平成 29 年度修士論文

属性付き記号的木変換器の合成

電気通信大学
大学院情報理工学研究科
コンピュータサイエンスプログラム

学籍番号　　: 1631102
氏名　　: 中川 涼太
主任指導教員 : 中野 圭介 准教授
指導教員　　: 岩崎 英哉 教授
提出日　　: 2018 年 1 月 29 日

要旨

関数プログラミングでは，関数間で受け渡される中間的な計算結果に関するコストを削減するため，2 つの関数の連続する適用を，中間的な計算結果を介さない 1 つの関数の適用で置き換える関数融合と呼ばれる最適化が重要である．蓄積引数を伴う再帰プログラムを関数融合により最適化するには，属性付き木変換器の合成アルゴリズムを応用する手法が有効であることが知られている．しかしこの手法は，2 つの関数がいずれもパターンマッチのガード式を扱う場合には対応していない．また，従来の木変換器は，有限の領域しか扱えないため，無限の領域に関する条件を記述したガード式を扱うことができない．これらの問題点を解決するため，本論文では，最も基本的な木変換器を無限の領域に対応させた記号的木変換器と同様の拡張手法を，属性付き木変換器に対して行うことで，無限の領域を扱うことが可能な属性付き記号的木変換器を提案する．そして，属性付き記号的木変換器で表現できる計算のクラスが記号的木変換器のクラスより大きいことを示す．さらに，記号的木変換器の合成で用いる手法を属性付き木変換器の合成アルゴリズムに取り入れることで，属性付き記号的木変換器の合成アルゴリズムを構成し，その正当性を証明する．証明では，属性付き記号的木変換器を，見かけ上等価な属性付き木変換器へ符号化することで，属性付き木変換器の合成アルゴリズムの正当性に帰着する．

目次

1　はじめに

関数プログラミングでは，基本的な計算を行う小さな関数を組み合わせて，大きな関数を定義する．つまり，ある関数 f を適用した式を，他の関数 g の実引数として渡す．このように連続する関数適用式を評価して値を得るときは通常，まず f が計算を実行して，その結果を返却する．次に g がそれを参照して，計算を実行し，その結果を返却する．このとき，f が返却する値を**中間データ**と呼び，f を**生産関数**，g を**消費関数**と呼ぶ．消費関数の計算結果に中間データが現れないときは，生産関数が中間データをメモリに確保したり，消費関数がそれを巡回しながら参照したり，参照した後にそのメモリ領域を解放したりするためのコストが無駄になる．

例えば，生産関数として，2 分木の葉節点にラベル付けされた整数値を，負数の場合は絶対値を取って自然数にしてから，リストに並べて返却する関数 *abs_leaves* を考える．また消費関数として，リストに並んだ自然数値を，偶数の場合は 2 で割ってから逆順に並べて返却する関数 *half_rev* を考える．2 分木の中間節点のラベルを bin であるとし，図 1.1のような 2 分木を $\mathrm{bin}(\mathrm{bin}(1,-2),\mathrm{bin}(\mathrm{bin}(-3,4),5))$ で表す．この 2 分木を ξ とする．*abs_leaves* は 2 引数関数であり，第 1 引数として 2 分木を受け取り，それを巡回しながら，数値を第 2 引数のリストに追加する．第 2 引数には初期値として空リスト ϵ を与える．*half_rev* もやはり 2 引数関数であり，第 1 引数としてリストを受け取ってそれを巡回しながら，数値を第 2 引数に追加する．$\mathit{half\_rev}(\mathit{abs\_leaves}(\xi,\epsilon),\epsilon)$ とすると，まず *abs_leaves* が計算を行って，その結果としてリスト $1(2(3(4(5(\epsilon)))))$ が得られる．次に，*half_rev* が計算を行って，$\mathit{half\_rev}(1(2(3(4(5(\epsilon))))),\epsilon)$ の結果としてリスト $5(2(3(1(1(\epsilon)))))$ が得られる．このとき，*abs_leaves* による計算結果のリスト $1(2(3(4(5(\epsilon)))))$ が中間データである．この中間的な計算結果を消費関数が巡回せず，初めの入力 $\mathrm{bin}(\mathrm{bin}(1,-2),\mathrm{bin}(\mathrm{bin}(-3,4),5))$ のみを巡回して最終結果のリスト $5(2(3(1(1(\epsilon)))))$ を直接計算することができれば，リスト $1(2(3(4(5(\epsilon)))))$ の生成と巡回を省略することができ，効率の良いプログラムとなる．この例では，2 分木を巡回しながら，葉の整数値が負かどうか判断し必要ならば絶対値を取った直後に，それが偶数かどうか判断し偶数ならば 2 で割る操作を行

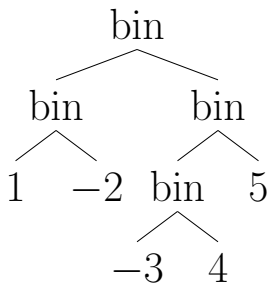


図 1.1: 2 分木 $\mathrm{bin}(\mathrm{bin}(1,-2),\mathrm{bin}(\mathrm{bin}(-3,4),5))$

い，2 分木の左側の要素がリストの末尾に来るように並べればよい．このように，2 つの関数定義から，中間データを介さないような 1 つの関数を定義し，その関数でもとの関数適用式を置き換えることを**関数融合**あるいは単に**融合**という．Darlington と Burstall [3] は，再帰プログラムに関数融合を機械的に施すための変換規則を考案した．これを基にした融合手法や，その他の様々な融合手法が与えれられている．

shortcut deforestaton [13] や warm fusion [17] は，圏の理論を応用し，特定の再帰パターンの高階関数に対して融合を施す手法である．stream fusion [4] は関数への入力が遅延リストやストリームである場合に融合できる手法である．supercompilation [23] や deforestation [26]，軽量関数融合 [19]，positive supercompilation [21] は，生産関数と消費関数の両方が多引数関数である場合にも融合することができる．しかしこれらの手法では，**蓄積引数**を使用する関数を融合できない場合がある．ここで蓄積引数とは，再帰プログラムによる計算の途中結果を蓄えておく引数である．例えば，*abs_leaves* と *half_rev* の第 2 引数はリストであり，そこに，巡回する 2 分木の葉節点の値やリストの要素を追加しているので，これは蓄積引数である．

木変換器の合成を応用した関数融合では，その他の手法では扱えない，蓄積引数を使用する関数に対して融合を施すことができる．木変換器とは，相互再帰関数の集まりであり，多くの再帰プログラムを形式化したモデルである．木変換器は入力された木構造を巡回し，注目する節点に対して相互再帰関数を適用し，その関数及び節点のラベルと一対一対応する**規則**を用いて出力の木構造を生成する．**降下型木変換器**（*top-down tree transducer*，TDTT）[20, 22] は最も基本的な木変換器であり，入力の木構造を，常に親節点から子節点へ下向きに巡回する．しかし，TDTT が持つ相互再帰関数の引数は 1 つに限定されるため，蓄積引数を伴う再帰プログラムを表現できない．**マクロ木変換器**（*macro tree transducer*，MTT） [6] は，蓄積引数を伴う再帰プログラムを直接的にモデル化した木変換器である．MTT が扱う相互再帰関数は多引数関数であり，蓄積引数に木構造を蓄積することができる．リストは木構造の一種なので，2 分木を巡回して葉を並べる計算や，リストを反転させる計算はいずれも MTT で表現できる．

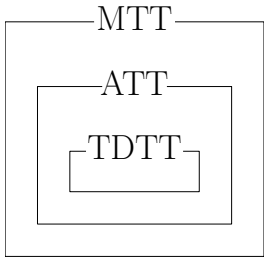
一方，いずれの計算も**属性付き木変換器**（*attributed tree transducer*，ATT） [7] でも表現することができる．ATT とは，TDTT が**継承属性**という特別な種類の関数を扱えるようにした木変換器である．継承属性は入力の木構造を，兄弟節点へ横向きに巡回したり，親節点へ上向きに巡回することができる．相互再帰関数の引数は 1 つであるが，ATT で継承属性が行う計算を，再帰プログラムで蓄積引数にデータを蓄積する処理に対応させることができるため，ATT は蓄積引数を伴う再帰プログラムの多くを表現することができる．

