &\mid a_1(w_1) \xrightarrow{\sigma} \eta \in R, \eta_{m_1} = \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{Head}(\mathrm{Tail}^{m_1}(\eta))), \\ &\quad \mathrm{occ}_{\eta_{m_1}}(\mathrm{Head}(\mathrm{Tail}^{m_2}(a_2(w_2)))) \neq \emptyset\} \end{aligned}$$

この時，簡約システム $(\mathrm{htform}(\mathrm{attr}(M,t)), \Rightarrow_{G,t})$ を以下のように定義する．

$$\mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(p_1))) \Rightarrow_{G,t} \mathrm{Head}(\mathrm{Tail}^{m_2}(a_2(p_2)))$$
$$\text{iff}$$
$$\left(\begin{array}{l} \exists p \in \mathrm{paths}(\sharp(t)), \sigma = \mathrm{label}_{\sharp(t)}(p), \\ \quad \langle \mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(w_1))), \mathrm{Head}(\mathrm{Tail}^{m_2}(a_2(w_2)))\rangle \in G_\sigma \,. \\ w_1[\mathrm{w} \leftarrow p] = p_1 \wedge w_2[\mathrm{w} \leftarrow p] = p_2 \end{array} \right)$$

□

依存グラフの簡約システムにおいて，属性とパスの組が別の組に簡約できるということは，その組が簡約結果の組に依存しているということを示す．そして，属性とパスの組は，StkATT の簡約システムにおいて依存関係を持つ属性とパスの組が出現する木に簡約できる．

補題 229. StkATT $M = (A, \Sigma, \Delta, \sharp, \alpha_0, R)$，$t \in \hat{T}_\Sigma$，$G = \mathrm{depgraph}(M)$，
$\mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(p_1))), \mathrm{Head}(\mathrm{Tail}^{m_2}(a_2(p_2))) \in \mathrm{htform}(\mathrm{attr}(M,t))$ について，以下の 2 条件は同値である．

- $\mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(p_1))) \Rightarrow^+_{G,t} \mathrm{Head}(\mathrm{Tail}^{m_2}(a_2(p_2)))$
- $\mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(p_1))) \Rightarrow^+_{M,\sharp(t)} \eta$ で，$\mathrm{occ}_{\mathrm{stkeval}_{\mathrm{elm}}(\eta)}(\mathrm{Head}(\mathrm{Tail}^{m_2}(a_2(p_2)))) \neq \emptyset$ を満たす η が存在する．

□

証明. $\mathrm{Head}(\mathrm{Tail}^{m'_1}(a'_1(p'_1))) \Rightarrow_{G,t} \mathrm{Head}(\mathrm{Tail}^{m'_2}(a'_2(p'_2)))$ について，定義より

$$\mathrm{Head}(\mathrm{Tail}^{m'_1}(a'_1(p'_1))) \Rightarrow_{M,\sharp(t)} \eta \Rightarrow^*_{M,\sharp(t)} \mathrm{stkeval}_{\mathrm{elm}}(\eta) = \eta'(\mathrm{Head}(\mathrm{Tail}^{m'_2}(a'_2(p'_2))))$$

と書ける η, η' が存在することは，容易に示せる．よって，$\mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(p_1))) \Rightarrow^+_{G,t}$ $\mathrm{Head}(\mathrm{Tail}^{m_2}(a_2(p_2)))$ について，

$$\begin{aligned}\mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(p_1))) &\Rightarrow^+_{M,\sharp(t)} \eta(\mathrm{Head}(\mathrm{Tail}^{m_2}(a_2(p_2))))\\ &= \mathrm{stkeval}_{\mathrm{elm}}(\eta(\mathrm{Head}(\mathrm{Tail}^{m_2}(a_2(p_2)))))\end{aligned}$$

となる η が存在することは，容易な m についての数学的帰納法で示せる．逆も同様に示せる．
よって，題意は示された． ■

さらに，StkATT M の依存グラフにおいて $\mathrm{maxavstk}(M)$ 以上の位置を参照し続ける依存関係を持つものがある時，それは Tail-簡約と対応する．

補題 230. StkATT $M = (A, \Sigma, \Delta, \sharp, \alpha_0, R)$，$t \in \hat{T}_\Sigma$，$G = \mathrm{depgraph}(M)$，$n > 0$，
$\{m_i\}_{0 \leq i \leq n}$，$\{a_i(p_i)\}_{0 \leq i \leq n}$ について，任意の $0 \leq i \leq n$ で $m_i \geq \mathrm{maxavstk}(M)$ の時，以下の 2 条件は同値である．

- $\mathrm{Head}(\mathrm{Tail}^{m_0}(a_0(p_0))) \Rightarrow_{G,t} \cdots \Rightarrow_{G,t} \mathrm{Head}(\mathrm{Tail}^{m_n}(a_n(p_n)))$
- 以下を満たす $\eta_1, \ldots, \eta_n$，$\eta'_1, \ldots, \eta'_n$ が存在する．
  - $\mathrm{Tail}^{m_0}(a_0(p_0)) \Rightarrow_{M,\sharp(t)} \eta_1 \Rightarrow^*_{M,\sharp(t)} \eta'_1 \Rightarrow_{M,\sharp(t)} \cdots \Rightarrow_{M,\sharp(t)} \eta_n \Rightarrow^*_{M,\sharp(t)} \eta'_n$
  - 任意の $i \in [n]$ について，$\mathrm{stkwkeval}_{\mathrm{stk}}(\eta_i) = \eta'_i = \mathrm{Tail}^{m_i}(a_i(p_i))$

□

証明. 補題 225，$\mathrm{maxavstk}(M)$ の定義より，容易に示せる． ■

ところで，依存グラフにおいて Tail-簡約が存在する条件として以下のものがある．

補題 231. StkATT $M = (A, \Sigma, \Delta, \sharp, \alpha_0, R)$，$t \in \hat{T}_\Sigma$，$G = \mathrm{depgraph}(M)$，$n > 0$，
$a_1(p_1), a_2(p_2) \in \mathrm{attr}(M,t)$，$m_1, m_2 \in \mathbb{N}$ について，以下の 2 条件は同値である．

1. 以下を満たす $n > 0$，$\{\mathrm{Head}(\mathrm{Tail}^{m'_i}(a'_i(p'_i)))\}_{0 \le i \le n}$ が存在する．
   - $a'_0(p'_0) = a_1(p_1)$，$a'_n(p'_n) = a_2(p_2)$
   - $m'_0 - m_1 = m'_1 - m_2$
   - 任意の $0 \le i \le n$ について，$m_i \ge \mathrm{maxavstk}(M)$
   - $\mathrm{Head}(\mathrm{Tail}^{m'_0}(a'_0(p'_0))) \Rightarrow_{G,t} \cdots \Rightarrow_{G,t} \mathrm{Head}(\mathrm{Tail}^{m'_n}(a'_n(p'_n)))$
2. 以下を満たす $k_0 \in \mathbb{N}$ が存在する．

$$\forall k \ge k_0 \,.\, \mathrm{Head}(\mathrm{Tail}^{m_1+k}(a_1(p_1))) \Rightarrow^{+}_{G,t} \mathrm{Head}(\mathrm{Tail}^{m_2+k}(a_2(p_2)))$$

□

証明. 1 の条件を満たす $n > 0$，$\{\mathrm{Head}(\mathrm{Tail}^{m'_i}(a'_i(p'_i)))\}_{0 \le i \le n}$ が存在する時，$k_0 = 0$ とおくと，補題 230，補題 226 より，2 の k_0 の条件を満たすことは明らか．また，逆は k を十分に大きく取ることで容易に示せる．よって，題意は示された． ■

ここから，依存グラフに対し，その非螺旋性を定義できる．

定義 232. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，依存グラフ $G = \mathrm{depgraph}(M)$ が螺旋であるとは，

- $\mathrm{Head}(\mathrm{Tail}^m(a(p))) \Rightarrow^{+}_{G,t} \mathrm{Head}(\mathrm{Tail}^m(a(p)))$
- 以下を満たす $c_m \ge m$ が存在する．

$$\forall k \in \mathbb{N} \,.\, \mathrm{Head}(\mathrm{Tail}^{m+k}(a(p))) \Rightarrow^{+}_{G,t} \mathrm{Head}(\mathrm{Tail}^{c_m+k}(a(p)))$$

のいずれかを満たす $t \in \hat{T}_\Sigma$，$\mathrm{Head}(\mathrm{Tail}^m(a(p))) \in \mathrm{htform}(\mathrm{attr}(M, t))$ が存在することを言う．G が螺旋でない時，G は非螺旋であるという． □

属性とノードの組の数は有限なため，非螺旋 StkATT において，Tail-簡約は，同じ属性とノードの組に行き着く前にスタックの要素が露出するようになるか，Tail の数を減らして同じ属性とノードの組に戻ってくる．

補題 233. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について，$G = \mathrm{depgraph}(M)$ が非螺旋であるとする．この時，ある $c_{M,t} \in \mathbb{N}$ が存在し，任意の $n > c_{M,t}$，$\{\mathrm{Head}(\mathrm{Tail}^{m_i}(a_i(p_i)))\}_{0 \le i \le n}$ について，

- $m_0 \ge \mathrm{maxavstk}(M)$
- $\mathrm{Head}(\mathrm{Tail}^{m_0}(a_0(p_0))) \Rightarrow_{G,t} \cdots \Rightarrow_{G,t} \mathrm{Head}(\mathrm{Tail}^{m_n}(a_n(p_n)))$

を満たす時，以下のいずれかを満たす．

- ある $i \le c_{M,t}$ で，$m_i < \mathrm{maxavstk}(M)$.
- ある $i_1 < i_2 \le c_{M,t}$ で，$m_{i_1} > m_{i_2}$ かつ $a_{i_1}(p_{i_1}) = a_{i_2}(p_{i_2})$.

□

証明. $c_{M,t} = |\mathrm{attr}(M,t)|$ とおく．この時，ある $i_1 < i_2 \leq c_{M,t}$ で $a_{i_1}(p_{i_1}) = a_{i_2}(p_{i_2})$ となる．ここで，以下のように場合分けを行う．

- 任意の $0 \leq j \leq i_2$ で，$m_j \geq \mathrm{maxavstk}(M)$ の時を考える．$m_{i_1} \leq m_{i_2}$ とすると，補題 231 より，G が非螺旋であることに矛盾する．よって，$m_{i_1} > m_{i_2}$ となり正しい．
- それ以外の時，$m_j < \mathrm{maxavstk}(M)$ となる $0 \leq j \leq i_2$ が存在するため，正しい．

よって，題意は示された． ■

ここから，非螺旋 StkATT において，Tail-簡約は最初の Tail の数に比例した簡約回数で必ず正規形になるか，スタックの要素が露出するようになる．

補題 234. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について，$G = \mathrm{depgraph}(M)$ が非螺旋であるとする．この時，ある $c_{M,t,1}, c_{M,t,2} \in \mathbb{N}$ が存在し，任意の $m \in \mathbb{N}$，$n > c_{M,t,1} + m \cdot c_{M,t,2}$，$\{\mathrm{Head}(\mathrm{Tail}^{m_i}(a_i(p_i)))\}_{0 \leq i \leq n}$ について，

- $m = m_0 \geq \mathrm{maxavstk}(M)$
- $\mathrm{Head}(\mathrm{Tail}^{m_0}(a_0(p_0))) \Rightarrow_{G,t} \cdots \Rightarrow_{G,t} \mathrm{Head}(\mathrm{Tail}^{m_n}(a_n(p_n)))$

を満たす時，$m_i < \mathrm{maxavstk}(M)$ を満たす $i \leq c_{M,t,1} + m \cdot c_{M,t,2}$ が存在する． □

証明. 補題 233 の $c_{M,t}$ について $c_1 \stackrel{\mathrm{def}}{=} c_{M,t}$ とおく．また，

$$
\begin{aligned}
c_2 &\stackrel{\mathrm{def}}{=} \mathrm{maxtailstk}(M) \\
c_3 &\stackrel{\mathrm{def}}{=} c_2 \cdot c_1 \\
c_{M,t,1} &\stackrel{\mathrm{def}}{=} c_1 \cdot (2 + c_3) \\
c_{M,t,2} &\stackrel{\mathrm{def}}{=} c_1
\end{aligned}
$$

とおく．ある $i \leq c_1$ で $m_i < \mathrm{maxavstk}(M)$ の時，$i \leq c_1 \leq c_{M,t,1} + m \cdot c_{M,t,2}$ より正しい．

それ以外の時，補題 233 より，ある $i_1 < i_2 \leq c_1$ で，$m_{i_1} > m_{i_2}$ かつ $a_{i_1}(p_{i_1}) = a_{i_2}(p_{i_2})$ を満たす．この時，$m_{i_1} \leq m + c_2 \cdot i_1 \leq m + c_3$ は，i_1 に関する数学的帰納法より容易に示せる．また，$k_0 = i_2 - i_1$ とした時，任意の $0 \leq k \leq n - i_2$ について，$k = k_0 \cdot k'_1 + k'_2$ と書ける $k'_1 \in \mathbb{N}$，$0 \leq k'_2 < k_0$ が存在する．さて，任意の $0 \leq j \leq k$ について $m_{i_2+j} \geq \mathrm{maxavstk}(M)$ とする．この時，

$$
\mathrm{Head}(\mathrm{Tail}^{m_{i_2+k}}(a_{i_2+k}(p_{i_2+k}))) = \mathrm{Head}(\mathrm{Tail}^{m_{i_1+k'_2} - (k'_1+1)\cdot(m_{i_1} - m_{i_2})}(a_{i_1+k'_2}(p_{i_1+k'_2})))
$$

であることは，k に関する容易な数学的帰納法で示せる．

$$
k_0 \cdot m_{i_1} + i_2 = i_1 + k_0 \cdot (m_{i_1} + 1)
$$

$$
\begin{aligned}
&\leq c_1 + c_1 \cdot (m + c_3 + 1) \\
&= c_1 \cdot (2 + c_3) + m \cdot c_1 \\
&= c_{M,t,1} + m \cdot c_{M,t,2}
\end{aligned}
$$

より，$k \stackrel{\text{def}}{=} k_0 \cdot m_{i_1}$ とした時，$k \leq c_{M,t,1} + m \cdot c_{M,t,2} - i_2 \leq n - i_2$ より k の条件を満たし，$k'_1 = m_{i_1}$，$k_2 = 0$ となる．この時，

$$
\begin{aligned}
m_{i_2+k} &= m_{i_1+k'_2} - (k'_1 + 1) \cdot (m_{i_1} - m_{i_2}) \\
&= m_{i_1} - (m_{i_1} + 1) \cdot (m_{i_1} - m_{i_2}) \\
&< 0
\end{aligned}
$$

となるが，これは $m_{i_2+k} \geq 0$ と矛盾する．よって，ある $i \leq i_2 + k_0 \cdot m_{i_1} \leq c_{M,t,1} + m \cdot c_{M,t,2}$ で，$m_i < \text{maxavstk}(M)$ となる．以上より，題意は示された． ■

特に，正規形になる場合，属性付き木変換器の簡約システムにおいてその形は制限される．

補題 235. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について，$G = \text{depgraph}(M)$ が非螺旋であるとする．この時，任意の $m \geq \text{maxavstk}(M)$，$a(p) \in \text{attr}(M, t)$ について，

- $\text{Tail}^m(a(p)) \Rightarrow^*_{M,\sharp(t)} \text{Empty}$.
- 以下を満たす $m' \in \mathbb{N}$，$a'(p') \in \text{attr}(M, t)$ が存在する．
  - $\text{Tail}^m(a(p)) \Rightarrow^*_{M,\sharp(t)} \text{Tail}^{m'}(a'(p'))$.
  - $\text{Tail}^{m'}(a'(p'))$ は既約.
- 以下を満たす $m' < \text{maxavstk}(M)$，$a'(p') \in \text{attr}(M, t)$ が存在する．

$$
\text{Tail}^m(a(p)) \Rightarrow^*_{M,\sharp(t)} \text{Tail}^{m'}(a'(p'))
$$

のいずれかを満たす． □

証明. 以下のような場合分けを行う．

- $\sigma^{(n)} \in \Sigma$，$a(w) \xrightarrow{\sigma} \eta \in R$，$p = w[\text{w} \leftarrow p']$，$\text{label}_t(p') = \sigma$ を満たす η，p' が存在する時，$\text{maxavstk}(M)$ の定義より $\eta' \stackrel{\text{def}}{=} \text{stkeval}_{\text{stk}}(\text{Tail}^m(\eta))$ は，
  C1　$\eta' = \text{Empty}$.
  C2　$\eta' = \text{Tail}^{m_1}(a_1(p_1))$ となる $m' \in \mathbb{N}$，$a_1(p_1) \in \text{attr}(M, t)$ が存在する．
  のいずれかを満たす．C1 の時，条件を満たすことは明らかである．C2 の時，$m_1 < \text{maxavstk}(M)$ ならばやはり条件を満たす．それ以外の時，$\text{Tail}^m(a(p)) \Rightarrow^+_{M,\sharp(t)} \text{Tail}^{m_1}(a_1(p_1))$ であり，これを新たに $\text{Tail}^m(a(p))$ として同じ操作を考える．
- それ以外の時，$\text{Tail}^m(a(p))$ は既約であり，条件を満たす．

この操作は，補題 234 から有限回の繰り返しで終了することが，容易に示せる．よって，題意は示された． ■

また，スタックの要素が露出している位置においては，非螺旋 StkATT における要素型の簡約は，必ずどこかで正規形になる．

補題 236. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について，$G = \mathrm{depgraph}(M)$ が非螺旋ならば，

$$\forall \mathrm{Head}(\mathrm{Tail}^m(a(p))) \in \mathrm{htform}(\mathrm{attr}(M,t)), m < \mathrm{maxavstk}(M)\ .$$
$$\mathrm{Head}(\mathrm{Tail}^m(a(p))) \text{ の } \Rightarrow_{G,t} \text{ での任意の簡約の長さは } c_{M,t} \text{ 以下である}$$

を満たす $c_{M,t} \in \mathbb{N}$ が存在する． □

証明. 補題 234 の $c_{M,t,1}$，$c_{M,t,2}$ について，$c_1 \stackrel{\mathrm{def}}{=} c_{M,t,1}$，$c_2 \stackrel{\mathrm{def}}{=} c_{M,t,2}$ とおく．また，

$$\begin{aligned} c_3 &\stackrel{\mathrm{def}}{=} \mathrm{maxrefstk}(M) \\ c_4 &\stackrel{\mathrm{def}}{=} \mathrm{maxavstk}(M) \cdot |\mathrm{attr}(M,t)| \\ c_5 &\stackrel{\mathrm{def}}{=} 1 + c_1 + c_3 \cdot c_2 \\ c_{M,t} &\stackrel{\mathrm{def}}{=} c_5 \cdot c_4 \end{aligned}$$

とおく．まず，$\mathrm{Head}(\mathrm{Tail}^m(a(p))), \mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(p_1))) \in \mathrm{htform}(\mathrm{attr}(M,t))$ について，

- $m < \mathrm{maxavstk}(M)$
- $\mathrm{Head}(\mathrm{Tail}^m(a(p))) \Rightarrow_{G,t} \mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(p_1)))$

を満たす時を考える．この時，$m_1 \leq c_3$ であることは，定義より明らか．よって，補題 234 より，$\mathrm{Head}(\mathrm{Tail}^m(a(p)))$ は，c_5 回以下の簡約で，

- 正規形になる．
- $m_2 < \mathrm{maxavstk}(M)$ を満たす $\mathrm{Head}(\mathrm{Tail}^{m_2}(a_2(p_2))) \in \mathrm{htform}(\mathrm{attr}(M,t))$ に簡約される．

のいずれかを満たす．ここまでの議論より，ある $\mathrm{Head}(\mathrm{Tail}^{m'}(a'(p'))) \in \mathrm{htform}(\mathrm{attr}(M,t))$ が $c_{M,t}$ より長い簡約を持つとすると，その簡約中には $m'_1 < \mathrm{maxavstk}(M)$ となる

$$\mathrm{Head}(\mathrm{Tail}^{m'_1}(a'_1(p'_1))) \Rightarrow^+_{G,t} \mathrm{Head}(\mathrm{Tail}^{m'_1}(a'_1(p'_1)))$$

のような簡約を含むが，これは G が非螺旋であることに矛盾する．よって，題意は示された． ■

以上から，非螺旋 StkATT において，属性とノード，スタック位置によらずその組は必ず正規形に簡約される．

系 237. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，$G = \mathrm{depgraph}(M)$ が非螺旋ならば，任意の $t \in \hat{T}_\Sigma$ について，

$$\forall \mathrm{Head}(\mathrm{Tail}^m(a(p))) \in \mathrm{htform}(\mathrm{attr}(M,t)), c_{M,t,m} = m \cdot c_{M,t,1} + c_{M,t,2}\ .$$

$$\mathrm{Head}(\mathrm{Tail}^m(a(p))) \text{ の } \Rightarrow_{G,t} \text{ での任意の簡約の長さは } c_{M,t} \text{ 以下である}$$

を満たす $c_{M,t,1}, c_{M,t,2} \in \mathbb{N}$ が存在する．□

証明. 補題 233 の $c_1 \stackrel{\text{def}}{=} c_{M,t,1}$，$c_2 \stackrel{\text{def}}{=} c_{M,t,2}$，補題 236 の $c_3 \stackrel{\text{def}}{=} c_{M,t}$ について，

$$c_{M,t,1} \stackrel{\text{def}}{=} c_2$$
$$c_{M,t,2} \stackrel{\text{def}}{=} c_1 + c_3$$

とおく．この時，補題 234 よりどんな簡約でも $c_1 + m \cdot c_2$ 回以下で，$m_1 \leq \mathrm{maxavstk}(M)$ を満たすある $\mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(p_1))) \in \mathrm{htform}(\mathrm{attr}(M,t))$ に行き着く．よって，補題 236 より $\mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(p_1)))$ は c_3 回以下の簡約で正規形になるため，全体として $c_1 + m \cdot c_2 + c_3 = m \cdot c_{M,t,1} + c_{M,t,2}$ 回以下の簡約で正規形になる．よって，題意は示された．■

以上から，非螺旋 StkATT は要素型については簡約システムが停止性を持つことが示せる．これを示すため，要素型に限定した簡約システムを導入する．

定義 238 (StkATT の要素型簡約システム). StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_{\Sigma_\sharp}$ について，簡約システム $(\mathrm{SF}_M(t)_{\mathrm{elm}}, \Rightarrow_{M,t,\mathrm{elm}})$ を，以下のように定義する．

$$\eta_1 \Rightarrow_{M,t,\mathrm{elm}} \eta_2 \text{ iff } \eta_1 \Rightarrow_{M,t} \eta_2$$

□

非螺旋 StkATT について，要素型に限定した簡約システムは停止性を持つ．

補題 239. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，$\mathrm{depgraph}(M)$ が非螺旋ならば，任意の $t \in \hat{T}_\Sigma$ について $\Rightarrow_{M,\sharp(t),\mathrm{elm}}$ は停止性を持つ．□

証明. 補題 109 と同様に，停止性を持たないと仮定すると系 237 に矛盾することから示せる．■

ここから，StkATT の well-defined 性の同値条件は，StkATT が非螺旋であること及び依存グラフが非螺旋であることが示された．

定理 240. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，以下の 3 条件は同値である．

1. M は well-defined.
2. M は非螺旋.
3. $\mathrm{depgraph}(M)$ は非螺旋

□

証明. 2 ならば 3 を示す．M が非螺旋である時，$G = \mathrm{depgraph}(M)$ が螺旋であると仮定する．この時，以下のいずれかを満たす $t \in \hat{T}_\Sigma$，$\mathrm{Head}(\mathrm{Tail}^n(a(p))) \in \mathrm{htform}(\mathrm{attr}(M,t))$ が存在

する．

C1　$\mathrm{Head}(\mathrm{Tail}^m(a(p))) \Rightarrow^+_{G,t} \mathrm{Head}(\mathrm{Tail}^m(a(p)))$

C2　以下を満たす $c_m \geq m$ が存在する．

$$\forall k \in \mathbb{N} \,.\, \mathrm{Head}(\mathrm{Tail}^{m+k}(a(p))) \Rightarrow^+_{G,t} \mathrm{Head}(\mathrm{Tail}^{c_m+k}(a(p)))$$

この条件について，それぞれ場合分けを行う．

C1　補題 229 より M は螺旋だがこれは矛盾．よって，正しい．

C2　補題 230 より M は螺旋だがこれは矛盾．よって，正しい．

1 ならば 2 を示す．M が well-defined である時，M は螺旋であると仮定する．この時，以下のいずれかを満たす $t \in \hat{T}_\Sigma$，$\mathrm{Head}(\mathrm{Tail}^m(a(p))) \in \mathrm{htform}(\mathrm{attr}(M,t))$ が存在する．

C3　$\exists \eta = \mathrm{stkeval}_{\mathrm{elm}}(\eta), \mathrm{occ}_\eta(\mathrm{Head}(\mathrm{Tail}^m(a(p)))) \neq \emptyset \,.\, \mathrm{Head}(\mathrm{Tail}^m(a(p))) \Rightarrow^+_{M,\sharp(t)} \eta$

C4　$\exists m' \geq m \,.\, \mathrm{Tail}^m(a(p)) \Rightarrow^+_{M,\sharp(t)} \mathrm{Tail}^{m'}(a(p))$

この条件について，以下のように場合分けを行う．

C3
- $\eta_0 = \mathrm{Head}(\mathrm{Tail}^m(a(p)))$
- $\eta_0 \Rightarrow_{M,\sharp(t)} \eta_1 \Rightarrow_{M,\sharp(t)} \cdots \Rightarrow_{M,\sharp(t)} \eta_n$

を満たす $n > 0$，$\eta_0, \ldots, \eta_n$，$\pi \in \mathrm{occ}_{\eta_n}(\mathrm{Head}(\mathrm{Tail}^m(a(p))))$ が存在する．この時，$\varphi = \eta_n[\pi \leftarrow *]$ とする．$n' > n$ について，$n' = c_1 n + c_2$ と書ける $c_1 \in \mathbb{N}$，$0 \leq c_2 < n$ が存在し，$\eta_{n'} = \varphi^{c_1}(\eta_{c_2})$ とすると，$\{\eta_n\}_{n\in\mathbb{N}}$ は $\mathrm{Head}(\mathrm{Tail}^m(a(p)))$ の無限簡約列になる．これは M が well-defined に矛盾する．よって，正しい．

C4　補題 219，補題 225 より，
- $\eta_0 = \mathrm{Tail}^m(a(p))$
- $\eta_n = \mathrm{Tail}^{m'}(a(p))$
- 任意の $k \in \mathbb{N}$ について，$\mathrm{Head}(\mathrm{Tail}^k(\eta_0)) \Rightarrow_{M,\sharp(t)} \mathrm{Head}(\eta_1) \Rightarrow_{M,\sharp(t)} \cdots \Rightarrow_{M,\sharp(t)} \mathrm{Head}(\mathrm{Tail}^k(\eta_n))$

を満たす $n > 0$, $\eta_0, \ldots, \eta_n$, $m' \in \mathbb{N}$ が存在する．この時, $n' \in \mathbb{N}$ について, $n' = c_1 \cdot n + c_2$ と書ける $c_1 \in \mathbb{N}$，$0 \leq c_2 < n$ が存在し，$\eta_{n'} = \mathrm{Head}(\mathrm{Tail}^{c_1\cdot(m'-m)}(\eta_{c_2}))$ とすると，$\{\eta_n\}_{n\in\mathbb{N}}$ は $\mathrm{Head}(\mathrm{Tail}^m(a(p)))$ の無限簡約列になる．これは M が well-defined に矛盾する．よって，正しい．

3 ならば 1 は補題 239 より明らか．よって，推移律より，題意は示された． ■

以降，特に断りのない限り StkATT は well-defined であるものとする．StkATT の簡約システムは局所合流性を持つ．

補題 241. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について，$\Rightarrow_{M,\sharp(t)}$ は局所合流性を持つ．
□

証明. $\eta \in \mathrm{SF}_M(t)_k$ について，$\eta \Rightarrow_{M,\sharp(t)} \eta'_1$，$\eta \Rightarrow_{M,\sharp(t)} \eta'_2$ となる，$\eta'_1, \eta'_2 \in \mathrm{SF}_M(t)_k$ を考える．この時，$\eta'_1 \Rightarrow^*_{M,\sharp(t)} \eta'_3$ かつ $\eta'_2 \Rightarrow^*_{M,\sharp(t)} \eta'_3$ となる $\eta'_3 \in \mathrm{SF}_M(t)_k$ を，η に関する構造的帰納法で示す．

IB1　$a \in A_{\mathrm{syn}}$，$p \in \mathrm{paths}(\sharp(t))$ について，$\eta = a(p)$ の時，$\eta'_1 = \eta'_2$ より，$\eta'_3 = \eta'_1 = \eta'_2$ とおく．

IB2　IB1 と同様．

IB3　$\delta \in \Delta^{(0)} \cup \{\mathrm{Empty}, \bot\}$ について，$\eta = \delta$ の時，そのような η'_1, η'_2 は存在しないため，適用不能．

IS　$n \geq 1$，$\delta^{(n)} \in \Delta \cup \{\mathrm{Head}, \mathrm{Tail}, \mathrm{Cons}\}$ について，$\eta = \delta(\eta_1, \ldots, \eta_n)$ の時を考える．$\mathrm{stkwkeval}_k(\eta) \neq \eta$ ならば，補題 219 より，$\eta'_1 = \eta'_2$ より，IB1 と同様．それ以外の時，$\eta = \mathrm{stkwkeval}_{\mathrm{stk}}(\eta)$ に関する場合分けで示す．

C1, C2, C3　適用不能．

C4　$\delta^{(n)} = \mathrm{Cons}$ の時，$\eta'_1 = \delta(\eta'_{1,1}, \ldots, \eta'_{1,n})$，$\eta'_2 = \delta(\eta'_{2,1}, \ldots, \eta'_{2,n})$ であり，以下を満たす $i_1, i_2 \in [n]$ が存在する．

$$\forall k \in [2], i \in [n] \,.\, \begin{cases} \eta_i \Rightarrow_{M,\sharp(t)} \eta'_{k,i} & (i = i_k) \\ \eta_i = \eta'_{k,i} & (\text{otherwise}) \end{cases}$$

$i_1 = i_2$ ならば，$\eta'_1 = \eta'_2$ より IB1 と同様．$i_1 \neq i_2$ ならば，$i \in [n]$ について，

$$\eta'_{3,i} = \begin{cases} \eta'_{1,i} & (i = i_1) \\ \eta'_{2,i} & (i = i_2) \\ \eta'_{1,i} & (\text{otherwise}) \end{cases}$$

として，$\eta'_3 = \delta(\eta'_{3,1}, \ldots, \eta'_{3,n})$ とおくと，$\eta'_1 \Rightarrow_{M,\sharp(t)} \eta'_3$，$\eta'_2 \Rightarrow_{M,\sharp(t)} \eta'_3$ である．

C5　$m \in \mathbb{N}$，$a(p) \in \mathrm{attr}(M, t)$ について，$\eta = \mathrm{Tail}^m(a(p))$ の時，$\eta'_1 = \eta'_2$ より IB1 と同様．

C6　C5 と同様．

C7　C4 と同様．

よって，題意は示された．■

よって，要素型に限定しても簡約システムは局所合流性を持つ．

系 242. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について，$\Rightarrow_{M,\sharp(t),\mathrm{elm}}$ は局所合流性を持つ．
□

証明. 補題 241，系 216 より．■

また，well-defined 性から StkATT の要素型に限定した簡約システムは停止性を持つため，合流性を持つ．

系 243. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について，$\Rightarrow_{M,\sharp(t),\mathrm{elm}}$ は合流性を持つ． □

証明. 補題 239，定理 240，系 242，補題 56 より． ■

StkATT の要素型の簡約システムは，合流性と停止性を持つため，補題 57 より一意な正規形を持つ．また，根の初期属性からの正規形は，必ず出力アルファベットと $\perp$ のみからなる木になる．

補題 244. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$，$n \in \mathbb{N}$ について，以下が成り立つ．

$$\mathrm{nf}(\Rightarrow_{M,\sharp(t),\mathrm{elm}}, \mathrm{Head}(\mathrm{Tail}^n(a_0(\epsilon)))) \in \hat{T}_{\Delta_\perp}$$

□

証明. 補題 197 と同様に示せる． ■

よって，StkATT の要素型の簡約システムにおいて，入力木の根の初期属性の正規形は，必ず一意でかつ出力木になる．この時 StkATT に対して，入力木を受け取り，その根の初期属性から StkATT の簡約システムで得られるスタックを出力とする，スタック木変換が考えられる．このスタック木変換を，意味論として導入する．

定義 245 (StkATT のスタック意味論). StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，$[\![M]\!]_\omega : \hat{T}_\Sigma \to (\mathbb{N} \to \hat{T}_{\Delta_\perp})$ を以下のように定める．

$$[\![M]\!]_\omega \stackrel{\mathrm{def}}{=} t \mapsto n \mapsto \mathrm{nf}(\Rightarrow_{M,\sharp(t),\mathrm{elm}}, \mathrm{Head}(\mathrm{Tail}^n(a_0(\epsilon))))$$

また，スタック木変換のクラス $\{[\![M]\!]_\omega \mid M$ は (全域的決定的 well-defined) StkATT$\}$ を StkATT_ω と表記する． □

なお，StkATT が全域的決定的な場合，StkATT の簡約システムを決定的にすることができる．

命題 246 (StkATT の決定的な簡約システム). StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_{\Sigma_\sharp}$ について，$U \stackrel{\mathrm{def}}{=} \mathrm{SF}_M(t)$ として，簡約システム $(\bigcup_{k \in \mathrm{SS}} U_k, \Rightarrow^D_{M,t})$ を，以下のように構造的帰納法で定義する．

IB1　$a \in A_{\mathrm{syn}}$，$\sigma \in \Sigma_\sharp$，$p \in \mathrm{paths}(t)$，$\mathrm{label}_t(p) = \sigma$，$a(\mathrm{w}) \xrightarrow{\sigma} \eta \in R$ について，

$$a(p) \Rightarrow^D_{M,t} \eta[\mathrm{w} \leftarrow p].$$

IB2　$b \in A_{\mathrm{inh}}$，$\sigma \in \Sigma_\sharp$，$p \in \mathrm{paths}(t)$，$\mathrm{label}_t(p) = \sigma$，$b(\mathrm{w}i) \xrightarrow{\sigma} \eta \in R$ について，

$$b(pi) \Rightarrow^D_{M,t} \eta[\mathrm{w} \leftarrow p].$$

IB3　$\delta : () \rightarrowtriangle_{\dot{\Delta} \cup \mathrm{STK}} s$ について，δ は既約．

IS　以下のように場合分けを行う．

- $\eta_1 \in U_{\mathrm{elm}}$，$\eta_2 \in U_{\mathrm{stk}}$ について，$\mathrm{Head}(\mathrm{Cons}(\eta_1, \eta_2)) \Rightarrow_{M,t} \eta_1$．
- $\eta_1 \in U_{\mathrm{elm}}$，$\eta_2 \in U_{\mathrm{stk}}$ について，$\mathrm{Tail}(\mathrm{Cons}(\eta_1, \eta_2)) \Rightarrow_{M,t} \eta_2$．
- $\mathrm{Head}(\mathrm{Empty}) \Rightarrow_{M,t} \bot$．
- $\mathrm{Tail}(\mathrm{Empty}) \Rightarrow_{M,t} \mathrm{Empty}$．
- それ以外の場合，$n \geq 1$，$\delta : (k_1, \ldots, k_n) \rightarrowtriangle_{\dot{\Delta} \cup \mathrm{STK}} k$，$\eta_1 \in U_{k_1}, \ldots, \eta_n \in U_{k_n}$ について，$\eta_1, \ldots, \eta_{i-1}$ が $\Rightarrow^D_{M,t}$ で既約で，$\eta_i \Rightarrow^D_{M,t} \eta_i'$ の時，

$$\delta(\eta_1, \ldots, \eta_i, \ldots, \eta_n) \Rightarrow^D_{M,t} \delta(\eta_1, \ldots, \eta_i', \ldots, \eta_n).$$

この時，任意の $\eta \in U_{\mathrm{elm}}$ に対して，$\mathrm{nf}(\Rightarrow_{M,t}, \eta) = \mathrm{nf}(\Rightarrow^D_{M,t}, \eta)$． □

証明. 命題 88 と同様に示せる． ■

StkATT のスタック木変換の意味論から，木変換の意味論を導入できる．

定義 247 (StkATT の意味論). StkATT M について，$[\![M]\!]$ を $[\![M]\!]_\omega(-)(0)$ で定義する．また，木変換のクラス $\{[\![M]\!] \mid M \text{ は StkATT}\}$ を StkATT と表記する． □

なお，スタックのどの位置で木変換を構築しても，それは StkATT の木変換のクラスに含まれる．

定理 248. StkATT M，$n \in \mathbb{N}$ について，以下が成り立つ．

$$[\![M]\!]_\omega(-)(n) \in \mathrm{StkATT}$$

□

証明. 定理 201 と同様に示せる． ■

ATT は，深さが常にひとつのスタックを使用する StkATT と見なすことができる．ここから，$\mathrm{ATT}_\omega \subseteq \mathrm{StkATT}_\omega$ となる．これは，定理 186 から，つまり $\mathrm{ATT} \subseteq \mathrm{StkATT}$ ということである．

定理 249 ([Nak09]). $\mathrm{ATT}_\omega \subseteq \mathrm{StkATT}_\omega$ □

証明. 定理 202 と同様に示せる． ■

また，定理 118 と同様に，合成属性だけの StkATT は，StkTT と対応する．よって，$\mathrm{StkTT} \subseteq \mathrm{StkATT}$ である．

定理 250. $\mathrm{StkTT}_\omega \subseteq \mathrm{StkATT}_\omega$ □

証明. 定理 118 と同様に示せる. ■

StkATT で表現できる木変換として，二つの記号で構成された文字列に対し，文字列の長さが片方の記号の数で割り切れるか判定するものがある．これは，記号の数分の周期を持つ無限の長さのスタックに対し，文字列の長さだけ Tail 操作を行うことで実現される.