2 つの ATT を合成するアルゴリズムは，**記述的合成**（*descriptional composition*，DC）[11, 10, 15] として知られている．ATT で表現可能な計算は全て MTT でも表現できるが，逆に MTT で表現できる計算の中で，ATT では表現できない計算が存在する．計算のクラスを制限した MTT は，ATT と相互に変換することが可能であり，それによって MTT に DC を応用することができる．この考え方を発展させて，ATT を経由せずに MTT を直接合成するアルゴリズム [25] が考案されている．以上の木変換器の計算のクラスの関係を図 1.2(a)に示す．

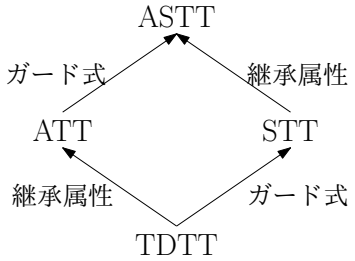
しかし，これらの木変換器では，入出力の木のラベルが有限の領域に属する必要がある．なぜなら，従来の木変換器では，それぞれの規則が，1 つのラベルに対してしか定義できず，無限個のラベルに対応する規則を無限個記述することはできないからである．*abs_leaves* と *half_rev* ではどちらも，節点のラベルが全ての整数値を取り得るから，蓄積引数を扱える ATT でも，これらの計算を表現することはできない．この問題に対し，無限の領域を扱えるように TDTT を拡張した**記号的木変換器**（STT） [24, 9] が提案されている．STT では，それぞれの規則が，ラベルの集合に対して定義できる．ラベルの集合は述語で表現され，無限集合でもよい．故に，STT では，無限の領域に属するラベルを持つ木に対して計算を行うことができる．ラベルの集合を表現する述語は，再帰プログラムのパターンマッチにおける**ガード式**に相当する．例えば，整数の正負を判断する述語を用いて負整数の集合と非負整数の集合を表すガード式を記述すれば，関数 *abs_leaves* を定義することができる．また，非負整数の偶奇を判断する述語を使い，偶数の集合と奇数の集合を表すガード式を記述することで，関数 *half_rev* を定義することができる．ただし，これらの再帰プログラムは蓄積引数を扱うため，TDTT の拡張である STT では表現できない．本研究で扱う木変換器の間の関係を図 1.2(b)に示す．

本研究では，ATT に対して STT と同様の拡張を施し，無限の領域を扱えるようにした**属性付き記号的木変換器**（ASTT）を提案する．ATT が TDTT より大きいクラスの関数を表現できることと同様の理由で，ASTT は STT より大きいクラスの関数を表現できる．また，STT の合成のアルゴリズムをもとに，ATT の合成アルゴリズムである DC を ASTT に対応させたアルゴリズムを考案する．そのために，DC が成功するための ATT に対する条件である single-use requirement を ASTT に対応させる．そして，ASTT の合成のアルゴリズムが正しいことを証明する．すなわち，合成後の 1 つの ASTT と，合成前の 2 つの ASTT の組み合わせの意味論が変わらないことを示す．この証明では，ASTT を見かけ上等価な ATT に「符号化」することで，ATT の合成の正当性に帰着する．

本論文の構成は次のとおりである．まず，2 章で，ASTT やその合成について定義するために必要となる基礎的な概念を導入する．また，STT や ASTT で扱う述語や関数に対する制約について述べる． 3 章と 4 章では，ASTT の考え方の基である 2 つの木変換器 ATT と STT を導入し，それらの合成アルゴリズムや主な性質について説明する．特に 3 章では DC や，DC が成功するための ATT への条件について説明し， 4 章では，ASTT の合成に応用する**記号的導出関**



(a) 主な木変換器のクラスの関係



(b) 本研究で扱う木変換器の関係

係について説明する．そして 5 章で ASTT を定義し，その具体例として先述の関数 *abs_leaves* と *half_rev* を ASTT で表現する．また，ASTT で表現できる計算のクラスが STT で表現できる計算のクラスより大きいことを示す． 6 章では ASTT に対する SUR と，ASTT の合成アルゴリズムを示し，その正当性を証明する． 7 章では，本研究に関連する先行研究について紹介する．最後に 8 章で，本研究の成果についてまとめ，関数融合への応用や，ATT より強力な木変換器への応用について考察する．

2 準備

本章では，次章以降のための基本的な概念を導入する．

2.1 文字集合，木，簡約系

空集合を $\emptyset$ で表す．全ての非負整数の集合を $\mathbb{N}$ で表し，全ての整数の集合を $\mathbb{Z}$ で表す．また $l \in \mathbb{N}$ に対して $\{1,\ldots,l\}$ を $[l]$ で表す．ただし $[0] = \emptyset$ である．有限集合 P の濃度を $|P|$ で表し，P の冪集合を $\wp(P)$ で表す．集合 P，Q の和集合を $P \uplus Q$ で表すとき，$P \cap Q = \emptyset$ であるとする．要素 p と q の対を (p,q) で表し，集合 P，Q の直積 $\{(p,q) \mid p \in P, q \in Q\}$ を $P \times Q$ で表す．直積は結合的であるとし，$(P \times Q) \times R = P \times (Q \times R)$ を単に $P \times Q \times R$ と表す．$p_1, p_2, \ldots, p_n \in P$ のとき，長さ n の列を $p_1 p_2 \cdots p_n$ で表し，空列を ε で表す．P 上の全ての有限長の列の集合を P^* で表す．$k \in \mathbb{N}$ として $P = \{n_0,\ldots,n_k\} \subseteq \mathbb{N}$ のとき，$\max(P) = \max\{n_0,\ldots,n_k\}$ で P の最大値を表す．

集合 P から集合 Q への関数 f を $f : P \to Q$ で表す．関数 f による集合 P の像を $f(P) = \{q \in Q \mid q = f(p), p \in P\}$ で定義する．関数 f と g の合成を $f \circ g(x) = g(f(x))$ で定義する．また，関数の集合 F と G に対して，その合成を $F \circ G = \{f \circ g \mid f \in F, g \in G\}$ で定義する．

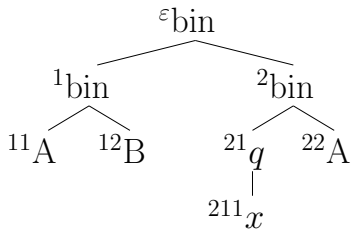
集合 $\Sigma \neq \emptyset$ と自然数の有限集合 $P \subseteq \mathbb{N}$ に対して，全射 $rk : \Sigma \to P$ が定義されているとき，Σ を**ランク付きランク付き文字集合**と呼び，任意の $\sigma \in \Sigma$ を**文字**，$rk(\sigma)$ を**ランク**と呼ぶ．σ のランクが l である任意の $\sigma \in \Sigma$ の集合を $\Sigma^{(l)}$ で表し，$\sigma \in \Sigma^{(l)}$ であるとき，σ を $\sigma^{(l)}$ と表すこともある．$\mathrm{maxrk}(\Sigma) = \max(P)$ と定義する．$X \cap \Sigma = \emptyset$ を満たす変数の集合 X による Σ 上の木の集合 $T_\Sigma(X)$ を，次を満たす最小の集合 T で定義する:

- $X \subseteq T$.
- 任意の $l \in \mathbb{N}, \sigma \in \Sigma^{(l)}, \xi_1, \ldots, \xi_l \in T$ に対して $\sigma(\xi_1,\ldots,\xi_l) \in T$.