例 251. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ を以下のように定義する.

$$
\begin{aligned}
A_{\mathrm{syn}} &\stackrel{\mathrm{def}}{=} \{a_0, a\} \\
A_{\mathrm{inh}} &\stackrel{\mathrm{def}}{=} \{b\} \\
\Sigma &\stackrel{\mathrm{def}}{=} \{\mathrm{a}^{(1)}, \mathrm{b}^{(1)}, \$\} \\
\Delta &\stackrel{\mathrm{def}}{=} \{\mathrm{T}^{(0)}, \mathrm{F}^{(0)}\} \\
R &\stackrel{\mathrm{def}}{=} \{a_0(\mathrm{w}) \xrightarrow{\sharp} a_0(\mathrm{w1}), \\
&\qquad a_0(\mathrm{w}) \xrightarrow{\mathrm{a}} \mathrm{Tail}(a_0(\mathrm{w1})), \\
&\qquad a_0(\mathrm{w}) \xrightarrow{\mathrm{b}} \mathrm{Tail}(a_0(\mathrm{w1})), \\
&\qquad a_0(\mathrm{w}) \xrightarrow{\$} \mathrm{Tail}(b(\mathrm{w})), \\
&\qquad b(\mathrm{w1}) \xrightarrow{\sharp} \mathrm{Cons}(\mathrm{T}, a(\mathrm{w1})), \\
&\qquad b(\mathrm{w1}) \xrightarrow{\mathrm{a}} b(\mathrm{w}), \\
&\qquad b(\mathrm{w1}) \xrightarrow{\mathrm{b}} b(\mathrm{w}), \\
&\qquad a(\mathrm{w}) \xrightarrow{\mathrm{a}} \mathrm{Cons}(\mathrm{F}, a(\mathrm{w1})), \\
&\qquad a(\mathrm{w}) \xrightarrow{\mathrm{b}} a(\mathrm{w1}), \\
&\qquad a(\mathrm{w}) \xrightarrow{\$} b(\mathrm{w})\}
\end{aligned}
$$

□

命題 252. 例 251 の M について，以下が成り立つ.

$$
[\![M]\!](t) = \begin{cases} \mathrm{T} & (\mathrm{size}(t) \bmod (|\mathrm{occ}_t(\mathrm{a})| + 1) = 0) \\ \mathrm{F} & (\mathrm{otherwise}) \end{cases}
$$

□

証明. $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$A_{\mathrm{syn}} = \{a_0, a\}$，$A_{\mathrm{inh}} = \{b\}$，$t \in \hat{T}_\Sigma$ について，

$$
\mathrm{Head}(a_0(\epsilon)) \Rightarrow^*_{M, \sharp(t)} \mathrm{Head}(\mathrm{Tail}^{\mathrm{size}(t)}(b(1)))
$$

は容易に分かる．また，$l_{\mathrm{a}} = |\mathrm{occ}_t(\mathrm{a})| + 1$ として，

$$\mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^m(b(1)))) = \begin{cases} \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^{m'}(b(1)))) & (m = m' + l_{\mathrm{a}}) \\ \mathrm{T} & (m = 0) \\ \mathrm{F} & (0 < m < l_{\mathrm{a}}) \end{cases}$$

は，m に関する数学的帰納法から容易に示せる．よって，$c_t = \mathrm{size}(t) \bmod l_{\mathrm{a}}$ として，

$$\begin{aligned} [\![M]\!](t) &= \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(a_0(\epsilon))) \\ &= \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^{\mathrm{size}(t)}(b(1)))) \\ &= \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^{c_t}(b(1)))) \\ &= \begin{cases} \mathrm{T} & (c_t = 0) \\ \mathrm{F} & (\text{otherwise}) \end{cases} \end{aligned}$$

となる． ■

スタック属性付き木変換器は，他のスタック木変換器と異なり，スタックの深さが有限に止まらない場合がある．これを利用することで，上記のような木変換を表現することが可能になる．

命題 253. 例 223 の $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，以下が成り立つ．

$$\forall t \in \hat{T}_\Sigma, n \in \mathbb{N} \,.\, [\![M]\!]_\omega(t)(n) \neq \bot$$

□

証明. 命題 224 より，任意の $t \in \hat{T}_\Sigma$，$n \in \mathbb{N}$ について，$[\![M]\!]_\omega(t)(n) = \$$ である．よって，題意は示された． ■

4.4 スタックマクロ木変換器

MTT を拡張したスタック木変換器を，スタックマクロ木変換器 (StkMTT) と呼ぶ．

定義 254 (スタックマクロ木変換器 (StkMTT)). スタックマクロ木変換器は，以下の要素の組 $M = (Q, \Sigma, \Delta, e, R)$ である．

Q　状態記号のランク付きアルファベットで，$Q = \bigcup_{n \geq 1} Q^{(n)}$ ．
Σ　入力記号のランク付きアルファベット．
Δ　出力記号のランク付きアルファベットで，$\bot \notin \Delta$ ．
e　初期式で，$e \in \mathrm{RHS}_M(\mathrm{X}_1, \emptyset)_{\mathrm{stk}}$．
R　書き換え規則の集合で，以下を満たす．

$$\begin{aligned} R \in \mathcal{P}_{\mathit{fin}}(\{ & q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \mathrm{y}_1, \ldots, \mathrm{y}_m) \to \eta \\ & \mid q^{(m+1)} \in Q, \sigma^{(n)} \in \Sigma, \eta \in \mathrm{RHS}_M(\mathrm{X}_n, \mathrm{Y}_m)_{\mathrm{stk}}\}) \end{aligned}$$

ただし，$\mathrm{RHS}_M(X,Y) \in \prod_{k\in \mathrm{SS}} \mathcal{S}_{\dot{\Delta}\cup\ddot{Q}}(\{\mathrm{elm} \mapsto X, \mathrm{stk} \mapsto Y\})_k$ は，以下のように帰納的に定義される集合の族 $\{U_k\}_{k\in \mathrm{SS}}$ で定義する．

IB1　$y \in Y$ について，$y \in U_{\mathrm{stk}}$．

IB2　$\delta : () \rightarrowtriangle_{\dot{\Delta}\cup \mathrm{STK}} k$ について，$\delta \in U_k$．

IS1　$q : (\mathrm{elm}, k_1, \ldots, k_m) \rightarrowtriangle_{\ddot{Q}} k$，$x \in X$，$\eta_1 \in U_{k_1}, \ldots, \eta_m \in U_{k_m}$ について，
$q(x, \eta_1, \ldots, \eta_m) \in U_k$．

IS2　$n \geq 1$，$\delta : (k_1, \ldots, k_n) \rightarrowtriangle_{\dot{\Delta}\cup \mathrm{STK}} k$，$\eta_1 \in U_{k_1}, \ldots, \eta_n \in U_{k_n}$ について，$\delta(\eta_1, \ldots, \eta_n) \in U_k$．

□

MTT と同様，StkMTT でも全域的決定的性を定義できる．

定義 255 (全域的決定的 StkMTT). StkMTT $M = (Q, \Sigma, \Delta, e, R)$ が

$$\forall q^{(m+1)} \in Q, \sigma^{(n)} \in \Sigma \,.\, \exists! q(\sigma(\vec{\mathrm{x}}), \vec{\mathrm{y}}) \to \eta \in R$$

を満たすとき，全域的決定的であるという． □

以降，特に断りのない限り StkMTT は全域的決定的であるものとする．StkMTT について，意味論を決定づける簡約システムを導入する．

定義 256 (StkMTT の簡約システム). StkMTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X, Y について，簡約システム $(\bigcup_{k\in \mathrm{SS}} \mathrm{SF}_M(X,Y)_k, \Rightarrow_{M,X,Y})$ を，以下のように定義する．

$$\mathrm{SF}_M(X,Y) \stackrel{\mathrm{def}}{=} \mathrm{RHS}_M(\hat{T}_\Sigma(X), Y)$$

$$\eta_1 \Rightarrow_{\mathrm{stkeval}} \eta_2 \text{ iff } \left(\begin{array}{l} \exists \pi \in \mathrm{paths}(\eta_1), \\ \quad \eta_1' = \mathrm{subtree}_{\eta_1}(\pi), \\ \quad \eta_2', \eta_2 = \eta_1[\pi \leftarrow \eta_2'] \,. \\ (\eta_1' \to \eta_2') \in R \end{array} \right)$$

ただし，$R \subseteq \bigcup_{k\in \mathrm{SS}} \{\eta_1 \to \eta_2 \mid \eta_1, \eta_2 \in \mathrm{SF}_M(X,Y)_k\}$ は以下のように定義する．

$$\begin{aligned} R \stackrel{\mathrm{def}}{=} & \{\mathrm{Head}(\mathrm{Empty}) \to \bot, \mathrm{Tail}(\mathrm{Empty}) \to \mathrm{Empty}\} \\ & \cup \{\mathrm{Head}(\mathrm{Cons}(\eta_1, \eta_2)) \to \eta_1 \mid \eta_1 \in \mathrm{SF}_M(X,Y)_{\mathrm{elm}}, \eta_2 \in \mathrm{SF}_M(X,Y)_{\mathrm{stk}}\} \\ & \cup \{\mathrm{Tail}(\mathrm{Cons}(\eta_1, \eta_2)) \to \eta_2 \mid \eta_1 \in \mathrm{SF}_M(X,Y)_{\mathrm{elm}}, \eta_2 \in \mathrm{SF}_M(X,Y)_{\mathrm{stk}}\} \\ & \cup \{q(\sigma(t_1, \ldots, t_n), \eta_1, \ldots, \eta_n) \to \eta[\mathrm{x}_i \leftarrow t_i]_{i\in[n]}[\mathrm{y}_i \leftarrow \eta_i]_{i\in[m]} \\ & \qquad \mid q(\sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \mathrm{y}_1, \ldots, \mathrm{y}_m) \to \eta \in R, t_1, \ldots, t_n \in \hat{T}_\Sigma, \eta_1, \ldots, \eta_m \in \mathrm{SF}_M(X,Y)_{\mathrm{stk}}\} \end{aligned}$$

この時，$\Rightarrow_{M,\emptyset,\emptyset}$ を $\Rightarrow_M$ と表記する． □

この簡約システムは，型を保存する．

系 257. StkMTT $M = (Q, \Sigma, \Delta, e, R)$，$\eta_1 \Rightarrow^*_{M,X,Y} \eta_2$ について，以下が成り立つ．

$$\eta_1 \in \mathrm{SF}_M(X,Y)_k \implies \eta_2 \in \mathrm{SF}_M(X,Y)_k$$

□

証明. 定義より明らか． ■

また，この簡約システムにおいて，スタックシステムとしての簡約ができる．

系 258. StkMTT $M = (Q, \Sigma, \Delta, e, R)$，$\eta_1, \eta_2 \in \mathrm{SF}_M(X,Y)_k$ について，以下が成り立つ．

$$\eta_1 \Rightarrow^*_{\mathrm{stkeval}} \eta_2 \text{ implies } \eta_1 \Rightarrow^*_{M,X,Y} \eta_2$$

□

証明. 定義より明らか． ■

さらに，この簡約システムは局所合流性を持つ．

補題 259. StkMTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X, Y について，$\Rightarrow_{M,X,Y}$ は局所合流性を持つ．
□

証明. 補題 194 と同様に示せる． ■

また，この簡約システムは停止性を持つ．

補題 260. StkMTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X, Y について，$\Rightarrow_{M,X,Y}$ は停止性を持つ．□

証明. 任意の $q^{(m+1)} \in Q$，$t \in \hat{T}_\Sigma(X)$，$l_1, \ldots, l_m \in \mathbb{N}$ について，$l(q, t, l_1, \ldots, l_m) \in \mathbb{N}$ を，以下のように t に関する構造的帰納法で定義する．

IB1　$x \in X$ について，$l(q, x, l_1, \ldots, l_m) \stackrel{\mathrm{def}}{=} 0$.

IB2　$\sigma \in \Sigma^{(0)}$，$q(\sigma) \to \eta \in R$ について，

$$l(q, \sigma, l_1, \ldots, l_m) \stackrel{\mathrm{def}}{=} 1 + \mathrm{count}(\eta, l_1, \ldots, l_m).$$

IS　$n \geq 1$，$\sigma^{(n)} \in \Sigma$，$t_1, \ldots, t_n \in \hat{T}_\Sigma(X)$，$q(\sigma) \to \eta \in R$ について，

$$l(q, \sigma(t_1, \ldots, t_n), l_1, \ldots, l_m) \stackrel{\mathrm{def}}{=} 1 + \mathrm{count}(\eta[\mathrm{x}_i \leftarrow t_i]_{i \in [n]}, l_1, \ldots, l_m).$$

ただし，$\eta \in \mathrm{SF}_M(X, Y \uplus \mathrm{Y}_m)_k$，$l_1, \ldots, l_m \in \mathbb{N}$ について，$\mathrm{count}(\eta, l_1, \ldots, l_m)$ は以下のように構造的帰納法で定義する．

IB1　以下のように場合分けを行う．

- $y \in Y$ について，$\mathrm{count}(y, \ldots) \stackrel{\mathrm{def}}{=} 0$.

- $i \in [m]$ について，$\mathrm{count}(\mathrm{y}_i, l_1, \ldots, l_m) \stackrel{\mathrm{def}}{=} l_i$.

IB2 $\delta \in \Delta^{(0)} \cup \{\bot, \mathrm{Empty}\}$ について，$\mathrm{count}(\delta, \ldots) \stackrel{\mathrm{def}}{=} 0$.

IS1 $q^{(k+1)} \in Q$，$t \in \hat{T}_\Sigma(X)$，$\eta_1, \ldots, \eta_k \in \mathrm{SF}_M(X,Y)_{\mathrm{stk}}$ について，$l_{\eta_i} = \mathrm{count}(\eta_i, l_1, \ldots, l_m)$ とした時，

$$\mathrm{count}(q(t, \eta_1, \ldots, \eta_k), l_1, \ldots, l_m) \stackrel{\mathrm{def}}{=} \sum_{i \in [k]} l_{\eta_i} + l(q, t, l_{\eta_1}, \ldots, l_{\eta_k}).$$

IS2 以下のように場合分けを行う．

- $n \geq 1$，$\delta^{(n)} \in \Delta \cup \{\mathrm{Cons}\}$ について，

$$\mathrm{count}(\delta(\eta_1, \ldots, \eta_n), l_1, \ldots, l_m) \stackrel{\mathrm{def}}{=} \sum_{i \in [n]} \mathrm{count}(\eta_i, l_1, \ldots, l_m).$$

- $\delta \in \{\mathrm{Head}, \mathrm{Tail}\}$ について，

$$\mathrm{count}(\delta(\eta_1), l_1, \ldots, l_m) \stackrel{\mathrm{def}}{=} 1 + \mathrm{count}(\eta_1, l_1, \ldots, l_m).$$

また，$t \in \hat{T}_\Sigma(X)$，$\eta \in \mathrm{SF}_M(X,Y)_k$ について，

$$\mathrm{deriv}_\eta(t) \stackrel{\mathrm{def}}{\iff} \forall q \in Q, \pi \in \mathrm{occ}_\eta(q) \,.\, \exists p \in \mathrm{paths}(t) \,.\, \mathrm{subtree}_\eta(\pi 1) = \mathrm{subtree}_t(p)$$

とおく．この時，任意の $t \in \hat{T}_\Sigma$ について，

- $q^{(m+1)} \in Q$，$\eta_1, \ldots, \eta_m \in \mathrm{SF}_M(X,Y)_{\mathrm{stk}}$ について，任意の $i \in [m]$ で $\mathrm{deriv}_{\eta_i}(t)$ の時，$\{l_i\}_{i \in [m]} = \{\mathrm{count}(\eta_i)\}_{i \in [m]}$ とすると $q(t, \eta_1, \ldots, \eta_m)$ の簡約が $\sum_{i \in [m]} l_i + l(q, t, l_1, \ldots, l_m)$ 回以下で終わること
- $\eta \in \mathrm{SF}_M(X,Y)_k$ について，$\mathrm{deriv}_\eta(t)$ の時，η の簡約が $\mathrm{count}(\eta)$ 回以下で終わること

は，t に関する容易な同時帰納法で示せる．ここから，任意の $\eta \in \mathrm{SF}_M(X,Y)_k$ について，η の簡約が $\mathrm{count}(\eta)$ 回以下で終わることも，η に関する容易な構造的帰納法で示せる．よって，題意は示された． ■

以上から，StkMTT の簡約システムは合流性を持つ．

系 261. StkMTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X, Y について，$\Rightarrow_{M,X,Y}$ は合流性を持つ． □

証明. 補題 260，補題 259，補題 56 より． ■

StkMTT の簡約システムは，合流性と停止性を持つため，補題 57 より一意な正規形を持つ．要素型の正規形は，$\Rightarrow_M$ においては必ず出力アルファベットのみからなる木になる．

定理 262. StkMTT $M = (Q, \Sigma, \Delta, e, R)$ について，以下が成り立つ．

$$\forall \eta \in \mathrm{SF}_M(\emptyset, \emptyset)_{\mathrm{elm}} \,.\, \mathrm{nf}(\Rightarrow_M, \eta) \in \hat{T}_{\Delta_\bot}$$

□

証明. 補題 197 と同様に示せる. ■

この時，StkMTT に対して，初期式に入力木を割り当てたものから StkMTT の簡約システムで得られるスタックを出力とするスタック木変換を，意味論として導入する.

定義 263 (StkMTT のスタック意味論). StkMTT $M = (Q, \Sigma, \Delta, e, R)$ について，$[\![M]\!]_\omega : \hat{T}_\Sigma \to (\mathbb{N} \to \hat{T}_{\Delta_\perp})$ を以下のように定義する.

$$[\![M]\!]_\omega \stackrel{\text{def}}{=} t \mapsto n \mapsto \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^n(e[\mathrm{x}_1 \leftarrow t])))$$

また，スタック木変換のクラス $\{[\![M]\!]_\omega \mid M$ は (全域的決定的) StkMTT$\}$ を StkMTT_ω と表記する. □

なお，StkMTT が全域的決定的な場合，StkMTT の簡約システムを決定的にすることができる.

命題 264 (StkMTT の決定的な簡約システム). StkMTT $M = (Q, \Sigma, \Delta, e, R)$，集合 X, Y について，簡約システム $(\bigcup_{k \in \mathrm{SS}} \mathrm{SF}_M(X,Y)_k, \Rightarrow^D_{M,X,Y})$ を，$U \stackrel{\text{def}}{=} \mathrm{SF}_M(X,Y)$ として，以下のように構造的帰納法で定義する.

IB1　$y \in Y$ について，y は既約.

IB2　$\delta : () \rightarrowtriangle_{\dot{\Delta} \cup \mathrm{STK}} k$ について，δ は既約.

IS1　以下のように場合分けを行う.

- $q : (\mathrm{elm}, k_1, \ldots, k_m) \rightarrowtriangle_{\ddot{Q}} k$，$\sigma \in \Sigma$，$\sigma(\vec{t}) \in \hat{T}_\Sigma(X)$，$q(\sigma(\vec{\mathrm{x}}), \vec{\mathrm{y}}) \to \eta \in R$ について，

$$q(\sigma(\vec{t}), \vec{\eta_y}) \Rightarrow^D_{M,X,Y} \eta[\vec{\mathrm{x}} \leftarrow \vec{t}][\vec{\mathrm{y}} \leftarrow \vec{\eta_y}].$$

- $q : (\mathrm{elm}, k_1, \ldots, k_m) \rightarrowtriangle_{\ddot{Q}} k$，$x \in X$，$\eta_1 \in U_{k_1}, \ldots, \eta_m \in U_{k_n}$，$i \in [m]$ について，$\eta_1, \ldots, \eta_{i-1}$ が $\Rightarrow^D_{M,X,Y}$ で既約で，$\eta_i \Rightarrow^D_{M,X,Y} \eta'_i$ の時，

$$q(x, \eta_1, \ldots, \eta_i, \ldots, \eta_m) \Rightarrow^D_{M,X,Y} q(x, \eta_1, \ldots, \eta'_i, \ldots, \eta_m).$$

IS2　以下のように場合分けを行う.

- $\eta_1 \in U_{\mathrm{elm}}$，$\eta_2 \in U_{\mathrm{stk}}$ について，$\mathrm{Head}(\mathrm{Cons}(\eta_1, \eta_2)) \Rightarrow^D_{M,X,Y} \eta_1$.
- $\eta_1 \in U_{\mathrm{elm}}$，$\eta_2 \in U_{\mathrm{stk}}$ について，$\mathrm{Tail}(\mathrm{Cons}(\eta_1, \eta_2)) \Rightarrow^D_{M,X,Y} \eta_2$.
- $\mathrm{Head}(\mathrm{Empty}) \Rightarrow^D_{M,X,Y} \perp$.
- $\mathrm{Tail}(\mathrm{Empty}) \Rightarrow^D_{M,X,Y} \mathrm{Empty}$.
- それ以外の場合，$n \geq 1$，$\delta : (k_1, \ldots, k_n) \rightarrowtriangle_{\dot{\Delta} \cup \mathrm{STK}} k$，$\eta_1 \in U_{k_1}, \ldots, \eta_n \in U_{k_n}$，$i \in [n]$ について，$\eta_1, \ldots, \eta_{i-1}$ が $\Rightarrow^D_{M,X,Y}$ で既約で，$\eta_i \Rightarrow^D_{M,X,Y} \eta'_i$ の時，

$$\delta(\eta_1, \ldots, \eta_i, \ldots, \eta_n) \Rightarrow^D_{M,X,Y} \delta(\eta_1, \ldots, \eta'_i, \ldots, \eta_n).$$

この時，任意の $\eta \in U_k$ に対して，$\mathrm{nf}(\Rightarrow_{M,X,Y}, \eta) = \mathrm{nf}(\Rightarrow^{D}_{M,X,Y}, \eta)$. □

証明. 命題 88 と同様に示せる. ■

StkMTT のスタック木変換の意味論から，木変換の意味論を導入できる.