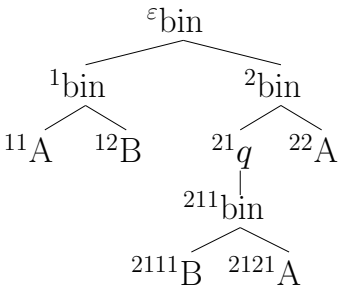
木 $\xi \in T_\Sigma(X)$ の全ての**経路**の集合を $path(\xi) \subseteq (\mathbb{N} \setminus \{0\})^*$ で表し，次で定義する:

- 任意の $x \in X$ に対して $path(x) = \{\varepsilon\}$.
- 任意の $l \in \mathbb{N}$，$\sigma \in \Sigma$，$\xi_1,\ldots,\xi_l \in T_\Sigma(X)$ に対して $path(\sigma(\xi_1,\ldots,\xi_l)) = \{\varepsilon\} \cup \{iw \mid i \in [l], w \in path(\xi_i)\}$.

以下，π は木変換器の構文中で経路を束縛する変数として用いる特別な文字とする．例えば $\pi 1$ は，π に束縛した経路に 1 を付け加えた経路を表す．また，木 $\xi \in T_\Sigma(X)$ に現れる全ての π を $w \in path(\xi)$ で置き換えて得られる木を $\xi[\pi := w] \in T_\Sigma(X)$ で表す．例えば，$a(\pi 1)$ が全体で 1



(a) 置換前の木 ξ



(b) 置換後の木 ζ

図 2.1: 木の上の変数の置換

つの変数であるとき，$\mathrm{bin}(1, a(\pi 1))$ に対して，$\mathrm{bin}(1, a(\pi 1))[\pi := 12] = \mathrm{bin}(1, a(121))$ である．

任意の木 $\xi \in T_\Sigma(X)$ と経路 $w \in path(\xi)$ に対して，まず，w **における** ξ **の部分木**を $sub_\xi(w) \in T_\Sigma(X)$ で表し，次で定義する:

- $sub_\xi(\varepsilon) = \xi$.
- 任意の $l \in \mathbb{N}, \sigma^{(l)} \in \Sigma, \xi_1, \ldots, \xi_l \in T_\Sigma(X), i \in [l]$ に対して $sub_{\sigma(\xi_1,\ldots,\xi_l)}(i\pi) = sub_{\xi_i}(\pi)$.

次に，$\xi \in T_\Sigma(X)$ のとき，w **における** ξ **の文字**を $lab_\xi(w) \in (\Sigma \cup X)$ で表し，次で定義する:

- $sub_\xi(w) = x \in X$ のとき，$lab_\xi(w) = x$.
- $l \in \mathbb{N}$，$\sigma \in \Sigma$，$\xi_1, \ldots, \xi_l \in T_\Sigma(X)$ として，$sub_\xi(w) = \sigma(\xi_1, \ldots, \xi_l)$ のとき，
  - $w = \varepsilon$ のとき，$lab_\xi(w) = \sigma$.
  - $w = iw'$ かつ $i \in [l]$ のとき，$lab_\xi(w) = lab_{\xi_i}(w')$.

ランク付き文字集合 Σ，Δ と変数の集合 X，Y に対し，木 $\xi \in T_\Sigma(X)$ の全ての $x \in X$ を $\eta \in T_\Delta(Y)$ で置き換えて得られる木を $\xi[x := \eta] \in T_{\Sigma\cup\Delta}(X \cup Y)$ で表す．すなわち次を満たす ζ で定義する:

- $path(\xi) = path(\zeta)$.
- 任意の $w \in path(\xi)$ に対して，
  - $lab_\xi(w) = x$ ならば $lab_\zeta(w) = \eta$.
  - $lab_\xi(w) \neq x$ ならば $lab_\zeta(w) = lab_\xi(w)$.

例えば，$q(x)$ を変数として，木 $\xi = \mathrm{bin}(\mathrm{bin}(\mathrm{A}, \mathrm{B}), \mathrm{bin}(q(x), \mathrm{A}))$ は図 2.1(a)で示される．各節点の左上に経路を記している．図 2.1(b)には木 $\zeta = \xi[x := \mathrm{bin}(\mathrm{B}, \mathrm{A})] = \mathrm{bin}(\mathrm{bin}(\mathrm{A}, \mathrm{B}), \mathrm{bin}(q(\mathrm{bin}(\mathrm{B}, \mathrm{A})), \mathrm{A}))$ を示す．このとき，$lab_\xi(211) = x$ かつ $lab_\zeta(211) = \mathrm{bin}$ である．また，$\xi[x_1 := \eta_1][x_2 := \eta_2] \cdots [x_n := \eta_n]$ を $\xi[x_1, x_2, \ldots, x_n := \eta_1, \eta_2, \ldots, \eta_n]$ で表す．これは $\xi[x_i := \eta_i]_{i\in[n]}$ や $\xi[x_i := \eta_i]_{x_i \in \{x_1,\ldots,x_n\}}$ で略記する．

ランク付き文字集合 Σ と 変数の集合 X に対して，木 $\eta \in T_\Sigma(X)$ の**高さ**を $ht(\eta)$ で表し，次

を満たす関数 $ht : T_\Sigma(X) \to \mathbb{N}$ で定義する:

- $\eta = x \in X$ のとき，$ht(\eta) = ht(x) = 1$.
- $\eta = \sigma(\eta_1, \ldots, \eta_l)$ のとき，$ht(\eta) = ht(\sigma(\eta_1, \ldots, \eta_l)) = 1 + \max\{ht(\eta_1), \ldots, ht(\eta_l)\}$.

集合 A と A 上の二項関係 $\Rightarrow$ に対して対 $(A, \Rightarrow)$ を**簡約系**と呼ぶ．任意の $i \in [n]$，$a_i \in A$，$a_1, \ldots, a_{n+1} \in A$ に対して $a_i \Rightarrow a_{i+1}$ が成立することを $a_1 \Rightarrow^n a_{n+1}$ で表す．なお，$a \Rightarrow^0 a$ である．さらに，$a, b \in A$ に対して $a \Rightarrow^n b$ を満たす $n \in \mathbb{N}$ が存在することを $a \Rightarrow^* b$ で表す．任意の $a \in A$ に対して $a \Rightarrow b$ を満たす $b \in A$ が存在するとき，a は**可約**であるといい，そうでないとき a は**既約**であるという．$a \Rightarrow^* b$ を満たす既約な $b \in A$ を a の**正規形**と呼ぶ．ある a に対して一意な正規形が存在するとき，それを $nf(\Rightarrow, a)$ で表す．任意の $a \in A$ に対して $nf(\Rightarrow, a)$ が存在するとき，この簡約系 $(A, \Rightarrow)$ は**決定的**であるといい，そうでないとき**非決定的**であるという．

2.2 述語と関数，ラベル構造

記号的木変換器では，ランク付き文字集合上の述語や関数を扱う．本節ではそれらを導入し，それらに対する制約を設ける．

Σ を可算のランク付きランク付き文字集合とし，Σ **上の単項述語**を写像 $\varphi : \Sigma \to \{0, 1\}$ で定義する．特に，任意の $l \in rk(\Sigma)$ に対して，$\Sigma^{(l)}$ **上の単項述語**を写像 $\varphi^{(l)} : \Sigma^{(l)} \to \{0, 1\}$ で定義する．以下，単項述語を単に述語と呼ぶ．$\mathrm{Pred}(\Sigma)$ で Σ 上の全ての述語の集合を表し，述語 $\varphi \in \mathrm{Pred}(\Sigma)$ に対して $[\![\varphi]\!] \stackrel{\mathrm{def}}{=} \{\sigma \in \Sigma \mid \varphi(\sigma) = 1\}$ と定義する．

任意の $\varphi, \psi \in \mathrm{Pred}(\Sigma)$ に対して，$\mathrm{Pred}(\Sigma)$ 上のブール演算 $\neg$，$\wedge$，$\vee$ をそれぞれ次で定義する:

$$[\![\neg\varphi]\!] = \Sigma \setminus [\![\varphi]\!], [\![\varphi \wedge \psi]\!] = [\![\varphi]\!] \cap [\![\psi]\!], [\![\varphi \vee \psi]\!] = [\![\varphi]\!] \cup [\![\psi]\!].$$

$\Phi \subseteq \mathrm{Pred}(\Sigma)$ を有限集合として，**ブール閉包**を $\mathrm{BC}(\Phi)$ で表し，次を満たす最小の集合 $B \subseteq \mathrm{Pred}(\Sigma)$ で定義する:

1. $\Phi \cup \{\top, \bot\} \subseteq B$.
2. 任意の $\varphi, \psi \in B$ に対して $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi \in B$.

ここで $\top$，$\bot$ は $[\![\top]\!] = \Sigma$ かつ $[\![\bot]\!] = \emptyset$ を満たす述語である．$\Phi \subseteq \mathrm{Pred}(\Sigma^{(l)})$ のときは単項述語と同様に $\varphi^{(l)}$ と表す．文脈上明らかなときは (l) を省略する．任意の述語 $\varphi \in \Phi$ に対し，$[\![\varphi \wedge \neg\varphi]\!] = \emptyset$ かつ $[\![\varphi \vee \neg\varphi]\!] = \Sigma$ であり，$\wedge$，$\vee$ は分配的だから，$\mathrm{BC}(\Phi)$ はブール代数である．

Σ 上の述語の集合 $\mathrm{Pred}_0(\Sigma)$ を，次を満たす最大の集合 $\Phi \subseteq \mathrm{Pred}(\Sigma)$ として定義する:

1. 各述語 $\varphi \in \Phi$ は原始再帰的，すなわち任意のラベル $\sigma \in \Sigma$ に対して $\varphi(\sigma)$ を計算するアルゴリズムが明示的に与えられている．

2. $\mathrm{BC}(\Phi)$ の任意の述語の空判定が決定可能，すなわち各述語 $\varphi \in \mathrm{BC}(\Phi)$ に対して $[\![\varphi]\!] = \emptyset$ の真偽が定まる．

$\Phi \subseteq \mathrm{Pred}_0(\Sigma)$ のとき，対 (Σ, Φ) を**ラベル構造**と呼ぶ．特に，$\top, \bot \in \mathrm{Pred}_0(\Sigma)$ であり，$(\Sigma, \{\top\})$ と $(\Sigma, \{\bot\})$ はそれぞれラベル構造である．

Σ，Δ をランク付き文字集合とする．任意の k，$l \in \mathbb{N}$ に対して，全ての単項の原始再帰関数 $f : \Sigma^{(k)} \to \Delta^{(l)}$ の集合を $F(\Sigma^{(k)} \to \Delta^{(l)})$ で表す．有限集合 $P \subseteq \mathbb{N}$ への全射 $rk : F \to P$ が定義されるとき，$F(\Sigma^{(k)} \to \Delta^{(l)})$ はランク付き文字集合である．これ以降は，任意の $f : \Sigma^{(k)} \to \Delta^{(l)} \in F(\Sigma^{(k)} \to \Delta^{(l)})$ に対して $rk(f) = l$ であると仮定する．$F(\Sigma \to \Delta) = \bigcup_{(k,l) \in rk(\Sigma) \times rk(\Delta)} F(\Sigma^{(k)} \to \Delta^{(l)})$ と定義する．$f \in F(\Sigma \to \Delta^{(l)})$ であることを $f^{(l)}$ で表す．また，文字の定数 $c \in \Delta$ に対し，任意の文字 $\sigma \in \Sigma$ に対して $f(\sigma) = c$ で定義される定数関数 f を $\hat{c}$ で表す．集合 X による $F(\Sigma \to \Delta)$ 上の木 $\xi \in T_{F(\Sigma \to \Delta)}(X)$ に現れる全ての関数 $f \in F(\Sigma \to \Delta)$ を，f に文字 $\sigma \in \Sigma$ を与えた結果の値 $f(\sigma)$ で置き換えた木を $give_\xi(\sigma) \in T_\Delta(X)$ で表す．すなわち次を満たす η で定義する:

- $path(\xi) = path(\eta)$.
- 任意の $w \in path(\xi)$ に対して
  - $lab_\xi(w) = f \in F(\Sigma \to \Delta)$ ならば $lab_\eta(w) = f(\sigma)$.
  - $lab_\xi(w) \in X$ ならば $lab_\eta(w) = lab_\xi(w)$.

また，集合 Y による $F(\Delta \to \Omega)$ 上の木 $\zeta \in T_{F(\Delta \to \Omega)}(Y)$ に現れる全ての関数 $g_i \in F(\Delta \to \Omega)$ を，f との合成関数 $f \circ g_i \in F(\Sigma \to \Omega)$ で置き換えた木を $comp_\zeta(f) \in T_{F(\Sigma \to \Omega)}(Y)$ で表す．すなわち次を満たす η で定義する:

- $path(\zeta) = path(\eta)$.
- 任意の $w \in path(\zeta)$ に対して
  - $lab_\zeta(w) = g \in F(\Sigma \to \Delta)$ ならば $lab_\eta(w) = f \circ g$.
  - $lab_\zeta(w) \in X$ ならば $lab_\eta(w) = lab_\zeta(w)$.

記号的木変換器や属性付き記号的木変換器の合成では，ランク付き文字集合上の述語や関数を合成し，新しい述語を構成する．本論文ではランク付き文字集合 Δ 上の述語が $\mathrm{Pred}_0(\Delta)$ に属すとき，それとランク付き文字集合 Σ 上の関数の合成によって得られる述語もまた $\mathrm{Pred}_0(\Sigma)$ に属すことを仮定する．つまり，これ以降の任意のランク付き文字集合 Σ と Δ に対して，

$$F(\Sigma \to \Delta) \circ \mathrm{Pred}_0(\Delta) \subseteq \mathrm{Pred}_0(\Sigma)$$

を仮定する．

3　属性付き木変換器

本章では，有限のランク付き文字集合を扱う**属性付き木変換器**（*attributed tree transducer*, ATT）を導入する．ATT は相互再帰関数の集まりであり，関数は 1 引数関数であり，第 1 引数に経路を受け取って，その経路における節点のラベルを読み取って木を出力する．そして再帰呼び出しを行って木を巡回する．ATT の特徴として，関数は木を上下左右に巡回できる．そのため，ある節点において，その子節点に対して適用した関数がその部分木を巡回して，適用した節点に戻ってくることができる．さらに，その関数が計算した結果を，兄弟節点の部分木に対する計算に受け渡すことができる．例えば，木 $\mathrm{bin}(\mathrm{bin}(\mathrm{A},\mathrm{B}),\mathrm{bin}(\mathrm{bin}(\mathrm{B},\mathrm{A}),\mathrm{A}))$ を下方向に巡回する関数 a と，上方向に巡回する関数 b が図 3.1の矢印の向きに巡回して，葉節点に到達するたびにそのラベルを出力していけば，葉節点のラベルを左から順に並べることができる．再帰プログラムや MTT では，この計算は蓄積引数を用いて表現される．

3.1　属性付き木変換器の定義

本節では，ATT を形式的に定義するが，その前にまず，ATT の関数の定義の形を定める集合を導入する．

定義 3.1. Σ，Δ をランク付き文字集合，Syn，Inh を集合，$l \in \mathbb{N}$ とする．このとき，

$$
\begin{aligned}
OCC(Syn, Inh, l) &\stackrel{\text{def}}{=} \{(a, \pi j) \mid a \in Syn, j \in [l]\} \cup \{(b, \pi) \mid b \in Inh\} \\
LHS(Syn, Inh, l) &\stackrel{\text{def}}{=} \{(a, \pi) \mid a \in Syn\} \cup \{(b, \pi i) \mid b \in Inh, i \in [l]\} \\
RHS(Syn, Inh, \Delta, l) &\stackrel{\text{def}}{=} T_\Delta(OCC(Syn, Inh, l))
\end{aligned}
$$