定義 265 (StkMTT の意味論). StkMTT M について，$[\![M]\!]$ を $[\![M]\!]_\omega(-)(0)$ で定義する．また，木変換のクラス $\{[\![M]\!] \mid M \text{ は StkMTT}\}$ を StkMTT と表記する. □

なお，スタックのどの位置で木変換を構築しても，それは StkMTT の木変換のクラスに含まれる.

定理 266. StkMTT M，$n \in \mathbb{N}$ について，以下が成り立つ.

$$[\![M]\!]_\omega(-)(n) \in \mathrm{StkMTT}$$

□

証明. 定理 201 と同様に示せる. ■

MTT は，深さが常にひとつのスタックを使用する StkMTT と見なすことができる．ここから，$\mathrm{MTT}_\omega \subseteq \mathrm{StkMTT}_\omega$ となる.

定理 267. $\mathrm{MTT}_\omega \subseteq \mathrm{StkMTT}_\omega$ □

証明. 定理 202 と同様に示せる. ■

また，定理 133 と同様に，コンテキストパラメータを持たない StkMTT は，StkTT と対応する．よって，StkTT $\subseteq$ StkMTT である.

定理 268. $\mathrm{StkTT}_\omega \subseteq \mathrm{StkMTT}_\omega$ □

証明. 定理 133 と同様に示せる. ■

StkTT 同様，StkMTT も入力木を固定すると，深さが有限のスタックしか構築できない.

定理 269. $[\![M]\!]_\omega : \hat{T}_\Sigma \to (\mathbb{N} \to \hat{T}_{\Delta_\perp}) \in \mathrm{StkMTT}_\omega$ について，以下が成り立つ.

$$\forall t \in \hat{T}_\Sigma \,.\, \exists c_{M,t} \in \mathbb{N}, \forall n \geq c_{M,t} \,.\, [\![M]\!]_\omega(t)(n) = \perp$$

□

証明. 定理 205 と同様に示せる. ■

MTT と同様，StkMTT にも滞在拡張を考えることができる．滞在拡張を施した StkMTT を**滞在付きスタックマクロ木変換器** ($stayStkMTT$) と呼ぶ.

定義 270 (滞在付きスタックマクロ木変換器 (stayStkMTT)). 滞在付きスタックマクロ木変換器は，以下の要素の組 $M = (Q, \Sigma, \Delta, e, R)$ である．

Q, Σ, Δ, e　StkMTT と同様.
R　　以下のようにする．

$$R \in \mathcal{P}_{fin}(\{q(\mathrm{x} = \sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \mathrm{y}_1, \ldots, \mathrm{y}_m) \to \eta \\ \mid q^{(m+1)} \in Q, \sigma^{(n)} \in \Sigma, \eta \in \mathrm{RHS}_M(\mathrm{X}_n \cup \{\mathrm{x}\}, \mathrm{Y}_m)_{\mathrm{stk}}\})$$

ただし，RHS_M は StkMTT と同様に定義する．　□

stayMTT の時と同様，stayStkMTT でも全域的決定的性を定義できる．

定義 271 (全域的決定的 stayStkMTT). stayStkMTT $M = (Q, \Sigma, \Delta, e, R)$ が

$$\forall q^{(m+1)} \in Q, \sigma^{(n)} \in \Sigma \,.\, \exists! q(\mathrm{x} = \sigma(\vec{\mathrm{x}}), \vec{\mathrm{y}}) \to \eta \in R$$

を満たすとき，全域的決定的であるという．　□

以降，特に断りのない限り stayStkMTT は全域的決定的であるものとする．stayStkMTT の意味論を決定づける簡約システムを，StkATT と同様，最外戦略によって導入する．これは，stayMTT と同様 stayStkMTT も一般に簡約システムが停止性を持たないためである．最外戦略を導入する際は，通常コンテキストパラメータの簡約を行わず先に規則を適用することになるが，この場合，入力木に変数を導入すると局所合流性を持たない簡約システムになってしまう．そこで，StkMTT と異なり，入力木への変数の導入も行わない．

定義 272 (stayStkMTT の簡約システム). stayStkMTT $M = (Q, \Sigma, \Delta, e, R)$，集合 Y について，$\mathrm{SF}_M(Y) \stackrel{\mathrm{def}}{=} \mathrm{RHS}_M(\hat{T}_\Sigma, Y)$ とする．この時，簡約システム $(\bigcup_{k \in \mathrm{SS}} \mathrm{SF}_M(Y)_k, \Rightarrow_{M,Y})$ を，$U \stackrel{\mathrm{def}}{=} \mathrm{SF}_M(Y)$ として，以下のように構造的帰納法で定義する．

IB1　$y \in Y$ について，y は既約.
IB2　$\delta : () \to_{\dot{\Delta} \cup \mathrm{STK}} k$ について，δ は既約.
IS1　$q : (\mathrm{elm}, k_1, \ldots, k_m) \to_{\ddot{Q}} k$，$\sigma \in \Sigma$，$t = \sigma(t_1, \ldots, t_n) \in \hat{T}_\Sigma$，
$q(\mathrm{x} = \sigma(\vec{\mathrm{x}}_1, \ldots, \mathrm{x}_n), \vec{\mathrm{y}}) \to \eta \in R$ について，

$$q(\sigma(\vec{t}), \vec{\eta_y}) \Rightarrow_{M,Y} \eta[\mathrm{x} \leftarrow t][\mathrm{x}_i \leftarrow t_i]_{i \in [n]}[\vec{\mathrm{y}} \leftarrow \vec{\eta_y}].$$

IS2　以下のように場合分けを行う．

- $\eta_1 \in U_{\mathrm{elm}}$，$\eta_2 \in U_{\mathrm{stk}}$ について，$\mathrm{Head}(\mathrm{Cons}(\eta_1, \eta_2)) \Rightarrow_{M,Y} \eta_1$.
- $\eta_1 \in U_{\mathrm{elm}}$，$\eta_2 \in U_{\mathrm{stk}}$ について，$\mathrm{Tail}(\mathrm{Cons}(\eta_1, \eta_2)) \Rightarrow_{M,Y} \eta_2$.
- $\mathrm{Head}(\mathrm{Empty}) \Rightarrow_{M,Y} \bot$.
- $\mathrm{Tail}(\mathrm{Empty}) \Rightarrow_{M,Y} \mathrm{Empty}$.

- それ以外の場合，$n \geq 1$，$\delta : (k_1, \ldots, k_n) \twoheadrightarrow_{\dot{\Delta} \cup \mathrm{STK}} k$，$\eta_1 \in A_{k_1}, \ldots, \eta_n \in A_{k_n}$，$i \in [n]$ について，$\eta_i \Rightarrow_{M,Y} \eta_i'$ の時，

$$\delta(\eta_1, \ldots, \eta_i, \ldots, \eta_n) \Rightarrow_{M,Y} \delta(\eta_1, \ldots, \eta_i', \ldots, \eta_n).$$

この時，$\Rightarrow_{M,\emptyset}$ を $\Rightarrow_M$ と表記する． □

StkATT と同じく stayStkMTT も，一般には簡約システムは停止性を持たない．特に，要素型に限定した場合の簡約システムも停止性を持たない．これは，stayStkMTT が全域的決定的であっても意味論が関数にならない場合があることを意味する．そのような場合を除くため，StkATT，stayMTT と同様に well-defined の概念を導入する．

定義 273 (stayStkMTT の well-defined 性). stayStkMTT $M = (Q, \Sigma, \Delta, e, R)$ について，

$$\forall t \in \hat{T}_\Sigma, q^{(m+1)} \in Q, n \in \mathbb{N}\,.$$
$$\mathrm{Head}(\mathrm{Tail}^n(q(t, \mathrm{y}_1, \ldots, \mathrm{y}_m))) \text{ は } \Rightarrow_{M,\emptyset,\mathrm{Y}_m} \text{ で無限簡約可能でない}$$

を満たす時 M は well-defined という． □

well-defined stayStkMTT は，StkATT と同様，要素型については簡約システムが停止性を持つことが示せる．これを示すため，要素型に限定した簡約システムを導入する．

定義 274 (stayStkMTT の要素型簡約システム). stayStkMTT $M = (Q, \Sigma, \Delta, e, R)$ について，簡約システム $(\mathrm{SF}_M(Y)_{\mathrm{elm}}, \Rightarrow_{M,Y,\mathrm{elm}})$ を，以下のように定義する．

$$\eta_1 \Rightarrow_{M,Y,\mathrm{elm}} \eta_2 \text{ iff } \eta_1 \Rightarrow_{M,Y} \eta_2$$

また，$\Rightarrow_{M,\mathrm{elm}} = \Rightarrow_{M,\emptyset,\mathrm{elm}}$ と表記する． □

well-defined stayStkMTT について，要素型に限定した簡約システムは停止性を持つ．

補題 275. stayStkMTT $M = (Q, \Sigma, \Delta, e, R)$ について，M が well-defined ならば，$\Rightarrow_{M,Y,\mathrm{elm}}$ は停止性を持つ． □

証明. $\Rightarrow_{M,Y,\mathrm{elm}}$ が停止性を持たないと仮定すると，無限簡約可能な $\eta \in \mathrm{SF}_M(Y)_{\mathrm{elm}}$ が存在する．この時，$\mathrm{occ}_\eta(q) \neq \emptyset$ を満たす $q \in Q$ が存在し，この q は well-defined の条件を満たさないことが，示せる．これは，M が well-defined に矛盾することから，題意は示された． ■

以降，特に断りのない限り stayStkMTT は well-defined であるものとする．stayStkMTT の簡約システムは，局所合流性を持つ．

補題 276. stayStkMTT $M = (Q, \Sigma, \Delta, e, R)$ について，$\Rightarrow_{M,Y}$ は局所合流性を持つ． □

証明. 補題 241 と同様に示せる． ■

よって，要素型に限定しても簡約システムは局所合流性を持つ．

系 277. stayStkMTT $M = (Q, \Sigma, \Delta, e, R)$ について，$\Rightarrow_{M,Y,\mathrm{elm}}$ は局所合流性を持つ． □

証明. 補題 276 より． ■

また，well-defined 性から stayStkMTT の要素型に限定した簡約システムは停止性を持つため，合流性を持つ．

系 278. stayStkMTT $M = (Q, \Sigma, \Delta, e, R)$ について，$\Rightarrow_{M,Y,\mathrm{elm}}$ は合流性を持つ． □

証明. 補題 275，系 277，補題 56 より． ■

stayStkMTT の要素型の簡約システムは，合流性と停止性を持つため，補題 57 より一意な正規形を持つ．要素型の正規形は，$\Rightarrow_{M,\mathrm{elm}}$ においては必ず出力アルファベットと $\perp$ のみからなる木になる．

補題 279. stayStkMTT $M = (Q, \Sigma, \Delta, e, R)$ について，以下が成り立つ．

$$\forall \eta \in \mathrm{SF}_M(\emptyset)_{\mathrm{elm}} \,.\, \mathrm{nf}(\Rightarrow_{M,\mathrm{elm}}, \eta) \in \hat{T}_{\Delta_\perp}$$

□

証明. 補題 197 と同様に示せる． ■

この時，stayStkMTT に対して，初期式に入力木を割り当てたものから stayStkMTT の要素型簡約システムで得られるスタックを出力とするスタック木変換を，意味論として導入する．

定義 280 (stayStkMTT のスタック意味論). stayStkMTT $M = (Q, \Sigma, \Delta, e, R)$ について，$[\![M]\!]_\omega : \hat{T}_\Sigma \to (\mathbb{N} \to \hat{T}_{\Delta_\perp})$ を以下のように定める．

$$[\![M]\!]_\omega \stackrel{\mathrm{def}}{=} t \mapsto n \mapsto \mathrm{nf}(\Rightarrow_{M,\mathrm{elm}}, \mathrm{Head}(\mathrm{Tail}^n(e[\mathrm{x}_1 \leftarrow t])))$$

また，スタック木変換のクラス $\{[\![M]\!]_\omega \mid M$ は (全域的決定的 well-defined) stayStkMTT$\}$ を $\mathrm{stayStkMTT}_\omega$ と表記する． □

なお，stayStkMTT が全域的決定的な場合，stayStkMTT の簡約システムを決定的にすることができる．

命題 281 (stayStkMTT の決定的な簡約システム). stayStkMTT $M = (Q, \Sigma, \Delta, e, R)$，集合 Y について，簡約システム $(\bigcup_{k \in \mathrm{SS}} U_k, \Rightarrow^D_{M,Y})$ を，$U \stackrel{\mathrm{def}}{=} \mathrm{SF}_M(Y)$ として，以下のように構造的帰納法で定義する．

IB1　$y \in Y$ について，y は既約．

IB2　$\delta : () \rightarrowtriangle_{\dot{\Delta} \cup \mathrm{STK}} k$ について，δ は既約．

IS1　$q : (\mathrm{elm}, k_1, \ldots, k_m) \rightarrowtriangle_{\ddot{Q}} k$，$\sigma \in \Sigma$，$\sigma(\vec{t}) \in \hat{T}_\Sigma$，$q(\sigma(\vec{\mathrm{x}}), \vec{\mathrm{y}}) \to \eta \in R$ について，

$$q(\sigma(\vec{t}), \vec{\eta_y}) \Rightarrow^D_{M,Y} \eta[\vec{\mathrm{x}} \leftarrow \vec{t}][\vec{\mathrm{y}} \leftarrow \vec{\eta_y}].$$

IS2　以下のように場合分けを行う．

- $\eta_1 \in U_{\mathrm{elm}}$，$\eta_2 \in U_{\mathrm{stk}}$ について，$\mathrm{Head}(\mathrm{Cons}(\eta_1, \eta_2)) \Rightarrow^D_{M,Y} \eta_1$．
- $\eta_1 \in U_{\mathrm{elm}}$，$\eta_2 \in U_{\mathrm{stk}}$ について，$\mathrm{Tail}(\mathrm{Cons}(\eta_1, \eta_2)) \Rightarrow^D_{M,Y} \eta_2$．
- $\mathrm{Head}(\mathrm{Empty}) \Rightarrow^D_{M,Y} \bot$．
- $\mathrm{Tail}(\mathrm{Empty}) \Rightarrow^D_{M,Y} \mathrm{Empty}$．
- それ以外の場合，$n \geq 1$，$\delta : (k_1, \ldots, k_n) \rightarrowtriangle_{\dot{\Delta} \cup \mathrm{STK}} k$，$\eta_1 \in U_{k_1}, \ldots, \eta_n \in U_{k_n}$，$i \in [n]$ について，$\eta_1, \ldots, \eta_{i-1}$ が $\Rightarrow^D_{M,Y}$ で既約で，$\eta_i \Rightarrow^D_{M,Y} \eta_i'$ の時，

$$\delta(\eta_1, \ldots, \eta_i, \ldots, \eta_n) \Rightarrow^D_{M,Y} \delta(\eta_1, \ldots, \eta_i', \ldots, \eta_n).$$

この時，任意の $\eta \in U_k$ に対して，$\mathrm{nf}(\Rightarrow_{M,Y}, \eta) = \mathrm{nf}(\Rightarrow^D_{M,Y}, \eta)$．　□

証明. 命題 88 と同様に示せる．　■

stayStkMTT のスタック木変換の意味論から，木変換の意味論を導入できる．

定義 282 (stayStkMTT の意味論). stayStkMTT M について，$[\![M]\!]$ を $[\![M]\!]_\omega(-)(0)$ で定義する．また，木変換のクラス $\{[\![M]\!] \mid M \text{ は stayStkMTT}\}$ を stayStkMTT と表記する．　□

なお，スタックのどの位置で木変換を構築しても，それは stayStkMTT の木変換のクラスに含まれる．

定理 283. stayStkMTT M，$n \in \mathbb{N}$ について，以下が成り立つ．

$$[\![M]\!]_\omega(-)(n) \in \mathrm{stayStkMTT}$$

□

証明. 定理 201 と同様に示せる．　■

滞在拡張を使用しない stayStkMTT は StkMTT と対応する．よって，StkMTT $\subseteq$ stayStkMTT である．

定理 284. $\mathrm{StkMTT}_\omega \subseteq \mathrm{stayStkMTT}_\omega$　□

証明. 定理 150 と同様に示せる．　■

第 5 章　スタック木変換器の表現力

この章では，スタック木変換器の表現力について，明らかになったことと未解決の問題について述べる．特に，他のスタック木変換器と異なる表現力を持つ StkATT の性質，スタック木変換器と木変換器の関係性について述べる．

スタック属性付き木変換器は，その簡約システムが例 223 の観察から分かる通り最外戦略でしか停止性を持たないこと，命題 253 から分かる通り無限の長さのスタックを持てることが，他のスタック木変換器と異なる点である．さらに，スタック属性付き木変換器に対し，定義 285 で与えられる簡約システムがスタック型の要素の簡約についても最外戦略で停止性を持つ fully-defined，定義 292 で与えられる簡約システムが戦略によらず停止性を持つ非巡回の制約を考えることができる．fully-defined 性は well-defined 性と同じく意味論的に与えられるが，より構文的な同値条件を与えることができる．その条件と正当性は，定理 287 で示している．well-defined 性，fully-defined 性，非巡回性によるクラスは，それぞれ異なると予想しているが，本研究では証明には至らなかった．ただ，これらのクラスの性質に違いによらず，スタック属性付き木変換器の樹高性は属性付き木変換器の樹高性と同じになることが分かった．その正当性は，定理 305 で示している．

スタック属性付き木変換器を除く 2 つのスタック木変換器，スタックトップダウン木変換器とスタックマクロ木変換器は，それぞれトップダウン木変換器，マクロ木変換器とスタック評価器の合成と同等の表現力を持つことが分かった．その正当性は定理 314 で示している．また，非巡回スタック属性付き木変換器が，属性付き木変換器とスタック評価器の合成と同等の表現力を持つことも，定理 318 で示している．スタック評価器は入力木のスタック操作に対応するコンストラクタを評価することで出力からは削除するようになっており，樹高性は入力木の高さに対して線形になる．その正当性は定理 320 で示している．ここから，スタックトップダウン木変換器とスタックマクロ木変換器の樹高性が，トップダウン木変換器とマクロ木変換器の樹高性と同様になることが，系 321 のように示せる．さらに樹高性により，スタックトップダウン木変換器，非巡回スタック属性付き木変換器，スタックマクロ木変換器の間に真の部分クラス関係により階層が作れることが，定理 323 で示した．これは本研究の 1 つの成果である．また，系 332 でスタックマクロ木変換器とスタック属性付き木変換器との間にこの順に部分クラス関係が成り立たないことも示した．本研究では，両者についての逆の関係も成り立たないと予想しており，その考察を行なったが，証明に至らなかった．この問題は予想 333 で形式化しており，スタック属性付き木変換器の部分クラスに対する未解決問題とも強く関連していることについても考察した．この考察は，5.3 で行なっている．

5.1 スタック属性付き木変換器の性質

命題 253 から分かるように，StkTT，StkMTT と異なり，StkATT が出力するスタックは，無限の深さを持つ場合がある．ただし，スタックの深さが有限になる StkATT も存在する．このような StkATT の部分クラスとして，スタック型でも簡約が停止するクラスが考えられる．このクラスを，*fully-defined* StkATT として導入する．

定義 285 (StkATT の fully-defined). StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，

$$\forall t \in \hat{T}_\Sigma, a(p) \in \mathrm{attr}(M, t)\ .\ a(p) \text{ は } \Rightarrow_{M,\sharp(t)} \text{ で無限簡約可能でない}$$

を満たす時 M は fully-defined であるという．

また，スタック木変換のクラス $\{[\![M]\!]_\omega \mid M \text{ は fully-defined StkATT}\}$ を $\mathrm{StkATT}_{\mathrm{fd},\omega}$，木変換のクラス $\{[\![M]\!] \mid M \text{ は fully-defined StkATT}\}$ を $\mathrm{StkATT}_{\mathrm{fd}}$ と表記する． □

well-defined に限らない StkATT でも，fully-defined であれば well-defined である．

命題 286. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，fully-defined なら well-defined. □

証明. $t \in \hat{T}_\Sigma$，$\eta \in \mathrm{SF}_M(t)_{\mathrm{stk}}$ について，η が無限簡約可能でないなら，$\mathrm{Head}(\mathrm{Tail}^m(\eta))$ が無限簡約可能であるとすると矛盾することは，補題 239 と同様の議論で示せる．ここから，fully-defined の時 well-defined であることが導かれる． ■

fully-defined StkATT の簡約システムは，スタック型であっても停止性を持つ．これは，fully-defined の同値条件として自身に戻ってくるような Tail-簡約が存在しないというものがあるからである．

定理 287. (well-defined な) StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，以下の 2 条件は同値である．

- M は fully-defined.
- 以下を満たす $t \in \hat{T}_\Sigma$，$a(p) \in \mathrm{attr}(M, t)$，$m_1, m_2 \in \mathbb{N}$ は存在しない．

$$\mathrm{Tail}^{m_1}(a(p)) \Rightarrow^+_{M,\sharp(t)} \mathrm{Tail}^{m_2}(a(p))$$

□

証明. M が fully-defined の時，

$$\mathrm{Tail}^{m_1}(a(p)) \Rightarrow^+_{M,\sharp(t)} \mathrm{Tail}^{m_2}(a(p))$$

を満たす $t \in \hat{T}_\Sigma$，$a(p) \in \mathrm{attr}(M, t)$，$m_1, m_2 \in \mathbb{N}$ が存在すると仮定する．この時，この簡約を続けていくと無限簡約列となり，ここから $a(p)$ の無限簡約列を作ることができるが，これは M

が fully-defined であることに矛盾する．また，このような Tail-簡約が存在しないとすると，スタック型の簡約システムの停止性は，要素型の簡約システムの停止性に帰着でき，well-defined 性よりスタック型の簡約システムは停止性を持つ．よって，fully-defined であることが直ちに分かる．よって，題意は示された． ■

ここから，fully-defined StkATT の簡約システムは，型によらず停止性を持つ．

補題 288. fully-defined StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について $\Rightarrow_{M,\sharp(t)}$ は停止性を持つ． □

証明. 定理 287 での議論より． ■

よって，fully-defined StkATT の簡約システムは，型によらず合流性を持つ．

系 289. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$，$t \in \hat{T}_\Sigma$ について，$\Rightarrow_{M,\sharp(t),\mathrm{elm}}$ は合流性を持つ． □

証明. 補題 288，補題 241，補題 56 より． ■

ここから，fully-defined StkATT では，スタックの深さは有限になる．

定理 290. $[\![M]\!]_\omega : \hat{T}_\Sigma \to (\mathbb{N} \to \hat{T}_{\Delta_\perp}) \in \mathrm{StkATT}_{\mathrm{fd},\omega}$ について，以下が成り立つ．

$$\forall t \in \hat{T}_\Sigma \,.\, \exists c_{M,t} \in \mathbb{N}, \forall n \geq c_{M,t} \,.\, [\![M]\!]_\omega(t)(n) = \bot$$

□

証明. 定理 205 と同様に示せる． ■

ただし，fully-defined であっても，自由な戦略を許すと簡約が停止性を持たないことがある．fully-defined であり，自由な戦略の元では簡約が停止しない例として，以下のものがある．

例 291. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ を以下のように定義する．

$$\begin{aligned}
A_{\mathrm{syn}} &\stackrel{\mathrm{def}}{=} \{a\} \\
A_{\mathrm{inh}} &\stackrel{\mathrm{def}}{=} \{b\} \\
\Sigma &\stackrel{\mathrm{def}}{=} \{\$^{(0)}\} \\
\Delta &\stackrel{\mathrm{def}}{=} \Sigma \\
a_0 &\stackrel{\mathrm{def}}{=} a \\
R &\stackrel{\mathrm{def}}{=} \{a(\mathrm{w}) \xrightarrow{\sharp} a(\mathrm{w1}), a(\mathrm{w}) \xrightarrow{\$} b(\mathrm{w}), b(\mathrm{w1}) \xrightarrow{\sharp} \mathrm{Cons}(\mathrm{Head}(\mathrm{Tail}(a(\mathrm{w1}))), \mathrm{Empty})\}
\end{aligned}$$

□

ところで，fully-defined StkATT の部分クラスで，自由な戦略の下でも簡約システムが停止性を持つものとして，ATT と同様の非巡回性を StkATT に課したものがある．これを，非巡回 StkATT として導入する．

定義 292 (StkATT の非巡回性). StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，ATT $(A, \Sigma, \Delta \cup \bar{\mathrm{STK}}, \sharp, a_0, R)$ が巡回である時，M は巡回であるという．M が巡回でない時，M は非巡回であるという．この時，スタック木変換のクラス $\{[\![M]\!]_\omega \mid M \text{ は非巡回 StkATT}\}$ を $\mathrm{StkATT}_{\mathrm{ncr},\omega}$，木変換のクラス $\{[\![M]\!] \mid M \text{ は非巡回 StkATT}\}$ を $\mathrm{StkATT}_{\mathrm{ncr}}$ と表記する． □

非巡回 StkATT は fully-defined である．つまり，$\mathrm{StkATT}_{\mathrm{ncr},\omega} \subseteq \mathrm{StkATT}_{\mathrm{fd},\omega}$ である．

定理 293. 非巡回 StkATT M は，fully-defined である． □

証明. 定理 110 と同様に示せる． ■

また，定理 249 で作成した StkATT は，非巡回性を満たす．

定理 294. $\mathrm{ATT}_\omega \subseteq \mathrm{StkATT}_{\mathrm{ncr},\omega}$ □

証明. 定理 249 の StkATT は，ATT の well-defined 性より明らかに非巡回より． ■

また，StkATT の非巡回性は，ATT の非巡回性から決定できることは明らかより，ただちに決定可能であることが分かる．

系 295. (well-defined に限らない) StkATT M について，M が非巡回かは決定可能． □

証明. 定理 111 より． ■

しかし，StkATT の well-defined 性の決定可能性は未解決である．本研究では，StkATT の well-defined 性は決定不能ではないかと考えている．

予想 296. StkATT M について，M が well-defined かは決定不能． □

この予想は，単項の入力木を持つ StkATT について，依存グラフの螺旋性判定が PCP に帰着できるのではないかという考察による．しかし，今のところ有用な帰着手段は見つかっていない．さらには，well-defined に限らない StkATT の fully-defined 性の決定可能性も未解決である．この問題も well-defined 性の判定と同様 PCP に帰着できるのではないかと予想しており，すなわち fully-defined かの判定も決定不能であると予想している．

予想 297. (well-defined に限らない) StkATT M について，M が fully-defined かは決定不能． □

ただし，well-defined StkATT が fully-defined かは決定可能である．

定理 298. (well-defined な) StkATT M について，M が fully-defined かは決定可能． □

証明. 定理 287 から，同じノードの同じ属性の結果に依存する Tail-簡約がない時，M は fully-defined である．これは，依存グラフを Tail-簡約のみ考慮するようにし，アルゴリズム 1 から決定できる．正当性についても，定理 111 と同様に示せる． ■

つまり，well-defined 性が決定可能であれば，fully-defined 性も決定可能になる．決定可能性は応用を考える上で重要であり，この予想は解決すべき大きな今後の課題の 1 つとなっている．

また，StkATT と fully-defined StkATT，非巡回 StkATT のクラスの関係についても未解決である．

予想 299. $\mathrm{StkATT}_{\mathrm{fd}} \not\subseteq \mathrm{StkATT}_{\mathrm{ncr}}$ □

予想 300. $\mathrm{StkATT} \not\subseteq \mathrm{StkATT}_{\mathrm{fd}}$ □

予想 301. $\mathrm{StkATT} \not\subseteq \mathrm{StkATT}_{\mathrm{ncr}}$ □

もし，$\mathrm{StkATT} \subseteq \mathrm{StkATT}_{\mathrm{ncr}}$ ならば $\mathrm{StkATT} = \mathrm{StkATT}_{\mathrm{fd}} = \mathrm{StkATT}_{\mathrm{ncr}}$ であり，系 295 より予想 296，予想 297 は否定的に解決される．逆に，それぞれ予想 296，予想 297 が成り立つならば，それぞれ予想 301，予想 299 は肯定的に解決される．予想 301 を支持する根拠のひとつに，次の予想がある．

予想 302. 例 251 の M について，$[\![M]\!] \notin \mathrm{StkATT}_{\mathrm{ncr}}$ □

これは，計算の複雑性から示せるのではないかと考えている．この予想は StkMTT と StkATT との関係性についても強く関連しており，より詳細な考察は 5.3 で述べる．また，予想 300 の関係はスタック木変換のクラスにおいては成り立つ．

補題 303. $\mathrm{StkATT}_{\omega} \not\subseteq \mathrm{StkATT}_{\mathrm{fd},\omega}$ □

証明. $\mathrm{StkATT}_{\omega} \subseteq \mathrm{StkATT}_{\mathrm{fd},\omega}$ が成り立つと仮定した時，例 251 の $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，$[\![M]\!]_{\omega} \in \mathrm{StkATT}_{\mathrm{fd},\omega}$ であり定理 290 より，$t \in \hat{T}_{\Sigma}$ について $[\![M]\!]_{\omega}(t)(n) = \bot$ を満たす $n \in \mathbb{N}$ が存在する．しかし，これは命題 253 と矛盾する．よって，題意は示された． ■

よって，スタック木変換のクラスとしては，well-defined な StkATT によるクラスは fully-defined な StkATT のクラスより真に大きい．

定理 304. $\mathrm{StkATT}_{\mathrm{fd},\omega} \subsetneq \mathrm{StkATT}_{\omega}$ □

証明. 命題 286，補題 303 より． ■

スタック木変換のクラスとして真に大きいということは，StkATT から自然に対応づけられる fully-defined な StkATT に変換するアルゴリズムがないことを意味する．これが予想 300 の根

拠となっている．

なお，非巡回かどうかに関わらず，ATT と同様の樹高性が StkATT でも成り立つ．

定理 305 (StkATT の樹高性). $[\![M]\!] : \hat{T}_\Sigma \to \hat{T}_{\Delta_\perp} \in \mathrm{StkATT}$ について，

$$\forall t \in \hat{T}_\Sigma \,.\, \mathrm{height}([\![M]\!](t)) \leq c_M \cdot \mathrm{size}(t)$$

を満たす $c_M \in \mathbb{N}$ が存在する． □

証明. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，

$$\begin{aligned} c_1 &\stackrel{\mathrm{def}}{=} \{\mathrm{height}(\eta) \mid a(w) \xrightarrow{\sigma} \eta \in R\} \\ c_2 &\stackrel{\mathrm{def}}{=} \sum_{k \in \mathrm{AS}} |A_k| \\ c_3 &\stackrel{\mathrm{def}}{=} \mathrm{maxavstk}(M) \\ c_M &\stackrel{\mathrm{def}}{=} 2 \cdot c_1 \cdot c_2 \cdot c_3 \end{aligned}$$

とおく．また，任意の $t \in \hat{T}_\Sigma$ について，$h_t : \mathrm{htform}(\mathrm{attr}(M,t)) \times \mathbb{N} \to \mathbb{N}$ を，以下のように数学的帰納法で定義する．

IB　$h_t(\mathrm{Head}(\mathrm{Tail}^m(a(p))), 0) \stackrel{\mathrm{def}}{=} 0$.

IS　補題 235 より，$\mathrm{Head}(\mathrm{Tail}^m(a(p)))$ は，

- C1　正規形 Empty になる．
- C2　正規形 $\mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(p_1))) \in \mathrm{htform}(\mathrm{attr}(M,t))$ になる．
- C3　ある $m_1 < \mathrm{maxavstk}(M)$，$a_1(p_1) \in \mathrm{attr}(M,t)$ について，$\mathrm{Head}(\mathrm{Tail}^{m_1}(a_1(p_1)))$ に簡約される．

のいずれかを満たす．

- C1 の時 $h_t(\mathrm{Head}(\mathrm{Tail}^m(a(p))), k+1) \stackrel{\mathrm{def}}{=} 1$
- C2 の時 $h_t(\mathrm{Head}(\mathrm{Tail}^m(a(p))), k+1) \stackrel{\mathrm{def}}{=} 0$

とおく．また，C3 の時，以下のように場合分けを行う．

- $\sigma^{(n)} \in \Sigma$，$a_1(w_1) \xrightarrow{\sigma} \eta \in R$，$p_1 = w_1[\mathrm{w} \leftarrow p_1']$，$\mathrm{label}_t(p_1') = \sigma$ を満たす η，p_1' が存在する時，h_η を以下のように η の標準形に関する構造的帰納法で定義する．

  IB1, IB2, IB3　適用不能．

  IB4　$\delta \cup \Delta^{(0)} \cup \{\perp\}$ について，$\mathrm{stkeval}_{\mathrm{elm}}(\eta) = \delta$ の時，$h_\eta \stackrel{\mathrm{def}}{=} 1$.

  IS1, IS2　適用不能．

  IS3　$a_2 \in \bigcup_k A_k$ について，$\mathrm{stkeval}_{\mathrm{elm}}(\eta) \stackrel{\mathrm{def}}{=} \mathrm{Head}(\mathrm{Tail}^{m_2}(a_2(w_2)))$ の時，

$$h_\eta \stackrel{\mathrm{def}}{=} h_t(\mathrm{Head}(\mathrm{Tail}^{m_2}(a_2(w_2[\mathrm{w} \leftarrow p_1']))), k).$$

IS4　$n \geq 1$，$\sigma^{(n)} \in \Sigma$ について，$\mathrm{stkeval}_{\mathrm{elm}}(\eta) = \sigma(\eta_1, \ldots, \eta_n)$ の時，

$$h_\eta \stackrel{\mathrm{def}}{=} 1 + \max\{h_{\eta_i} \mid i \in [n]\}.$$

この時，$h_t(\mathrm{Head}(\mathrm{Tail}^m(a(p))), k+1) \stackrel{\mathrm{def}}{=} h_\eta$ とおく．

- それ以外の時，$h_t(\mathrm{Head}(\mathrm{Tail}^m(a(p))), k+1) \stackrel{\mathrm{def}}{=} 0$.

この時，$h_t(\mathrm{Head}(\mathrm{Tail}^m(a(p))), k) \leq k \cdot c_1$ であることは，k に関する数学的帰納法で容易に示せる．また，$h_t(\mathrm{Head}(\mathrm{Tail}^m(a(p))), k) < h_t(\mathrm{Head}(\mathrm{Tail}^m(a(p))), k')$ を満たす k' が存在する時，$G = \mathrm{depgraph}(M)$ として，

- 任意の $0 \leq i \leq k$ について，$m_i < \mathrm{maxavstk}(M)$
- $\mathrm{Head}(\mathrm{Tail}^{m_0}(a_0(p_0))) \Rightarrow^+_{G,t} \cdots \Rightarrow^+_{G,t} \mathrm{Head}(\mathrm{Tail}^{m_k}(a_k(p_k)))$

を満たす $\{\mathrm{Head}(\mathrm{Tail}^{m_i}(a_i(p_i)))\}_{0 \leq i \leq k}$ が存在することも，k に関する数学的帰納法で容易に示せる．

さて，$\mathrm{height}(\mathrm{nf}(\Rightarrow_{M,\sharp(t)}, a(\epsilon))) > c_1 \cdot c_2 \cdot c_3 \cdot \mathrm{size}(\sharp(t))$ となる $t \in \hat{T}_\Sigma$ が存在するとする．この時，$h_t(a(p), c_2 \cdot c_3 \cdot \mathrm{size}(\sharp(t))) < h_t(a(p), k)$ となる k が存在するが，この時，$\mathrm{Head}(\mathrm{Tail}^{m'_1}(a'(p'))) \Rightarrow^+_{G,t} \mathrm{Head}(\mathrm{Tail}^{m'_1}(a'(p')))$ となる $\mathrm{Head}(\mathrm{Tail}^{m'_1}(a'(p'))) \in \mathrm{htform}(\mathrm{attr}(M,t))$ が存在することも容易に示せる．しかしこれは，M が well-defined であることに矛盾する．よって，

$$\begin{aligned}
\mathrm{height}([\![M]\!](t)) &= \mathrm{height}(\mathrm{nf}(\Rightarrow_{M,\sharp(t)}, a(\epsilon))) \\
&\leq c_1 \cdot c_2 \cdot c_3 \cdot \mathrm{size}(\sharp(t)) \\
&= c_1 \cdot c_2 \cdot c_3 + c_1 \cdot c_2 \cdot c_3 \cdot \mathrm{size}(t) \\
&\leq 2 \cdot c_1 \cdot c_2 \cdot c_3 \cdot \mathrm{size}(t) \\
&= c_M \cdot \mathrm{size}(t)
\end{aligned}$$

となる． ■

ここから，StkATT は ATT と同様に，例 135 のような MTT を表現できない．

定理 306. $\mathrm{MTT} \not\subseteq \mathrm{StkATT}$ □

証明. 例 135 の M について，補題 154 と同様に定理 305 から $[\![M]\!] \notin \mathrm{StkATT}$ より． ■

StkTT，StkMTT の樹高性は後ほど示すが，それぞれ TDTT，MTT と同様になることが分かっている．ここから，出力木の高さに関する階層は保たれる．しかし，出力木の高さに関する階層が，直接それぞれの表現力に関する階層を導くとは限らないことも分かっている．この問題については後述するが，予想 301 が大きく関連してくることはここで述べておく．

5.2 木変換器とスタック木変換器の対応

木変換器がスタック木変換器に含まれていることは既に示した．加えて，スタック木変換器は，木変換器より真に大きい．これは，木変換器のクラスに含まれない木変換を，スタック木変換器が表現できることを意味する．Stk は，その 1 つである．木変換器は，定理 160 より RTL の逆像が RTL で閉じている．しかし，Stk は RTL の逆像が RTL で閉じていない場合がある．

命題 307. $\Sigma = \{\top^{(0)}\}$ について，$[\![\mathrm{Stk}_\Sigma]\!]^{-1}[\{\top\}] \notin \mathrm{RTL}$ □

証明. $L = [\![\mathrm{Stk}_\Sigma]\!]^{-1}[\{\top\}]$ とし，$L \in \mathrm{RTL}$ と仮定する．補題 66 の p_L について，$t_1 = \mathrm{Tail}^{p_L}(\mathrm{x}_1)$，$t_2 = \underbrace{\mathrm{Cons}(\bot, \cdots \mathrm{Cons}(\bot}_{p_L \text{ 個}}, \mathrm{Cons}(\top, \mathrm{Empty}))\cdots))$ を考える．$[\![\mathrm{Stk}_\Sigma]\!](t_1(t_2)) = \top$ より，$t_1(t_2) \in L$ である．この時，以下を満たす u_1, u_2, u_3 が存在する．

- $t_1 = u_1(u_2(u_3))$
- $\mathrm{height}(u_2(u_3)) \leq p_L$
- $\mathrm{height}(u_2) \geq 1$
- 任意の $n \in \mathbb{N}$ について，$u_1(u_2^n(u_3(t_2))) \in L$

特に，$u_1(u_3(t_2)) \in L$ である．ところで，$m = \mathrm{height}(u_2)$ とすると，$u_2 = \mathrm{Tail}^m(\mathrm{x}_1)$ であり，$u_1(u_3) = \mathrm{Tail}^{p_L - m}(\mathrm{x}_1)$ である．また，$m \geq 1$ より，$p_L - m < p_L$ であり，つまり $[\![\mathrm{Stk}_\Sigma]\!](u_1(u_3(t_2))) = \bot \notin L$ となるが，これは矛盾．よって，題意は示された． ■

一般に，Stk による RTL の逆像は，CFTL になる．このことを示すため，まずスタックシステムへの逆像が CFTL で閉じていることを示す．

補題 308. ランク付きアルファベット Σ について，以下が成り立つ．

$$\forall L \in \mathcal{P}(\hat{T}_{\Sigma_\bot}) \cap \mathrm{RTL} \,.\, \mathrm{stkeval}^{-1}_{\mathrm{elm}}[L] \in \mathrm{CFTL}$$

□

証明. $L \in \mathcal{P}(\hat{T}_{\Sigma_\bot}) \cap \mathrm{RTL}$ について，$[\![M]\!] = L$ となる FTA $M = (Q, \Sigma_\bot, q_0, R)$ が存在する．この時，PDTA $M' = (Q', \Sigma', \Gamma', q_0', u_0', R')$ を，以下のように定義する．