と定義する．

木変換器では，関数は木の上の変数であり，関数を置換する手続きを，規則という単位で定義

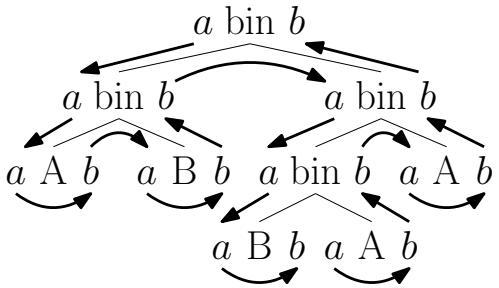


図 3.1: 木 $\mathrm{bin}(\mathrm{bin}(\mathrm{A},\mathrm{B}),\mathrm{bin}(\mathrm{bin}(\mathrm{B},\mathrm{A}),\mathrm{A}))$ を巡回する ATT の関数 a と b

する．ATT の規則では，関数名と木の節点の相対的な経路を記述することで，ある時点で到達した節点から，移動する先の節点を指定する．LHS は，定義する関数の名前と，他の節点に移動するための基点となる経路の指定の仕方を定める．OCC は，他の節点に移動することを指定する仕方を定める．RHS は，出力する木の形を定める．

次に，ATT の構文を定義する．

定義 3.2 (属性付き木変換器)**．属性付き木変換器**（ATT）とは，以下を満たす 7 つ組 $M = (Syn, Inh, \Sigma, \Delta, in, \sharp, R)$ である:

- Syn と Inh は $Syn \cap Inh = \emptyset$ を満たす有限集合であり，その要素をそれぞれ**合成属性**，**継承属性**と呼ぶ．また，$a \in Syn \cup Inh, i \in \mathbb{N}$ のとき (a, π) や $(a, \pi i)$ をそれぞれ $a(\pi)$ や $a(\pi i)$ で表す．
- Σ と Δ は $(Syn \uplus Inh) \cap (\Sigma \cup \Delta) = \emptyset$ を満たす有限のランク付き文字集合であり，それぞれ**入力ランク付き文字集合**，**出力ランク付き文字集合**と呼ぶ．
- $in \in Syn$ を**初期属性**と呼ぶ．
- $\sharp^{(1)} \notin \Sigma \cup Syn \uplus Inh$ は文字であり，**初期記号**と呼ぶ．また Σ^+ で $\Sigma \uplus \{\sharp\}$ を表す．
- $R \subseteq LHS(Syn, Inh, \mathrm{maxrk}(\Sigma)) \times \Sigma^+ \times RHS(Syn, Inh, \Delta, \mathrm{maxrk}(\Sigma))$ は以下を満たす: $RHS(Syn, Inh, \Delta, l)$ を $RHS^{(l)}$ と表すとき，
  - 任意の $\sigma^{(l)} \in \Sigma, a \in Syn$ に対して $(a(\pi), \sigma, \eta) \in R$ を満たす $\eta \in RHS^{(l)}$ が唯 1 つ存在する．
  - 任意の $\sigma^{(l)} \in \Sigma, b \in Inh, i \in [l]$ に対して $(b(\pi i), \sigma, \eta) \in R$ を満たす $\eta \in RHS^{(l)}$ が唯 1 つ存在する．
  - 任意の $b \in Inh, i \in [l]$ に対して $(b(\pi i), \sharp, \eta) \in R$ を満たす $\eta \in RHS^{(1)}$ が唯 1 つ存在する．
  - $(in(\pi), \sharp, \eta) \in R$ を満たす $\eta \in RHS^{(1)}$ が唯 1 つ存在する．

  なお，$\rho = (a(w), \sigma^{(l)}, \eta) \in R$ を**属性規則**と呼び，次の形で表す: $a(w) \overset{\sigma^{(l)}}{\to} \eta$．また，$R^{(\sigma)}$ を $\{(a(w), \sigma', \eta) \in R \mid \sigma' = \sigma\}$ で定義する．

例 3.3. ATT $M_{leaves} = (\{a_1\}, \{b_1\}, \{\mathrm{bin}^{(2)}, \mathrm{A}^{(0)}, \mathrm{B}^{(0)}\}, \{\mathrm{A}^{(1)}, \mathrm{B}^{(1)}, \epsilon^{(0)}\}, a_1, \sharp_1, R_1)$ と $M_{rev} = (\{a_2\}, \{b_2\}, \{\mathrm{A}^{(1)}, \mathrm{B}^{(1)}, \epsilon^{(0)}\}, \{\mathrm{A}^{(1)}, \mathrm{B}^{(1)}, \epsilon^{(0)}\}, a_2, \sharp_2, R_2)$ を図 3.2の R_1 と R_2 で定義する．M_{leaves} は 2 分木を巡回して，葉節点のラベルを左から順にリストに並べる．M_{rev} はリストを逆順に並べ替える．2 分木の葉節点のラベルやリストの要素は A と B の 2 種類である．

次に，ATT の意味論を定義する．

定義 3.4 (ATT による導出関係)**．** $M = (Syn, Inh, \Sigma, \Delta, in, \sharp, R)$ を ATT として，$\xi \in T_{\Sigma^+}$ とする．このとき，ξ における M による**導出関係**を，$T_\Sigma(\{a(w) \mid a \in Syn \cup Inh, w \in path(\xi)\})$ 上の最小の二項関係 $\overset{M,\xi}{\Rightarrow}$ で定義する:

$$
\begin{array}{ll}
R_1 = \{\ a_1(\pi) \overset{\sharp_1}{\to} a_1(\pi 1) & R_2 = \{\ a_2(\pi) \overset{\sharp_2}{\to} a_2(\pi 1) \\
\quad a_1(\pi) \overset{\mathrm{bin}}{\to} a_1(\pi 1) & \quad a_2(\pi) \overset{\mathrm{A}}{\to} a_2(\pi 1) \\
\quad a_1(\pi) \overset{\mathrm{A}}{\to} \mathrm{A}(b_1(\pi)) & \quad a_2(\pi) \overset{\mathrm{B}}{\to} a_2(\pi 1) \\
\quad a_1(\pi) \overset{\mathrm{B}}{\to} \mathrm{B}(b_1(\pi)) & \quad a_2(\pi) \overset{\epsilon}{\to} b_2(\pi) \\
\quad b_1(\pi 1) \overset{\sharp_1}{\to} \epsilon & \quad b_2(\pi 1) \overset{\sharp_2}{\to} \epsilon \\
\quad b_1(\pi 1) \overset{\mathrm{bin}}{\to} a_1(\pi 2) & \quad b_2(\pi 1) \overset{\mathrm{A}}{\to} \mathrm{A}(b_2(\pi)) \\
\quad b_1(\pi 2) \overset{\mathrm{bin}}{\to} b_1(\pi) \quad \} & \quad b_2(\pi 1) \overset{\mathrm{B}}{\to} \mathrm{B}(b_2(\pi)) \quad \}
\end{array}
$$

図 3.2: M_{leaves} と M_{rev} の規則

- $a(w) \overset{M,\xi}{\Rightarrow} \eta[\pi := w] \iff a \in Syn, (a(\pi) \overset{\sigma^{(l)}}{\to} \eta) \in R, lab_\xi(w) = \sigma.$
- $b(wi) \overset{M,\xi}{\Rightarrow} \eta[\pi := w] \iff b \in Inh, (b(\pi i) \overset{\sigma^{(l)}}{\to} \eta) \in R, lab_\xi(w) = \sigma.$
- $\delta(\eta_1, \ldots, \eta_i, \ldots, \eta_l) \overset{M,\xi}{\Rightarrow} \delta(\eta_1, \ldots, \eta_i', \ldots, \eta_l)$
  $\iff \delta \in \Delta^{(l)}, \eta_i \overset{M,\xi}{\Rightarrow} \eta_i', \forall j \in [i-1].\, \eta_j \in T_\Delta.$

3 つ目の項目より，η_i の左側の全ての木は変数を持たない簡約済みの木であるから，この導出関係による簡約は最外最左簡約である．従って，この簡約系は決定的である．従って，正規形が存在するとき，それは一意である．ただし，導出が循環するときは，正規形が存在しない．そこで，以降の ATT には次の定義可能性を仮定する．

定義 3.5 (ATT の意味論)．$M = (Syn, Inh, \Sigma, \Delta, in, \sharp, R)$ を ATT とする．M による**木変換**とは，次で定義する関数 $\mathbf{T}_M : T_\Sigma \to T_\Delta$ である: 任意の $\xi \in T_\Sigma$ に対して，$\mathbf{T}_M(\xi) = nf(\overset{M,\sharp(\xi)}{\Rightarrow}, in(\varepsilon))$.