$Q' \stackrel{\mathrm{def}}{=} Q \cup \{q_\mathrm{B}\} \cup \{q_\mathrm{stk}, q_\mathrm{elm}\}$

$\Sigma' \stackrel{\mathrm{def}}{=} \Sigma \cup \mathrm{S\bar{T}K}$

$\Gamma' \stackrel{\mathrm{def}}{=} \{\mathrm{A}^{(0)}, \mathrm{H}^{(0)}, \mathrm{T}^{(1)}\}$

$q_0' \stackrel{\mathrm{def}}{=} q_0$

$u_0' \stackrel{\mathrm{def}}{=} \mathrm{A}$

$$
\begin{aligned}
R' \stackrel{\text{def}}{=} & \{q_s(\sigma(\mathrm{x}_1,\dots,\mathrm{x}_n),\mathrm{A}) \to \sigma(q_{s_1}(\mathrm{x}_1,\mathrm{A}),\dots,q_{s_n}(\mathrm{x}_n,\mathrm{A})) \mid \sigma:(s_1,\dots,s_n) \rightarrowtriangle_{\dot\Sigma\cup\mathrm{STK}} s\} \\
& \cup \{q(\sigma(\mathrm{x}_1,\dots,\mathrm{x}_n),\mathrm{A}) \to \sigma(q_1(\mathrm{x}_1,\mathrm{A}),\dots,q_n(\mathrm{x}_n,\mathrm{A})) \\
& \qquad \mid q(\sigma(\mathrm{x}_1,\dots,\mathrm{x}_n)) \to \sigma(q_1(\mathrm{x}_1),\dots,q_n(\mathrm{x}_n)) \in R\} \\
& \cup \{q(\mathrm{Head}(\mathrm{x}_1),\mathrm{A}) \to \mathrm{Head}(q(\mathrm{x}_1,\mathrm{H})) \mid q \in Q\} \\
& \cup \{q(\mathrm{Head}(\mathrm{x}_1),\mathrm{A}) \to \mathrm{Head}(q_B(\mathrm{x}_1,\mathrm{H})) \mid q(\bot) \to \bot \in R\} \\
& \cup \{q(\mathrm{Tail}(\mathrm{x}_1),\mathrm{H}) \to \mathrm{Tail}(q(\mathrm{x}_1,\mathrm{T}(\mathrm{H}))) \mid q \in Q \cup \{q_\mathrm{B}\}\} \\
& \cup \{q(\mathrm{Tail}(\mathrm{x}_1),\mathrm{T}(\mathrm{z}_1)) \to \mathrm{Tail}(q(\mathrm{x}_1,\mathrm{T}(\mathrm{T}(\mathrm{z}_1)))) \mid q \in Q \cup \{q_\mathrm{B}\}\} \\
& \cup \{q(\mathrm{Cons}(\mathrm{x}_1,\mathrm{x}_2),\mathrm{H}) \to \mathrm{Cons}(q(\mathrm{x}_1,\mathrm{A}),q_\mathrm{stk}(\mathrm{x}_2,\mathrm{A})) \mid q \in Q\} \\
& \cup \{q(\mathrm{Cons}(\mathrm{x}_1,\mathrm{x}_2),\mathrm{T}(\mathrm{z}_1)) \to \mathrm{Cons}(q_\mathrm{elm}(\mathrm{x}_1,\mathrm{A}),q(\mathrm{x}_2,\mathrm{z}_1)) \mid q \in Q \cup \{q_\mathrm{B}\}\} \\
& \cup \{q_\mathrm{B}(\mathrm{Empty},\mathrm{H}) \to \mathrm{Empty}, q_\mathrm{B}(\mathrm{Empty},\mathrm{T}(\mathrm{z}_1)) \to \mathrm{Empty}\}
\end{aligned}
$$

この時，$\mathrm{stkeval}^{-1}_{\mathrm{elm}}[[\![M]\!]] = [\![M']\!]$ であることは，容易に確認できる． ■

これを元に，一般のランク付き木において，Stk による RTL の逆像が CFTL で閉じていることを示すことができる．

定理 309. ランク付きアルファベット Σ について，以下が成り立つ．

$$\forall L \in \mathcal{P}(\hat{T}_{\Sigma_\bot}) \cap \mathrm{RTL}\,.\,[\![\mathrm{Stk}_\Sigma]\!]^{-1}[L] \in \mathrm{CFTL}$$

□

証明. $L \in \mathcal{P}(\hat{T}_{\Sigma_\bot}) \cap \mathrm{RTL}$ について，$[\![M]\!] = L$ となる FTA $M = (Q, \Sigma_\bot, q_0, R)$ が存在する．この時，補題 308 の PDTA $M' = (Q', \Sigma', \Gamma', q'_0, u'_0, R')$ に対して，$M'' = (Q'', \Sigma', \Gamma', q''_0, u'_0, R'')$ を，以下のように定義する．

$$
\begin{aligned}
Q'' \stackrel{\text{def}}{=} & \; Q' \times \mathrm{SS} \times \mathrm{SS} \cup Q' \times Q' \\
q''_0 \stackrel{\text{def}}{=} & \; \langle q'_0, \mathrm{elm}, \mathrm{elm}\rangle \\
R'' \stackrel{\text{def}}{=} & \{\langle q', s, s'\rangle(\sigma(\mathrm{x}_1,\dots,\mathrm{x}_n),\gamma(\vec{\mathrm{z}})) \to \sigma(\langle q'_1,s_1,s_1\rangle(\mathrm{x}_1,u_1),\dots,\langle q'_n,s_n,s_n\rangle(\mathrm{x}_n,u_n)) \\
& \qquad \mid s,s' \in S, \sigma:(s_1,\dots,s_n) \rightarrowtriangle_{\dot\Sigma\cup\mathrm{STK}} s', \\
& \qquad\quad q(\sigma(\vec{\mathrm{x}}),\gamma(\mathrm{z})) \to \sigma(q'_1(\mathrm{x}_1,u_1),\dots,q'_n(\mathrm{x}_n,u_n)) \in R'\} \\
& \cup \{\langle q'_1,\mathrm{elm},\mathrm{elm}\rangle(\mathrm{x},\gamma(\vec{\mathrm{z}})) \to \langle q'_2,\mathrm{elm},\mathrm{stk}\rangle(\mathrm{x},u) \\
& \qquad \mid q'_1(\mathrm{Head}(\mathrm{x}_1),\gamma(\vec{\mathrm{z}})) \to \mathrm{Head}(q'_2(\mathrm{x}_1,u)) \in R'\} \\
& \cup \{\langle q',\mathrm{stk},\mathrm{stk}\rangle(\mathrm{x},\gamma(\vec{\mathrm{z}})) \to \langle q'_1,q'_2\rangle(\mathrm{x},u) \\
& \qquad \mid q'(\mathrm{Cons}(\mathrm{x}_1,\mathrm{x}_2),\gamma(\vec{\mathrm{z}})) \to \mathrm{Cons}(q'_1(\mathrm{x}_1,\mathrm{A}),q'_2(\mathrm{x}_2,u)) \in R'\} \\
& \cup \{\langle q'_1,q'_2\rangle(\mathrm{x},\gamma(\vec{\mathrm{z}})) \to \langle q'_1,\mathrm{stk},\mathrm{elm}\rangle(\mathrm{x},\mathrm{A}) \\
& \qquad \mid q'_2(\mathrm{Empty},\gamma(\vec{\mathrm{z}})) \to \mathrm{Empty} \in R'\}
\end{aligned}
$$

この時，$[\![\mathrm{Stk}_\Sigma]\!]^{-1}[[\![M]\!]] = [\![M'']\!]$ を示す．これは，

$$[\![\mathrm{Stk}_\Sigma]\!]^{-1}[[\![M]\!]] = \mathrm{stkproj}^{-1}_{\mathrm{elm}}[\mathrm{stkeval}^{-1}_{\mathrm{elm}}[[\![M]\!]]] \qquad (\because \text{定理 209})$$

$$= \mathrm{stkproj}^{-1}_{\mathrm{elm}}[[\![M']\!]] \qquad (\because \text{補題 308})$$

より，$\mathrm{stkproj}^{-1}_{\mathrm{elm}}[[\![M']\!]] = [\![M'']\!]$ と等しい．さらに，$\mathrm{stkproj}^{-1}_{\mathrm{elm}}[[\![M']\!]] = [\![M'']\!]$ が成り立つことは容易に確認できる．よって，題意は示された． ■

さて，Stk は命題 307 より RTL を超える CFTL を逆像に持つことがあるため，木変換器で表現できるとは限らない．

補題 310. $\mathrm{Stk} \not\subseteq \mathrm{MTT}$ □

証明. $\mathrm{Stk} \subseteq \mathrm{MTT}$ とする．この時，定理 160 より，

$$\forall L \in \mathcal{P}(\hat{T}_\Sigma) \cap \mathrm{RTL} \,.\, [\![\mathrm{Stk}_\Sigma]\!]^{-1}[L] \in \mathrm{RTL}$$

である．つまり，$L \overset{\mathrm{def}}{=} \{\mathrm{T}\}$ について，$\mathrm{Stk}^{-1}_{\{\mathrm{T}^{(0)}\}}[L] \in \mathrm{RTL}$ だが，これは命題 307 に矛盾する．よって，題意は示された． ■

Stk はスタック木変換器の最下層にあるクラスであるため，ここからスタック木変換器一般において，木変換器より真に大きいことが示された．ところで，Stk は RTL の逆像が CFTL で閉じていたが，StkTT には CFTL を超える RTL の逆像を持つものが存在する．

定理 311. 以下を満たす $[\![M]\!] : \hat{T}_\Sigma \to \hat{T}_{\Delta_\perp} \in \mathrm{StkTT}$ が存在する．

$$\exists L \in \mathcal{P}(\hat{T}_\Delta) \cap \mathrm{RTL} \,.\, [\![M]\!]^{-1}[L] \notin \mathrm{CFTL}$$

□

証明. 例 203 の $M = (Q, \Sigma, \Delta, e, R)$ について，$L = \{\wedge(\top, \top)\} \in \mathrm{RTL}$ だが，命題 204 より，$[\![M]\!]^{-1}[L] = \{\mathrm{a}^n(\mathrm{b}^n(\mathrm{c}^n(\$))) \mid n \in \mathbb{N}\} \notin \mathrm{CFTL}$．よって，題意は示された． ■

一般に，StkTT と StkMTT の RTL の逆像による言語は，Stk の RTL の逆像による言語と，その言語の TDTT，MTT での逆像による言語で特徴付けられる．なぜなら，StkTT，StkMTT は，TDTT，MTT と Stk の合成によって記述できるからである．

補題 312.

- $\mathrm{StkTT}_\omega \subseteq \mathrm{TDTT}; \mathrm{Stk}_\omega$
- $\mathrm{StkMTT}_\omega \subseteq \mathrm{MTT}; \mathrm{Stk}_\omega$

□

証明. StkMTT $M = (Q, \Sigma, \Delta, e, R)$ について，MTT $M_1 = (Q_1, \Sigma, \Delta_1, e_1, R_1)$ を以下のように定義する．

$$Q_1 \overset{\mathrm{def}}{=} Q$$

$$\Delta_1 \stackrel{\text{def}}{=} \Delta \cup \mathrm{S\bar{T}K}$$
$$e_1 \stackrel{\text{def}}{=} e$$
$$R_1 \stackrel{\text{def}}{=} R$$

$Q = Q^{(1)}$，つまり M が StkTT の場合，M_1 は TDTT になる．この時，$M_2 = \mathrm{Stk}_\Delta$ として，$[\![M]\!]_\omega = [\![M_1]\!]; [\![M_2]\!]_\omega$ を示す．まず，

$$\forall k \in \mathrm{SS}, \eta_1, \eta_2 \in \mathrm{SF}_M(\emptyset, \emptyset)_k \,.\, \eta_1 \Rightarrow_{M_1} \eta_2 \text{ implies } \eta_1 \Rightarrow_M \eta_2$$

で，$\Rightarrow_{M_1}$ は $\bigcup_{k \in \mathrm{SS}} \mathrm{SF}_M(\emptyset, \emptyset)_k$ で閉じていることは，η_1 に対する構造的帰納法で容易に示せる．よって，$t \in \hat{T}_\Sigma$ について，

$$e[\mathrm{x}_1 \leftarrow t] \Rightarrow_M^* \mathrm{nf}(\Rightarrow_{M_1}, e[\mathrm{x}_1 \leftarrow t])$$

である．ここから，$t \in \hat{T}_\Sigma$，$m \in \mathbb{N}$ について，

$$\begin{aligned}
\mathrm{Head}(\mathrm{Tail}^m(e[\mathrm{x}_1 \leftarrow t])) &\Rightarrow_M^* \mathrm{Head}(\mathrm{Tail}^m(\mathrm{nf}(\Rightarrow_{M_1}, e[\mathrm{x}_1 \leftarrow t]))) \\
&\Rightarrow_M^* \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{Head}(\mathrm{Tail}^m([\![M_1]\!](t)))) \\
&= [\![M_2]\!]_\omega([\![M_1]\!](t))(m) \\
&= \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^m(e[\mathrm{x}_1 \leftarrow t]))) \\
&= [\![M]\!]_\omega(t)(m)
\end{aligned}$$

である．よって，題意は示された． ■

また，逆の関係も成り立つ．

補題 313.

- $\mathrm{TDTT}; \mathrm{Stk}_\omega \subseteq \mathrm{StkTT}_\omega$
- $\mathrm{MTT}; \mathrm{Stk}_\omega \subseteq \mathrm{StkMTT}_\omega$

□

証明. MTT $M_1 = (Q_1, \Sigma, \Delta \cup \mathrm{S\bar{T}K}, e_1, R_1)$ に対して，StkMTT $M = (Q, \Sigma, \Delta, e, R)$ を以下のように定義する．

$$Q \stackrel{\text{def}}{=} Q_1$$
$$e \stackrel{\text{def}}{=} \mathrm{stkproj}_{\mathrm{stk}}(e_1)$$
$$R \stackrel{\text{def}}{=} \{q(\sigma(\vec{\mathrm{x}}), \vec{\mathrm{y}}) \to \mathrm{stkproj}_{\mathrm{stk}}(\eta) \mid q(\sigma(\vec{\mathrm{x}}), \vec{\mathrm{y}}) \to \eta \in R\}$$

$Q_1 = Q_1^{(1)}$，つまり M_1 が TDTT の場合，M は StkTT になる．この時，$M_2 = \mathrm{Stk}_\Delta$ として，$[\![M_1]\!]; [\![M_2]\!]_\omega = [\![M]\!]_\omega$ を示す．定理 202 と同様にして，

$$\begin{aligned}\mathrm{Head}(\mathrm{Tail}^m(e[\mathrm{x}_1 \leftarrow t])) &\Rightarrow_M^* \mathrm{stkproj}_{\mathrm{elm}}(\mathrm{Head}(\mathrm{Tail}^m(\mathrm{nf}(\Rightarrow_{M_1}, e_1[\mathrm{x}_1 \leftarrow t]))))\\ &= \mathrm{stkproj}_{\mathrm{elm}}(\mathrm{Head}(\mathrm{Tail}^m([\![M_1]\!](t))))\end{aligned}$$

が示せる．ここから，

$$\begin{aligned}\mathrm{Head}(\mathrm{Tail}^m(e[\mathrm{x}_1 \leftarrow t])) &\Rightarrow_M^* \mathrm{stkproj}_{\mathrm{elm}}(\mathrm{Head}(\mathrm{Tail}^m([\![M_1]\!](t))))\\ &\Rightarrow_M^* \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(\mathrm{Head}(\mathrm{Tail}^m([\![M_1]\!](t)))))\\ &= [\![M_2]\!]_\omega([\![M_1]\!](t))(m)\\ &= \mathrm{nf}(\Rightarrow_M, \mathrm{Head}(\mathrm{Tail}^m(e[\mathrm{x}_1 \leftarrow t])))\\ &= [\![M]\!]_\omega(t)(m)\end{aligned}$$

も容易に示せる．よって，題意は示された． ■

よって，StkTT，StkMTT は，それぞれ TDTT，MTT と Stk の合成と同等の表現力を持つ．

定理 314.

- $\mathrm{StkTT}_\omega = \mathrm{TDTT}; \mathrm{Stk}_\omega$
- $\mathrm{StkMTT}_\omega = \mathrm{MTT}; \mathrm{Stk}_\omega$

□

証明. 補題 312，補題 313 より． ■

これは，木変換器の階層が StkTT，StkMTT でも保存されることを示している．しかし，これは StkATT では成り立たない．なぜなら，StkATT を同様に ATT と Stk に分解した場合，分解された ATT が巡回する場合があるからである．具体的に例 291 を考えると，この例を ATT と思った場合は巡回する．一般に，StkATT が ATT と Stk に分解できるかは未解決であり，本研究では同等ではないと予想している．

予想 315. $\mathrm{ATT}; \mathrm{Stk} \subsetneq \mathrm{StkATT}$ □

ただし，非巡回 StkATT の場合，同様の方法で ATT と Stk に分解できる．

補題 316. $\mathrm{StkATT}_{\mathrm{ncr},\omega} \subseteq \mathrm{ATT}; \mathrm{Stk}_\omega$ □

証明. StkATT $M = (A, \Sigma, \Delta, a_0, \sharp, R)$ について，ATT $M_1 = (A, \Sigma, \Delta \cup \mathrm{S\bar{T}K}, a_0, \sharp, R)$ を考える．この時，$M_2 = \mathrm{Stk}_\Delta$ として，$[\![M]\!]_\omega = [\![M_1]\!]; [\![M_2]\!]_\omega$ は，補題 312 と同様に示せる． ■

また，逆の関係も成り立つ．

補題 317. $\mathrm{ATT}; \mathrm{Stk}_\omega \subseteq \mathrm{StkATT}_{\mathrm{ncr},\omega}$ □

証明. 定理 294 と同様に示せる. ■

よって，非巡回 StkATT は，ATT と Stk の合成と同等の表現力を持つ.

定理 318. $\mathrm{StkATT}_{\mathrm{ncr},\omega} = \mathrm{ATT}; \mathrm{Stk}_{\omega}$ □

証明. 補題 316，補題 317 より. ■

つまり，予想 301 と予想 315 は同値であり，どちらかが肯定的に解決されればもう一方も肯定的に解決される．ところで，以上の結果より，StkTT，非巡回 StkATT，StkMTT の間で，部分関係による階層を為すことを示すことができる.

系 319.

- $\mathrm{StkTT}_{\omega} \subseteq \mathrm{StkATT}_{\mathrm{ncr},\omega}$
- $\mathrm{StkATT}_{\mathrm{ncr},\omega} \subseteq \mathrm{StkMTT}_{\omega}$

□

証明. 定理 314，定理 318，定理 152，定理 155 より. ■

これは, さらに真部分関係に強めることができる．それは, StkTT, 非巡回 StkATT, StkMTT がそれぞれ TDTT，ATT，MTT と同様の樹高性を持つことによる．これを示すために，まず Stk の樹高性を明らかにする．Stk は入力木のスタックに関する記号を全て評価し，$\perp$ 以外は出力に残さない．それに加え，$\perp$ 以外に入力木にない記号を生成することはない．よって，一般に出力木の高さは入力木の高さ以下になる.

定理 320 (Stk の樹高性). ランク付きアルファベット Σ について，以下が成り立つ.

$$\forall t \in \hat{T}_{\Sigma \cup \mathrm{S}\bar{\mathrm{T}}\mathrm{K}} \,.\, \mathrm{height}([\![\mathrm{Stk}_{\Sigma}]\!](t)) \leq \mathrm{height}(t)$$

□

証明. 定理 209 より，題意は

$$\forall t \in \hat{T}_{\Sigma \cup \mathrm{S}\bar{\mathrm{T}}\mathrm{K}} \,.\, \mathrm{height}(\mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(t))) \leq \mathrm{height}(t)$$

と等しい．さらに，これは

$$\forall t \in \hat{T}_{\Sigma \cup \mathrm{S}\bar{\mathrm{T}}\mathrm{K}}, m \in \mathbb{N} \,.\, \mathrm{height}(\mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(\mathrm{Head}(\mathrm{Tail}^{m}(t))))) \leq \mathrm{height}(t)$$

から導かれる．なお，系 210 より，

$$\mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(\mathrm{Head}(\mathrm{Tail}^{0}(t)))) = \mathrm{stkeval}_{\mathrm{elm}}(\mathrm{stkproj}_{\mathrm{elm}}(t))$$

に注意．よって，これを以下のように構造的帰納法で示す．なお，スペースの都合上 height を h，$\mathrm{stkeval}_{\mathrm{elm}}$ を f，$\mathrm{stkproj}_{\mathrm{elm}}$ を p_1，$\mathrm{stkproj}_{\mathrm{stk}}$ を p_2，$t \mapsto \mathrm{Head}(\mathrm{Tail}^m(t))$ を g_m と略記する．

IB1　適用不能．

IB2　以下のように場合分けを行う．

- 任意の $\sigma \in \Sigma^{(0)} \cup \{\bot\}$ について，$t = \sigma$ の場合，

$$\begin{aligned} h(f(p_1(g_m(\sigma)))) &= h(f(g_m(\mathrm{Cons}(\sigma, \mathrm{Empty})))) \\ &= \begin{cases} h(\sigma) & (m = 0) \\ h(\bot) & (m \geq 1) \end{cases} \\ &\leq 1 = h(\sigma) \end{aligned}$$

  より正しい．

- $t = \mathrm{Empty}$ の場合，

$$\begin{aligned} h(f(p_1(g_m(\mathrm{Empty})))) &= h(f(g_m(\mathrm{Empty}))) \\ &= h(\bot) \\ &\leq 1 = h(\mathrm{Empty}) \end{aligned}$$

  より正しい．

IS　以下のように場合分けを行う．

- 任意の $n \geq 1$，$\sigma^{(n)} \in \Sigma$，$t_1, \ldots, t_n \in \hat{T}_{\Sigma \cup \mathrm{S\bar{T}K}}$ について，

$$\begin{aligned} h(f(p_1(g_m(\sigma(t_1, \ldots, t_n))))) &= h(f(g_m(\mathrm{Cons}(\sigma(p_1(t_1), \ldots, p_1(t_n)), \mathrm{Empty})))) \\ &= \begin{cases} h(\sigma(f(p_1(t_1)), \ldots, f(p_1(t_n)))) & (m = 0) \\ h(\bot) & (m \geq 1) \end{cases} \\ &\leq 1 + \max\{h(f(p_1(g_0(t_i)))) \mid i \in [n]\} \\ &\leq 1 + \max\{h(t_i) \mid i \in [n]\} \\ &\qquad (\because \text{i.h.}) \\ &= h(\sigma(t_1, \ldots, t_n)) \end{aligned}$$

  より正しい．

- $t_1, t_2 \in \hat{T}_{\Sigma \cup \mathrm{S\bar{T}K}}$ について，

$$\begin{aligned} h(f(p_1(g_m(\mathrm{Cons}(t_1, t_2))))) &= h(f(g_m(\mathrm{Cons}(p_1(t_1), p_2(t_2))))) \\ &= \begin{cases} h(f(p_1(t_1))) & (m = 0) \\ h(f(p_1(g_{m-1}(t_2)))) & (m \geq 1) \end{cases} \\ &\leq \begin{cases} h(t_1) & (m = 0) \\ h(t_2) & (m \geq 1) \end{cases} \qquad (\because \text{i.h.}) \\ &\leq 1 + \max\{h(t_i) \mid i \in [2]\} \end{aligned}$$

$$= h(\mathrm{Cons}(t_1, t_2))$$

より正しい.

- $t_1 \in \hat{T}_{\Sigma \cup \mathrm{S\bar{T}K}}$ について,

$$\begin{aligned} h(f(p_1(g_m(\mathrm{Head}(t_1))))) &= h(f(g_m(\mathrm{Cons}(\mathrm{Head}(p_1(t_1)), \mathrm{Empty})))) \\ &= \begin{cases} h(f(p_1(g_0(t_1)))) & (m = 0) \\ h(\bot) & (m \geq 1) \end{cases} \\ &\leq \begin{cases} h(t_1) & (m = 0) \\ h(\bot) & (m \geq 1) \end{cases} \qquad (\because \text{i.h.}) \\ &\leq 1 + h(t_1) = h(\mathrm{Head}(t_1)) \end{aligned}$$

より正しい.

- $t_1 \in \hat{T}_{\Sigma \cup \mathrm{S\bar{T}K}}$ について,

$$\begin{aligned} h(f(p_1(g_m(\mathrm{Tail}(t_1))))) &= h(f(p_1(g_{m+1}(t_1)))) \\ &\leq h(t_1) \qquad (\because \text{i.h.}) \\ &\leq h(\mathrm{Tail}(t_1)) \end{aligned}$$

より正しい.

■

ここから，定理 314 を合わせて，StkTT，StkMTT の樹高性が，TDTT，MTT と同様になることが示せる.

系 321 (StkTT と StkMTT の樹高性).

- 任意の $[\![M]\!] : \hat{T}_\Sigma \to \hat{T}_\Delta \in \mathrm{StkTT}$ について,

$$\exists c_M \in \mathbb{N} \,.\, \forall t \in \hat{T}_\Sigma \,.\, \mathrm{height}([\![M]\!](t)) \leq c_M \cdot \mathrm{height}(t)$$

- 任意の $[\![M]\!] : \hat{T}_\Sigma \to \hat{T}_\Delta \in \mathrm{StkMTT}$ について,

$$\exists c_M \in \mathbb{N} \,.\, \forall t \in \hat{T}_\Sigma \,.\, \mathrm{height}([\![M]\!](t)) \leq \exp(c_M \cdot \mathrm{height}(t))$$

がそれぞれ成り立つ. □

証明. それぞれ定理 314 より,

- $[\![M]\!] = [\![M_1]\!]; [\![M_2]\!]$ を満たす $[\![M_1]\!] \in \mathrm{TDTT}$，$[\![M_2]\!] \in \mathrm{Stk}$ について，定理 94 の c_{M_1} に対して $c_M \stackrel{\mathrm{def}}{=} c_{M_1}$，$f \stackrel{\mathrm{def}}{=} n \mapsto c_{M_1} \cdot n$
- $[\![M]\!] = [\![M_1]\!]; [\![M_2]\!]$ を満たす $[\![M_1]\!] \in \mathrm{MTT}$，$[\![M_2]\!] \in \mathrm{Stk}$ について，定理 134 の c_{M_1} に対して $c_M \stackrel{\mathrm{def}}{=} c_{M_1}$，$f \stackrel{\mathrm{def}}{=} n \mapsto \exp(c_{M_1} \cdot n)$

とおく．この時，それぞれ任意の $t \in \hat{T}_\Sigma$ について，

$$\begin{aligned}\mathrm{height}([\![M]\!](t)) &= \mathrm{height}([\![M_2]\!]([\![M_1]\!](t))) \\ &\leq \mathrm{height}([\![M_1]\!](t)) && (\because \text{定理 320}) \\ &\leq f(\mathrm{height}(t))\end{aligned}$$

となる． ■

もちろん，非巡回 StkATT の樹高性についても同様に示せるが，これは定理 305 と同様の結果になる．それぞれの樹高性から，StkTT，非巡回 StkATT，StkMTT のクラスについて，逆の包含関係は成り立たないことが示された．

補題 322.

- $\mathrm{StkATT}_{\mathrm{ncr}} \not\subseteq \mathrm{StkTT}$
- $\mathrm{StkMTT} \not\subseteq \mathrm{StkATT}_{\mathrm{ncr}}$

□

証明. それぞれ，

- $\mathrm{StkATT}_{\mathrm{ncr}} \subseteq \mathrm{StkTT}$
- $\mathrm{StkMTT} \subseteq \mathrm{StkATT}_{\mathrm{ncr}}$

が成り立つと仮定する．この時，

- 定理 294 より，$\mathrm{ATT} \subseteq \mathrm{StkATT}_{\mathrm{ncr}} \subseteq \mathrm{StkTT}$
- 定理 267 より，$\mathrm{MTT} \subseteq \mathrm{StkMTT} \subseteq \mathrm{StkATT}_{\mathrm{ncr}}$

であり，

- 例 99 の M について，$[\![M]\!] \in \mathrm{StkTT}$
- 例 135 の M について，$[\![M]\!] \in \mathrm{StkATT}_{\mathrm{ncr}}$

である．しかし，これはそれぞれ，補題 151，補題 154 と同様に，系 321，定理 305 に矛盾する．よって，題意は示された． ■

以上から，StkTT，非巡回 StkATT，StkMTT の間で，真の部分関係による階層を為すことを示すことができる．

定理 323.

- $\mathrm{StkTT} \subsetneq \mathrm{StkATT}_{\mathrm{ncr}}$
- $\mathrm{StkATT}_{\mathrm{ncr}} \subsetneq \mathrm{StkMTT}$

□

証明. 系 319，補題 322 より． ■

推移律から，StkTT は StkMTT より真に小さいことも示された．

系 324. StkTT $\subsetneq$ StkMTT □

証明. 定理 323 より． ■

系 207 より，Stk はスタック木変換器の階層において，最下層にあたる．この Stk が補題 310 より MTT に含まれないことと以上の結果から，以下の関係が導かれる．

補題 325.

- StkTT $\not\subseteq$ TDTT
- $\mathrm{StkATT}_{\mathrm{ncr}} \not\subseteq$ ATT
- StkMTT $\not\subseteq$ MTT

□

証明. それぞれ

- StkTT $\subseteq$ TDTT
- $\mathrm{StkATT}_{\mathrm{ncr}} \subseteq$ ATT
- StkMTT $\subseteq$ MTT

が成り立つと仮定する．この時，定理 323 より，

- 系 156 より，StkTT $\subseteq$ TDTT $\subseteq$ MTT
- 定理 155 より，StkTT $\subseteq \mathrm{StkATT}_{\mathrm{ncr}} \subseteq$ ATT $\subseteq$ MTT
- StkTT $\subseteq$ StkMTT $\subseteq$ MTT

である．また系 207 より，それぞれ Stk $\subseteq$ StkTT $\subseteq$ MTT となるが，これは補題 310 に矛盾する．よって，題意は示された． ■

以上より，スタック木変換器は木変換器より真に大きいことが示された．

定理 326.

- TDTT $\subsetneq$ StkTT
- ATT $\subsetneq \mathrm{StkATT}_{\mathrm{ncr}}$
- MTT $\subsetneq$ StkMTT

□

証明. 定理 202，定理 294，定理 267，補題 325 より． ■

また，$\mathrm{StkATT}_{\mathrm{ncr}} \subseteq \mathrm{StkATT}$ より，StkATT は ATT より真に大きいことも分かる．

系 327. $\mathrm{ATT} \subsetneq \mathrm{StkATT}$ □

証明. 定理 326 より． ■

一般に，定理 311 に見られるように，スタック木変換器では RTL の逆像が RTL に閉じていない木変換を表現できるが，そのようなものは定理 160 に違反するため木変換器では表現できない．ただし，RTL の逆像が RTL に閉じている木変換で，木変換器では表現できないが，スタック木変換器では表現できるものが存在するかという問題は，未解決である．予想では，RTL の逆像が RTL で閉じている木変換で，スタック木変換器で表現できるものは，木変換器でも表現できる．

予想 328. $[\![M]\!] : \hat{T}_{\Sigma} \to \hat{T}_{\Delta_\perp} \in \mathrm{StkMTT}$ について，以下の 2 条件は同値である．

- $[\![M]\!] \in \mathrm{MTT}$
- $\forall L \in \mathcal{P}(\hat{T}_{\Delta_\perp}) \cap \mathrm{RTL} \,.\, [\![M]\!]^{-1}[L] \in \mathrm{RTL}$

□

ところで，線形 StkTT が，線形 TDTT と Stk に分解できることも，定理 314 の分解方法を適用することで示せる．

定理 329. $\mathrm{StkTT}_{\mathrm{lu},\omega} = \mathrm{TDTT}_{\mathrm{lu}}; \mathrm{Stk}_{\omega}$ □

証明. M が線形 StkTT の場合，補題 312 の M_1 は線形 TDTT．また，M_1 が線形 TDTT の場合，補題 313 の M は線形 StkTT．よって，題意は示された． ■

ここから，以下が導かれる．

定理 330. 予想 163 が成り立つならば，$[\![M]\!] : \hat{T}_{\Sigma} \to \hat{T}_{\Delta_\perp} \in \mathrm{StkTT}_{\mathrm{lu}}$ について，以下が成り立つ．

$$\forall L \in \mathcal{P}(\hat{T}_{\Delta_\perp}) \cap \mathrm{RTL} \,.\, [\![M]\!]^{-1}[L] \in \mathrm{CFTL}$$

□

証明. 定理 329 より，$[\![M]\!] = [\![M_1]\!]; [\![M_2]\!]$ を満たす $[\![M_1]\!] \in \mathrm{TDTT}_{\mathrm{lu}}$，$[\![M_2]\!] \in \mathrm{Stk}$ が存在する．この時，$L \in \mathcal{P}(\hat{T}_{\Delta_\perp}) \cap \mathrm{RTL}$ について，定理 309 より $[\![M_2]\!]^{-1}[L] \in \mathrm{CFTL}$ であり，予想 163 より $[\![M]\!]^{-1}[L] = [\![M_1]\!]^{-1}[[\![M_2]\!]^{-1}[L]] \in \mathrm{CFTL}$．よって，題意は示された． ■

一般に，スタック木変換器において RTL の逆像が CFTL に閉じているクラスの一つとして，線形 StkTT が該当するのではないかと考えている．これは，定理 329 から，線形 TDTT が定理 309 の証明で与えられた文脈自由文法による言語の逆像について，CFTL で閉じていることを意味する．

予想 331. $[\![M]\!] : \hat{T}_{\Sigma} \to \hat{T}_{\Delta_\perp} \in \mathrm{StkTT}_{\mathrm{lu}}$ について，以下が成り立つ．

$$\forall L \in \mathcal{P}(\hat{T}_{\Delta_\perp}) \cap \mathrm{RTL} \,.\, [\![M]\!]^{-1}[L] \in \mathrm{CFTL}$$

□

5.3 木変換器と異なるスタック木変換器の性質

樹高性より，StkATT では表現できない木変換を持つ StkMTT が存在する．

系 332. $\mathrm{StkMTT} \not\subseteq \mathrm{StkATT}$ □

証明. 定理 306，定理 326 より． ■

その逆もやはり，木変換クラスの部分クラス関係は成立しないのではないかという予想がある．

予想 333. $\mathrm{StkATT} \not\subseteq \mathrm{StkMTT}$ □

この根拠は，次の予想である．

予想 334. 例 251 の M について，$[\![M]\!] \notin \mathrm{StkMTT}$ □

例 251 の StkATT は，素直に構造的帰納的に定義できない．しかし，StkMTT はスタック機構を持つとは言え構造的帰納的操作しか行えない．このため，直感的には例 251 は，StkMTT で表現できそうにはない．しかしながら，この予想は本研究では証明に至れていない．計算の複雑性のクラスとして，StkMTT のクラスで許容される計算では書けないのではないかと考えているが，現状，どのような指標による複雑性の違いかについては分かっていない．なお，この予想が肯定的に解決されれば，予想 302 も肯定的に解決される．

もし，予想 333 が成り立つならば，幾つかの木変換のクラスに関する問題が解決する．まず，StkATT が ATT と Stk に分解できないことが導かれる．

補題 335. 予想 333 が成り立つならば，$\mathrm{StkATT} \not\subseteq \mathrm{ATT};\mathrm{Stk}$ □

証明. もし，$\mathrm{StkATT} \subseteq \mathrm{ATT};\mathrm{Stk}$ ならば，定理 155，定理 314 より

$$\mathrm{StkATT} \subseteq \mathrm{ATT};\mathrm{Stk} \subseteq \mathrm{MTT};\mathrm{Stk} = \mathrm{StkMTT}$$

となるが，これは予想 333 に矛盾する．よって，題意は示された． ■

ここから，$\mathrm{StkATT}_{\mathrm{ncr}} \subseteq \mathrm{StkATT}$ の逆の関係は成り立たないことが分かる．

系 336. 予想 333 が成り立つならば，$\mathrm{StkATT} \not\subseteq \mathrm{StkATT}_{\mathrm{ncr}}$ □

証明. 定理 318，補題 335 より． ■

これは，StkATT と非巡回 StkATT が木変換のクラスとして一致しないことを意味する．

定理 337. 予想 333 が成り立つならば，$\mathrm{StkATT}_{\mathrm{ncr}} \subsetneq \mathrm{StkATT}$ □

証明. 系 336 より． ■

もうひとつ解決する問題として，stayStkMTT に関する問題がある．まず，木変換器理論の結果と同様，StkATT と stayStkMTT のクラスはこの順に部分クラス関係が成り立つ．

補題 338. $\mathrm{StkATT}_{\omega} \subseteq \mathrm{stayStkMTT}_{\omega}$ □

証明. StkATT $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，$A_{\mathrm{inh}} = \{b_1, \ldots, b_m\}$ のように継承属性を $[m] = [|A_{\mathrm{inh}}|]$ で整列させ，$r = \mathrm{maxrank}(\Sigma)$ とおく．この時，$\mathrm{stayStkMTT}$ $M' = (Q', \Sigma, \Delta, e', R')$ を以下のように定義する．

$$
\begin{aligned}
Q &\stackrel{\mathrm{def}}{=} \{a^{(m+1)} \mid a \in A_{\mathrm{syn}}\} \cup \{\langle a, i\rangle^{(m+1)} \mid a \in A_{\mathrm{syn}}, i \in [r]\} \\
e' &\stackrel{\mathrm{def}}{=} \mathrm{conv}_{\sharp,1}(\eta_0) \\
R' &\stackrel{\mathrm{def}}{=} \{a(\mathrm{x} = \sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \mathrm{y}_1, \ldots, \mathrm{y}_m) \to \mathrm{conv}_{\sigma,n}(\eta) \mid \sigma^{(n)} \in \Sigma, a(\mathrm{w}) \xrightarrow{\sigma} \eta \in R\} \\
&\quad \cup \{\langle a, i\rangle(\mathrm{x} = \sigma(\mathrm{x}_1, \ldots, \mathrm{x}_n), \mathrm{y}_1, \ldots, \mathrm{y}_m) \to \mathrm{iconv}_{\sigma,n}(a, i) \mid \sigma^{(n)} \in \Sigma, i \in [r]\}
\end{aligned}
$$

ただし，η_0 は $a_0(\mathrm{w}) \xrightarrow{\sharp} \eta_0 \in R$ を満たすものであり，$\mathrm{conv}_{\sigma,n} : \mathrm{RHS}_M(\{\mathrm{w}\}, [n])_k \to \mathrm{RHS}_{M'}(\{\mathrm{x}\} \cup \mathrm{X}_n, \mathrm{Y}_m)_k$ は，以下のように $U \stackrel{\mathrm{def}}{=} \mathrm{RHS}_M(\{\mathrm{w}\}, [n])$ に関する構造的帰納法で定義する．

IB1 $a \in A_{\mathrm{syn}}$，$i \in [n]$ について，$\mathrm{conv}_{\sigma,n}(a(\mathrm{w}i)) \stackrel{\mathrm{def}}{=} \langle a, i\rangle(\mathrm{x}, \mathrm{y}_1, \ldots, \mathrm{y}_m)$.

IB2 $j \in [m]$ について，$\mathrm{conv}_{\sigma,n}(b_j(\mathrm{w})) \stackrel{\mathrm{def}}{=} \mathrm{y}_j$.

IB3 $\delta \in \Delta^{(0)} \cup \{\bot, \mathrm{Empty}\}$ について，$\mathrm{conv}_{\sigma,n}(\delta) \stackrel{\mathrm{def}}{=} \delta$.