これが定義可能（well-defined）であるためには，次が成立すれば十分である: 任意の $\xi \in T_\Sigma, a \in Syn \cup Inh, w \in path(\sharp(\xi))$ に対して，$\zeta = nf(\overset{M,\sharp(\xi)}{\Rightarrow}, a(w)), (a, w) \notin Inh \times \{\varepsilon\}$ を満たす $\zeta \in T_\Delta$ が存在する．

これが成立するとき，M は**定義可能**（*well-defined*）であるという．

以降の ATT は，指定のない限り定義可能とする．全ての定義可能な ATT による木変換の集合を ATT で表す．

例 3.6. 木 $\xi = \mathrm{bin}(\mathrm{bin}(\mathrm{A}, \mathrm{B}), \mathrm{bin}(\mathrm{bin}(\mathrm{B}, \mathrm{A}), \mathrm{A})) \in T_{\{\mathrm{bin},\mathrm{A},\mathrm{B}\}}$ に対して $\mathbf{T}_{M_{leaves}}(\xi) = nf(\overset{M_{leaves},\sharp_1(\xi)}{\Rightarrow}, a_1(\varepsilon)) = \mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(\mathrm{A}(\epsilon))))) \in T_{\{\mathrm{A},\mathrm{B},\epsilon\}}$ であり，これは図 3.4(a)の簡約で計算される．$\sharp_1(\xi)$ を図 3.3に示す．各節点の左上にその節点の経路を記している．出力の木を ζ とすると，$\mathbf{T}_{M_{rev}}(\zeta) = nf(\overset{M_{rev},\sharp_2(\zeta)}{\Rightarrow}, a_2(\varepsilon)) = \mathrm{A}(\mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(\epsilon)))))$ であり，これは図 3.3(b) の簡約で

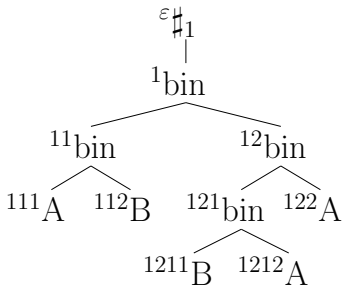


図 3.3: 木 $\sharp_1(\mathrm{bin}(\mathrm{bin}(\mathrm{A},\mathrm{B}),\mathrm{bin}(\mathrm{bin}(\mathrm{B},\mathrm{A}),\mathrm{A})))$

計算される．従って $\mathbf{T}_{M_{leaves}} \circ \mathbf{T}_{M_{rev}}(\mathrm{bin}(\mathrm{bin}(\mathrm{A},\mathrm{B}),\mathrm{bin}(\mathrm{bin}(\mathrm{B},\mathrm{A}),\mathrm{A}))) = \mathrm{A}(\mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(\epsilon)))))$ である． M_{leaves} は 2 分木の葉節点を左から順に並べたリストを出力する．M_{rev} はリストを反転する．

3.2 属性付き木変換器の合成

ATT の合成アルゴリズムは**記述的合成**（*descriptional composition*, DC）と呼ばれる．DC について述べる前に，DC が成功するための ATT に対する条件について述べる．

定義 3.7 (単一使用制約)**.** ATT $M = (Syn, Inh, \Sigma, \Delta, in, \sharp, R)$ に対して，次が成立するとき，M は単一使用制約（*single-use requirement*, SUR）を満たすという: 任意の $\sigma \in \Sigma^+, a(w) \in OCC(Syn, Inh, \mathrm{maxrk}(\Sigma))$ に対して，

$$\forall w_1 \in path(\eta_1), w_2 \in path(\eta_2).\, (x_1, w_1) \neq (x_2, w_2), sub_{\eta_1}(w_1) = sub_{\eta_2}(w_2) = a(w).$$

を満たす $(x_1 \xrightarrow{\sigma} \eta_1), (x_2 \xrightarrow{\sigma} \eta_2) \in R^{(\sigma)}$ が存在しない．

なお，SUR を満たす全ての定義可能な ATT による木変換の集合を ATT_{su} で表す．

SUR を満たす ATT では，ある 1 つの文字に対応する全ての規則の右辺を集めたとき，その右辺全体で，同じ関数の再帰呼び出しが丁度 1 回である．

DC によって 2 つの ATT M_1, M_2 を構文的に合成し，1 つの ATT を得るとき，その ATT を $M_1 \fatsemi M_2$ で表す．

定義 3.8 (記述的合成)**.** 任意の 2 つの ATT M_1, M_2 に対して，

$$M_1 = (Syn_1, Inh_1, \Sigma_1, \Delta_1, in_1, \sharp_1, R_1)$$
$$M_2 = (Syn_2, Inh_2, \Sigma_2, \Delta_2, in_2, \sharp_2, R_2)$$

かつ，$\Delta_1 \subseteq \Sigma_2$ であり，$Inh_1 = \emptyset$ または M_2 が SUR を満たすとき，まず，ATT M_2' を次で定義する:

$$M_2' = (Syn_2, Inh_2, \Sigma_2', \Delta_2', in_2, \sharp_2, R_2')$$

$$
\begin{aligned}
a_1(\varepsilon) &\Rightarrow a_1(1)\\
&\Rightarrow a_1(11)\\
&\Rightarrow a_1(111)\\
&\Rightarrow \mathrm{A}(b_1(111))\\
&\Rightarrow \mathrm{A}(a_1(112))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(b_1(112)))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(b_1(11)))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(a_1(12)))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(a_1(121)))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(a_1(1211)))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(\mathrm{B}(b_1(1211))))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(\mathrm{B}(a_1(1212))))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(b_1(1212)))))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(b_1(121)))))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(a_1(122)))))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(\mathrm{A}(b_1(122))))))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(\mathrm{A}(b_1(12))))))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(\mathrm{A}(b_1(1))))))\\
&\Rightarrow \mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(\mathrm{A}(\epsilon))))))
\end{aligned}
$$

(a) M_{leaves} による簡約例

$$
\begin{aligned}
a_2(\varepsilon) &\Rightarrow a_2(1)\\
&\Rightarrow a_2(11)\\
&\Rightarrow a_2(111)\\
&\Rightarrow a_2(1111)\\
&\Rightarrow a_2(11111)\\
&\Rightarrow a_2(111111)\\
&\Rightarrow b_2(111111)\\
&\Rightarrow \mathrm{A}(b_2(11111))\\
&\Rightarrow \mathrm{A}(\mathrm{A}(b_2(1111)))\\
&\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(b_2(111))))\\
&\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\mathrm{B}(b_2(11)))))\\
&\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(b_2(1)))))))\\
&\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(\epsilon))))))
\end{aligned}
$$

(b) M_{rev} による簡約例

図 3.4: 2 つの ATT による簡約例

ただし，

$$
\begin{aligned}
\Sigma_2' &= \Sigma_2 \cup \{(a_1(w))^{(0)} \mid a_1(w) \in OCC(Syn_1, Inh_1, \mathrm{maxrk}(\Delta_1^+))\}\\
\Delta_2' &= \Delta_2 \cup \{(\langle a_1, a_2\rangle(w))^{(0)} \mid \langle a_1, a_2\rangle(w) \in OCC(Syn_1 \times Syn_2, Syn_1 \times Inh_2, \mathrm{maxrk}(\Delta_1^+))\}\\
R_2' &= R_2 \cup \bigcup_{a_1(w) \in OCC(Syn_1, Inh_1, \mathrm{maxrk}(\Delta_1^+))} \{a_2(\pi) \overset{a_1(w)}{\to} (\langle a_1, a_2\rangle(w))^{(0)} \mid a_2 \in Syn_2\}.
\end{aligned}
$$

ここで w に含まれる π は変数ではなく文字の一部であるとする．

次に，$M_1 \fatsemi M_2$ を，次を満たす $(Syn, Inh, \Sigma_1, \Delta_2, \langle in_1, in_2\rangle, \sharp_1, R)$ で定義する:

$$
\begin{aligned}
Syn &= (Syn_1 \times Syn_2) \cup (Inh_1 \times Inh_2)\\
Inh &= (Syn_1 \times Inh_2) \cup (Inh_1 \times Syn_2)
\end{aligned}
$$

ただし，R は次を満たす:

- 任意の $\sigma^{(l)} \in \Sigma$ に対して，$R^{(\sigma)}$ は次の属性規則を含む.
  - 任意の $a_1 \in Syn_1, (a_1(\pi) \xrightarrow{\sigma} \eta) \in R_1$ に対して:
    - 任意の $a_2 \in Syn_2$ に対して，

$$\langle a_1, a_2 \rangle(\pi) \xrightarrow{\sigma} nf(\overset{M_2', \eta}{\Rightarrow}, a_2(\varepsilon))[b_2'(\varepsilon) := \langle a_1, b_2' \rangle(\pi)]_{b_2' \in Inh_2}.$$

    - 任意の $b_2 \in Inh_2, w \in path(\eta), j' \in [l], a_1'(\pi j') = sub_\eta(w), a_1' \in Syn_1$ に対して，

$$\langle a_1', b_2 \rangle(\pi j') \xrightarrow{\sigma} nf(\overset{M_2', \eta}{\Rightarrow}, b_2(\varepsilon))[b_2'(\varepsilon) := \langle a_1, b_2' \rangle(\pi)]_{b_2' \in Inh_2}.$$

    - 任意の $b_2 \in Inh_2, w \in path(\eta), b_1'(\pi) = sub_\eta(w), b_1' \in Inh_1$ に対して，

$$\langle b_1', b_2 \rangle(\pi) \xrightarrow{\sigma} nf(\overset{M_2', \eta}{\Rightarrow}, b_2(\varepsilon))[b_2'(\varepsilon) := \langle a_1, b_2' \rangle(\pi)]_{b_2' \in Inh_2}.$$

  - 任意の $b_1 \in Inh_1, j \in [l], (b_1(\pi j) \xrightarrow{\sigma} \eta) \in R_1$ に対して:
    - 任意の $a_2 \in Syn_2$ に対して，

$$\langle b_1, a_2 \rangle(\pi j) \xrightarrow{\sigma} nf(\overset{M_2', \eta}{\Rightarrow}, a_2(\varepsilon))[b_2'(\varepsilon) := \langle b_1, b_2' \rangle(\pi j)]_{b_2' \in Inh_2}.$$

    - 任意の $b_2 \in Inh_2, w \in path(\eta), j' \in [l], a_1'(\pi j') = sub_\eta(w), a_1' \in Syn_1$ に対して，

$$\langle a_1', b_2 \rangle(\pi j') \xrightarrow{\sigma} nf(\overset{M_2', \eta}{\Rightarrow}, b_2(\varepsilon))[b_2'(\varepsilon) := \langle b_1, b_2' \rangle(\pi j)]_{b_2' \in Inh_2}.$$

    - 任意の $b_2 \in Inh_2, w \in path(\eta), b_1'(\pi) = sub_\eta(w), b_1' \in Inh_1$ に対して，

$$\langle b_1', b_2 \rangle(\pi) \xrightarrow{\sigma} nf(\Rightarrow_S M_2'\eta, b_2(\varepsilon))[b_2'(\varepsilon) := \langle b_1, b_2' \rangle(\pi j)]_{b_2' \in Inh_2}.$$

- さらに $R^{(\sharp_1)}$ は次の規則を含む
  - $in_1(\pi) \xrightarrow{\sharp_1} \eta \in R_1$ に対して:
    - $\langle in_1, in_2 \rangle(\pi) \xrightarrow{\sharp_1} nf(\overset{M_2, \sharp_2(\eta)}{\Rightarrow}, in_2(\varepsilon))$.
    - 任意の $w \in path(\eta), a_1' \in Syn_1, b_2 \in Inh_2, a_1'(\pi 1) = sub_\eta(w)$ に対して，

$$\langle a_1', b_2 \rangle(\pi 1) \xrightarrow{\sharp_1} nf(\overset{M_2, \sharp_2(\eta)}{\Rightarrow}, b_2(1w)).$$

  - 任意の $b_1 \in Inh_1$ と $b_1(\pi 1) \xrightarrow{\sharp_1} \eta \in R_1$ に対して:
    - 任意の $a_2 \in Syn_2$ に対して，

$$\langle b_1, a_2 \rangle(\pi 1) \xrightarrow{\sharp_1} nf(\overset{M_2, \eta}{\Rightarrow}, a_2(\varepsilon))[b_2(\varepsilon) := \langle b_1, b_2' \rangle(\pi 1)]_{b_2' \in Inh_2}.$$

    - 任意の $w \in path(\eta), a_1' \in Syn_1, b_2 \in Inh_2, a_1'(\pi 1) = sub_\eta(w)$ に対して，

$$\langle a_1', b_2 \rangle(\pi 1) \xrightarrow{\sharp_1} nf(\overset{M_2, \eta}{\Rightarrow}, b_2(w))[b_2'(\varepsilon) := \langle b_1, b_2' \rangle(\pi 1)]_{b_2' \in Inh_2}.$$

上の M_2 が定義可能なら M_2' も定義可能であるから，$M_1 \fatsemi M_2$ は唯 1 つ存在し，定義可能である．ATT の合成の正当性は Ganzinger らによって示されている [11, 10, 15].

定理 3.9 (DC の正当性)**.** 任意の定義可能な ATT M_1, M_2 に対して，M_1 と M_2 の両方が SUR を満たすなら，$M_1 \fatsemi M_2$ は SUR を満たす定義可能な ATT であり，$\mathbf{T}_{M_1 \fatsemi M_2} = \mathbf{T}_{M_1} \circ \mathbf{T}_{M_2}$ を満たす．M_2 が SUR を満たすなら，$M_1 \fatsemi M_2$ は定義可能な ATT であり，$\mathbf{T}_{M_1 \fatsemi M_2} = \mathbf{T}_{M_1} \circ \mathbf{T}_{M_2}$ を満たす.

定理 3.10. 任意の定義可能な ATT M_1, M_2 に対して，M_1 が継承属性を持たないとき，$M_1 \fatsemi M_2$ は定義可能な ATT であり，$\mathbf{T}_{M_1 \fatsemi M_2} = \mathbf{T}_{M_1} \circ \mathbf{T}_{M_2}$ を満たす.

継承属性を持たない ATT を**降下型木変換器**（*top-down tree transducer*, TDTT）と呼び，全ての定義可能な TDTT による木変換の集合を $TDTT$ で表す．$TDTT \subsetneq ATT$ [7, 8] が知られており，この証明の方法を 5 章で ASTT に応用する.

以上から即座に次が導かれる.

系 3.11.

1. $ATT_{su} \circ ATT_{su} = ATT_{su}$.
2. $ATT_{su} \circ ATT = ATT$.
3. $ATT \circ TDTT = ATT$.