IS $l \geq 1$，$\delta : (k_1, \ldots, k_l) \to_{\dot{\Delta} \cup \{\mathrm{Head}, \mathrm{Tail}, \mathrm{Cons}\}} k$，$\eta_1 \in U_{k_1}, \ldots, \eta_l \in U_{k_l}$ について，

$$\mathrm{conv}_{\sigma,n}(\delta(\eta_1, \ldots, \eta_l)) \stackrel{\mathrm{def}}{=} \delta(\mathrm{conv}_{\sigma,n}(\eta_l), \ldots, \mathrm{conv}_{\sigma,n}(\eta_l)).$$

また，

$$
\mathrm{iconv}_{\sigma,n}(a, i) \stackrel{\mathrm{def}}{=} \begin{cases} \mathrm{Empty} & (i > n) \\ a(\mathrm{x}_i, \mathrm{conv}_{\sigma,n}(\eta_1), \ldots, \mathrm{conv}_{\sigma,n}(\eta_m)) & \\ \qquad (i \in [n], \forall j \in [m] \,.\, b_j(\mathrm{w}i) \xrightarrow{\sigma} \eta_j \in R) & \end{cases}
$$

と定義する．この時，$[\![M]\!] = [\![M']\!]$ であることは，補題 153 と同様に示せる． ■

逆は，樹高性より成り立たない．

補題 339. stayStkMTT $\not\subseteq$ StkATT □

証明. stayStkMTT $\subseteq$ StkATT と仮定すると，定理 284 より

$$\text{StkMTT} \subseteq \text{stayStkMTT} \subseteq \text{StkATT}$$

となるが，これは系 332 に矛盾する．よって，題意は示された． ■

よって，StkATT と stayStkMTT のクラスはこの順に真の部分集合関係が成り立つ．

定理 340. StkATT $\subsetneq$ stayStkMTT □

証明. 補題 338，補題 339 より． ■

この時，予想 333 を仮定すると，stayStkMTT の滞在拡張は除去できないことが分かる．

補題 341. 予想 333 が成り立つならば，stayStkMTT $\not\subseteq$ StkMTT □

証明. もし，stayStkMTT $\subseteq$ StkMTT が成り立つならば，定理 340 より

$$\text{StkATT} \subseteq \text{stayStkMTT} \subseteq \text{StkMTT}$$

となるが，これは予想 333 に矛盾する．よって，題意は示された． ■

これは，木変換器理論の結果と異なり，stayStkMTT は StkMTT より木変換のクラスとして真に大きいことを意味する．

定理 342. 予想 333 が成り立つならば，StkMTT $\subsetneq$ stayStkMTT □

証明. 定理 284，補題 341 より． ■

また，StkATT と同じく stayStkMTT も stayMTT と Stk に分解できないことも分かる．

定理 343. 予想 333 が成り立つならば，stayStkMTT $\not\subseteq$ stayMTT; Stk □

証明. もし，stayStkMTT $\subseteq$ stayMTT; Stk が成り立つならば，定理 158，定理 314 より

$$\text{stayStkMTT} \subseteq \text{stayMTT}; \text{Stk} = \text{MTT}; \text{Stk} = \text{StkMTT}$$

となるが，これは定理 342 に矛盾する．よって，題意は示された． ■

以上の結果は，スタック木変換のクラスとしては明らかになっている．まず，StkATT と StkMTT はスタック木変換のクラスとしては比較不能である．

定理 344. $\mathrm{StkATT}_\omega \not\subseteq \mathrm{StkMTT}_\omega$ □

証明. $\mathrm{StkATT}_\omega \subseteq \mathrm{StkMTT}_\omega$ が成り立つと仮定した時，例 251 の $M = (A, \Sigma, \Delta, \sharp, a_0, R)$ について，$[\![M]\!]_\omega \in \mathrm{StkMTT}_\omega$ であり定理 269 より，$t \in \hat{T}_\Sigma$ について $[\![M]\!]_\omega(t)(n) = \bot$ を満たす $n \in \mathbb{N}$ が存在する．しかし，これは命題 253 と矛盾する．よって，題意は示された． ■

また，stayStkMTT は StkMTT にはスタック木変換のクラスとしては含まれない．

補題 345. $\mathrm{stayStkMTT}_\omega \not\subseteq \mathrm{StkMTT}_\omega$ □

証明. $\mathrm{stayStkMTT}_\omega \subseteq \mathrm{StkMTT}_\omega$ と仮定すると，補題 338 より，$\mathrm{StkATT}_\omega \subseteq \mathrm{stayStkMTT}_\omega \subseteq \mathrm{StkMTT}_\omega$ だが，これは定理 344 に矛盾する．よって，題意は示された． ■

これは，stayStkMTT が StkMTT より，スタック木変換のクラスとしては真に大きいことを意味する．

定理 346. $\mathrm{StkMTT}_\omega \subsetneq \mathrm{stayStkMTT}_\omega$ □

証明. 定理 284，補題 345 より． ■

ところで，ATT の場合，非巡回であれば同じ属性とノードの組は自身の結果に依存しないが，StkATT の場合は例 291 の例に見られるように fully-defined であっても自身の結果に依存する場合がある．しかし，fully-defined であれば，自身の結果を参照する回数は，$\mathrm{maxavstk}(M)$ で抑えられる．これは，定理 287 より，自身の結果に依存するような Tail-簡約が存在しないため，非螺旋性を満たすよう参照するにはスタックの位置を毎回変えなければならず，変えられるスタックの位置が $\mathrm{maxavstk}(M)$ 以下であるからである．つまり，ATT から MTT への変換を拡張することで，fully-defined StkATT は StkMTT に変換できるのではないかと考えている．しかし，これはまだ正当性が示せておらず，未解決である．

予想 347. $\mathrm{StkATT}_{\mathrm{fd},\omega} \subseteq \mathrm{StkMTT}_\omega$ □

証明. ATT から MTT への変換において，conv，iconv のとる引数 C に何回目の参照かの情報を付け加え，参照が $\mathrm{maxavstk}(M)$ を超える場合にその部分を Empty に書き換える．この変換の正当性は，上の観察から示せる． ■

ここから，例 251 は fully-defined StkATT でも表せないという予想が立つ．

予想 348. 例 251 の M について，$[\![M]\!] \notin \mathrm{StkATT}_{\mathrm{fd}}$ □

もし，この予想が成り立つならば，定理 287 より自身に返ってくるような Tail-簡約を持つ場合と持たない場合では，異なる計算クラスに属することを意味する．これは，非螺旋性からも分かるように，Tail-簡約が StkATT に対し，ATT に見られない性質を与えているのではないかと考えられる．

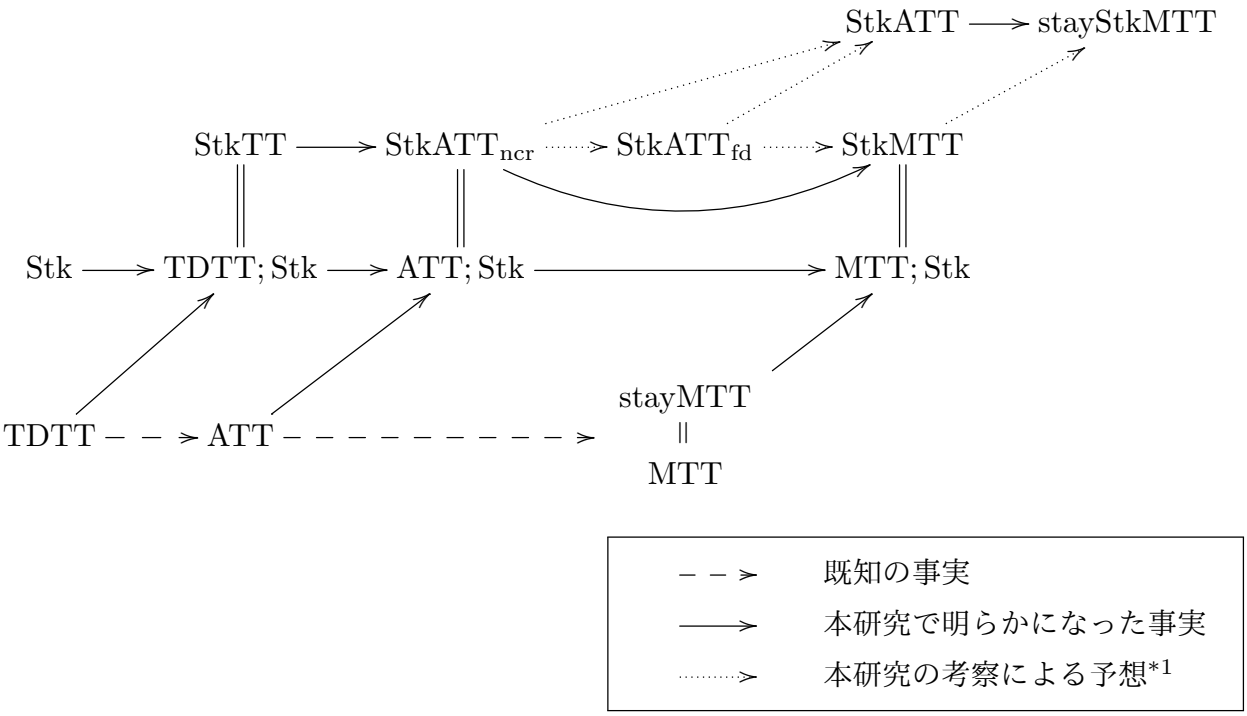


図 5.1　変換クラスの階層 (矢印は $\subsetneq$ を示す)

以上から，最終的に得られる木変換クラスの関係を，予想も含めて表すと，図 5.1 のようになる．

*1 ただし，StkATT$_{\mathrm{fd}} \subseteq$ StkMTT を除いた $\subseteq$ の関係は，本研究で明らかになっている．

第 6 章　関連研究

定理 161，定理 311 では，それぞれ木変換器，スタック木変換器で CFTL，RTL の逆像が，CFTL に閉じていないことについて述べた．これに関連することとして，PDTA において幾つかの拡張は**チューリング機械** (*turing machine*) を模倣できる [Gue83][CDGV94] ことが知られている．ひとつの拡張として，規則において，以下のようなプッシュダウン木の先読みを付けることを許す拡張を考える．

$$q(\sigma(\mathrm{x}_1), \gamma(\gamma_1(\mathrm{z}_1), \gamma_2(\mathrm{z}_2))) \to \sigma(q(\mathrm{x}_1, \gamma(\mathrm{z}_1, \mathrm{z}_2)))$$

この拡張は，チューリング機械を模倣できることが知られている [Gue83]．また，別の拡張として，規則において，以下のように複数のプッシュダウン木をとることを許す拡張を考える．

$$q(\sigma(\mathrm{x}_1), \gamma_1(\mathrm{z}_1), \gamma_2(\mathrm{z}_2)) \to \sigma(q(\mathrm{x}_1, \mathrm{z}_1, \mathrm{z}_2))$$

この拡張もやはり，チューリング機械を模倣できることが知られている [CDGV94]．スタック木変換器の RTL による逆像は，これらの拡張と同等の機能を要請することから，それを受理する機械は同様にチューリング機械を模倣できることが予想される．これは，木変換器やスタック木変換器が一般には有力な CFTL，RTL の逆像型検査 [FH07] のアルゴリズムを持たないことを意味する．しかし，予想 163，予想 331 に見られるように，線形性を要求すると逆像が CFTL に収まる可能性がある．これらが成り立つなら，線形性を持つ木変換器，スタック木変換器に限っては CFTL，RTL の逆像型検査を構築することが可能になる．

RTL の逆像が RTL を超える木変換を表現する，木変換器の別の拡張として**プッシュダウン木変換器** (*pushdown tree transducer*) [Yam00] がある．プッシュダウン木変換器では，これは，PDTA を直接木変換器に拡張したものである．プッシュダウン木変換器の RTL に対する逆像は，スタック木変換器と同様 CFTL を超える．これは，スタック木変換器の場合と同様，CFTL の積を RTL の逆像で表現できるからである．よって，プッシュダウン木変換器においても逆像型検査アルゴリズムを考える際，スタック木変換器と同様の問題が生まれる．さらに，線形性を要求すると逆像が CFTL に収まる可能性があることについても同様である．しかし，プッシュダウン木変換器は，全域的決定的な場合表現力が非常に限られる．特に，スタック木変換器のように部分木に対して進めた簡約の結果をスタック操作で加工するということができないため，同様のことをする場合大きな工夫を要する．これはストリーム処理をモデル化する際，プッシュダウン木変換器よりもスタック木変換器がより直感的なモデルとなることを示唆している．逆にスタック木変換器は，部分木に対する出力を加工しない場合，扱えるプッシュダウン木がスタックシステムだけに限定されたプッシュダウン木変換器と捉えられる．このような制約を満たしうる木変換に対しては，プッシュダウン木変換器の方がより直感的なモデルとなる．また，両者の表現力の関係性は未解決であり，今後の課題として挙げられるだろう．

また，スタック木変換器と似た機能を持つ木変換器の拡張として，**ストリーム木変換器** (*streaming tree transducer*) [AD17] がある．ストリーム木変換器は，入出力木を**ネストされた文字列** (*nested word*) として扱い，スタックに対し文字列の出し入れを行いながら入力を辿って行き，その際出力を構築していく計算モデルになっている．ストリーム木変換器は，木変換のクラスとしてはマクロ木変換器より小さいが，XML 文書のストリーム処理に直接対応する計算モデルとして有用であり，その応用が期待されている．特にストリーム処理では入力を一度しか辿らない制約が重要になり，ストリーム木変換器はこの制約を満たすよう設計されている．スタックマクロ木変換器は，線形制約を満たす範囲でもマクロ木変換器で表現できない木変換を表せる．ここから，スタックマクロ木変換器は，ストリーム木変換器では扱えないストリーム処理の計算モデルとなりえる．特に，ストリーム木変換器は入力の文書に対してネストされた文字列としての解釈を必要とするが，スタック木変換器では直接扱うことができる．これらの考察から，スタックマクロ木変換器が，ストリーム処理に対する計算モデルの面でもストリーム木変換器より有用な場面があると考えている．また，他の拡張として**多値返し木変換器** (*multi-return tree transducer*) [IHM08] がある．これは木変換器に対して，入力木の走査の際，一度に複数の出力木を返せるように拡張した木変換器である．この木変換器は，全域的決定的な場合クラスとしてマクロ木変換器と一致する．多値返し木変換器の機能は，固定長のスタックを持つスタック木変換器によって表現できる．スタック木変換器は，それに加えて可変長の返り値も持て，またマクロ木変換器よりクラスとして真に大きいため，機能としても，クラスとしても，多値返し木変換器を真に含んでいると言えるだろう．

第 7 章　おわりに

本研究では，中野 [Nak09] が示した属性付き木変換器の拡張を元に，トップダウン木変換器，マクロ木変換器にも同様の拡張を行い，その表現力を比較した．また，それらの拡張，スタック木変換器で，木変換器理論の結果の多くが流用できることを示した．トップダウン木変換器，属性付き木変換器，マクロ木変換器のスタック木変換器への拡張，スタックトップダウン木変換器，スタック属性付き木変換器，スタックマクロ木変換器において，スタックトップダウン木変換器，非巡回スタック属性付き木変換器，スタックマクロ木変換器の間には，木変換器理論で見られた真の部分集合関係が成り立つことが明らかになった．また，これらは元になった木変換器とスタックシステム評価器に分解でき，同様の樹高性を持つことを示した．また，スタック属性付き木変換器が，他のスタック木変換器と異なり，深さが有限でないスタックを出力できること，属性付き木変換器と異なる停止性のための条件を持つことを示した．また，ここからスタック木変換のクラスとして，スタック属性付き木変換器とスタックマクロ木変換器が比較不能であること，滞在付きスタックマクロ木変換器について木変換器の場合と異なり滞在拡張が除去できないことを示した．これらの結果は，スタック木変換器について [VK04] の手法を応用するのに有用である期待している．

スタック属性付き木変換器の well-defined 性の決定可能性，スタック属性付き木変換器とスタックマクロ木変換器の木変換クラスとしての比較不能性は，本研究では示すに至らず，未解決問題となった．また，正規木言語の逆像が文脈自由木言語に閉じているスタック木変換器のクラスや，一般のスタック木変換器の正規木言語の逆像がどの木言語クラスに属するかについても，まだ明らかになっていない．これらの問題を解決することが，今後の課題である．

参考文献

[AD17] Rajeev Alur and Loris D'antoni, *Streaming Tree Transducers*, Journal of the ACM **64** (2017), no. 5, 1–55.

[CDGV94] Jean Luc Coquidé, Max Dauchet, Rémi Gilleron, and Sándor Vágvölgyi, *Bottom-up tree pushdown automata: classification and connection with rewrite systems*, Theoretical Computer Science **127** (1994), no. 1, 69–98.

[CLRS09] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein, *Introduction to Algorithms*, 3rd ed., The MIT Press, 2009.

[EM03a] Joost Engelfriet and Sebastian Maneth, *A comparison of pebble tree transducers with macro tree transducers*, Acta Informatica **39** (2003), no. 9, 613–698.

[EM03b] Joost Engelfriet and Sebastian Maneth, *Macro Tree Translations of Linear Size Increase are MSO Definable*, SIAM Journal on Computing **32** (2003), no. 4, 950–1006.

[Eng80] Joost Engelfriet, *SOME OPEN QUESTIONS AND RECENT RESULTS ON TREE TRANSDUCERS AND TREE LANGUAGES*, Formal Language Theory, Elsevier, 1980, pp. 241–286.

[Eng15] Joost Engelfriet, *Tree Automata and Tree Grammars*, oct 2015.

[EV85] Joost Engelfriet and Heiko Vogler, *Macro tree transducers*, Journal of Computer and System Sciences **31** (1985), no. 1, 71–146.

[FH07] Alain Frisch and Haruo Hosoya, *Towards Practical Typechecking for Macro Tree Transducers*, Database Programming Languages (2007), 246–260.

[Fül81] Zoltán Fülöp, *On attributed tree transducers*, Acta Cybernetica **5** (1981), no. 3, 261–279.

[FV98] Zoltán Fülöp and Heiko Vogler, *Syntax-Directed Semantics: Formal Models Based on Tree Transducers*, 1st ed., Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.

[Gue83] Inène Guessarian, *Pushdown tree automata*, Mathematical Systems Theory **16** (1983), no. 1, 237–263.

[HMU00] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman, *Introduction to automata theory, languages, and computation*, 2nd ed., Addison-Wesley, 2000.

[IHM08] Kazuhiro Inaba, Haruo Hosoya, and Sebastian Maneth, *Multi-Return Macro Tree Transducers*, Implementation and Applications of Automata, vol. 5148 LNCS, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 102–111.

[Knu68] Donald E Knuth, *Semantics of context-free languages*, Mathematical Systems Theory **2** (1968), no. 2, 127–145.

[Küh98] Armin Kühnemann, *Benefits of Tree Transducers for Optimizing Functional Programs*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 1530 LNCS, 1998, pp. 146–157.

[MSV03] Tova Milo, Dan Suciu, and Victor Vianu, *Typechecking for XML transformers*, Journal of Computer and System Sciences **66** (2003), no. 1, 66–97.

[Nak09] Keisuke Nakano, *Composing Stack-Attributed Tree Transducers*, Theory of Computing Systems **44** (2009), no. 1, 1–38.

[NN01] Keisuke Nakano and Susumu Nishimura, *Deriving Event-Based Document Transformers from Tree-Based Specifications*, Electronic Notes in Theoretical Computer Science **44** (2001), no. 2, 181–205.

[Pos46] Emil L. Post, *A variant of a recursively unsolvable problem*, Bulletin of the American Mathematical Society **52** (1946), no. 4, 264–269.

[Rou68] William C. Rounds, *Trees, transducers and transformations*, phd, Stanford University, 1968.

[VK04] Janis Voigtländer and Armin Kühnemann, *Composition of functions with accumulating parameters*, Journal of functional Programming **14** (2004), no. 3, 317–363.

[Yam00] Katsunori Yamasaki, *Some Notes on Domain Tree Languages of Top-Down Pushdown Tree Transducers*, IEICE Transactions on Information and Systems **E83D** (2000), no. 9, 1713–1720.

謝辞

本研究を行うにあたり，終始変わらぬご指導を賜りました岩崎英哉教授に感謝致します．また，指導委託という形でご指導頂きました東北大学の中野圭介教授，本論文の執筆に関して助言や指摘をして頂いた東北大学の浅田和之助教に感謝致します．最後に，岩崎研究室及び旧電気通信大学中野研究室，東北大学中野研究室の皆様には，本研究について有益なご意見を頂き大変お世話になりました．改めて，御礼申し上げます．