例 3.12. ATT M_{leaves} と M_{rev} に対して，M_{leaves} が SUR を満たすので

$$M_{leaves} \fatsemi M_{rev} = (Syn, Inh, \{\mathrm{bin}^{(2)}, \mathrm{A}^{(0)}, \mathrm{B}^{(0)}\}, \{\mathrm{A}^{(1)}, \mathrm{B}^{(1)}, \epsilon^{(0)}\}, \langle a_1, a_2\rangle, \sharp_1, R)$$

は ATT であり，次を満たす:

$$\begin{aligned}
Syn &= \{a_1\} \times \{a_2\} \cup \{b_1\} \times \{b_2\} = \{\langle a_1, a_2\rangle, \langle b_1, b_2\rangle\} \\
Inh &= \{a_1\} \times \{b_2\} \cup \{b_1\} \times \{a_2\} = \{\langle a_1, b_2\rangle, \langle b_1, a_2\rangle\} \\
R = \{\ & \langle a_1, a_2\rangle(\pi) \xrightarrow{\sharp_1} \langle a_1, a_2\rangle(\pi 1) \qquad \langle a_1, a_2\rangle(\pi) \xrightarrow{\mathrm{B}} \langle b_1, a_2\rangle(\pi) \\
& \langle a_1, b_2\rangle(\pi 1) \xrightarrow{\sharp_1} \epsilon \qquad\qquad \langle b_1, b_2\rangle(\pi) \xrightarrow{\mathrm{B}} (\mathrm{B}\langle a_1, b_2\rangle(\pi)) \\
& \langle a_1, a_2\rangle(\pi) \xrightarrow{\mathrm{bin}} \langle a_1, a_2\rangle(\pi 1) \qquad \langle b_1, a_2\rangle(\pi 1) \xrightarrow{\sharp_1} \langle b_1, b_2\rangle(\pi 1) \\
& \langle a_1, a_2\rangle(\pi) \xrightarrow{\mathrm{A}} \langle b_1, a_2\rangle(\pi) \qquad \langle b_1, a_2\rangle(\pi 1) \xrightarrow{\mathrm{bin}} \langle a_1, a_2\rangle(\pi 2) \\
& \langle b_1, b_2\rangle(\pi) \xrightarrow{\mathrm{A}} \mathrm{A}(\langle a_1, b_2\rangle(\pi)) \quad \langle b_1, a_2\rangle(\pi 2) \xrightarrow{\mathrm{bin}} \langle b_1, a_2\rangle(\pi) \quad \}.
\end{aligned}$$

各規則は次のように導かれる:

- $(a_1(\pi) \xrightarrow{\sharp_1} a_1(\pi 1)) \in R_1$ に対して
  - a_2 を適用して

$$a_2(\varepsilon) \overset{M_{rev}', \sharp_2(a_1(\pi 1))}{\Rightarrow} a_2(1) \qquad (\because (a_2(\pi) \xrightarrow{\sharp_2} a_2(\pi 1)) \in R_2)$$

$$\overset{M'_{rev},\sharp_2(a_1(\pi 1))}{\Rightarrow} \langle a_1, a_2\rangle(\pi 1)$$

よって $(\langle a_1, a_2\rangle(\pi) \overset{\sharp_1}{\to} \langle a_1, a_2\rangle(\pi 1)) \in R$.

– b_2 を適用して

$$b_2(1) \overset{M'_{rev},\sharp_2(a_1(\pi 1))}{\Rightarrow} \epsilon \qquad (\because (b_2(\pi 1) \overset{\sharp_2}{\to} \epsilon\,) \in R_2)$$

よって $(\langle a_1, b_2\rangle(\pi 1) \overset{\sharp_1}{\to} \epsilon\,) \in R$.

- $(a_1(\pi) \overset{\text{bin}}{\to} a_1(\pi 1)) \in R_1$ に対して,
  – a_2 を適用して

$$a_2(\varepsilon) \overset{M'_{rev},a_1(\pi 1)}{\Rightarrow} \langle a_1, a_2\rangle(\pi 1)$$

よって $(\langle a_1, a_2\rangle(\pi) \overset{\text{bin}}{\to} \langle a_1, a_2\rangle(\pi 1)) \in R$.

– b_2 を適用して，$b_2(\varepsilon)$ を $\langle a_1, b_2\rangle(\pi)$ に置換し，$(\langle a_1, b_2\rangle(\pi) \overset{\text{bin}}{\to} \langle a_1, b_2\rangle(\pi)) \in R$.

- $(a_1(\pi) \overset{\text{A}}{\to} \text{A}(b_1(\pi))) \in R_1$ に対して
  – a_2 を適用して

$$a_2(\varepsilon) \overset{M'_{rev},\text{A}(b_1(\pi))}{\Rightarrow} a_2(1) \qquad (\because (a_2(\pi) \overset{\text{A}}{\to} a_2(\pi 1)) \in R_2)$$
$$\overset{M'_{rev},\text{A}(b_1(\pi))}{\Rightarrow} \langle b_1, a_2\rangle(\pi)$$

よって $(\langle a_1, a_2\rangle(\pi) \overset{\text{A}}{\to} \langle b_1, a_2\rangle(\pi)) \in R$.

– b_2 を適用して

$$b_2(1) \overset{M'_{rev},\text{A}(b_1(\pi))}{\Rightarrow} \text{A}(b_2(\varepsilon)) \qquad (\because (b_2(\pi 1) \overset{\text{A}}{\to} \text{A}(b_2(\pi))) \in R_2)$$

$b_2(\varepsilon)$ を $\langle a_1, b_2\rangle(\pi)$ に置換し，$(\langle b_1, b_2\rangle(\pi) \overset{\text{A}}{\to} \text{A}(\langle a_1, b_2\rangle(\pi))) \in R$.

- $(a_1(\pi) \overset{\text{B}}{\to} \text{B}(b_1(\pi))) \in R_1$ に対して
  – a_2 を適用して

$$a_2(\varepsilon) \overset{M'_{rev},\text{B}(b_1(\pi))}{\Rightarrow} a_2(1) \qquad (\because (a_2(\pi) \overset{\text{B}}{\to} a_2(\pi 1)) \in R_2)$$
$$\overset{M'_{rev},\text{B}(b_1(\pi))}{\Rightarrow} \langle b_1, a_2\rangle(\pi)$$

よって $(\langle a_1, a_2\rangle(\pi) \overset{\text{B}}{\to} \langle b_1, a_2\rangle(\pi)) \in R$.

– b_2 を適用して

$$b_2(1) \overset{M'_{rev},\text{B}(b_1(\pi))}{\Rightarrow} \text{B}(b_2(\varepsilon)) \qquad (\because (b_2(\pi 1) \overset{\text{A}}{\to} \text{B}(b_2(\pi))) \in R_2)$$

$b_2(\varepsilon)$ を $\langle a_1, b_2\rangle(\pi)$ に置換し，$(\langle b_1, b_2\rangle(\pi) \overset{\text{B}}{\to} \text{B}(\langle a_1, b_2\rangle(\pi))) \in R$.

- $(b_1(\pi 1) \overset{\sharp_1}{\to} \epsilon) \in R_1$ に対して a_2 を適用して

$$a_2(\varepsilon) \overset{M'_{rev},\epsilon}{\Rightarrow} b_2(\varepsilon) \qquad (\because (a_2(\pi) \overset{\epsilon}{\to} b_2(\pi)) \in R_2)$$

  $b_2(\varepsilon)$ を $\langle b_1, b_2\rangle(\pi 1)$ に置換し，$(\langle b_1, a_2\rangle(\pi 1) \overset{\sharp_1}{\to} \langle b_1, b_2\rangle(\pi 1)) \in R$.
- $(b_1(\pi 1) \overset{\text{bin}}{\to} a_1(\pi 2)) \in R_1$ に対して
  - a_2 を適用して

$$a_2(\varepsilon) \overset{M'_{rev},a_1(\pi 2)}{\Rightarrow} \langle a_1, a_2\rangle(\pi 2)$$

    よって $(\langle b_1, a_2\rangle(\pi 1) \overset{\text{bin}}{\to} \langle a_1, a_2\rangle(\pi 2)) \in R$.
  - b_2 を適用して，$b_2(\varepsilon)$ を $\langle b_1, b_2\rangle(\pi 1)$ に置換し，$(\langle a_1, b_2\rangle(\pi 2) \overset{\text{bin}}{\to} \langle b_1, b_2\rangle(\pi 1)) \in R$.
- $(b_1(\pi 2) \overset{\text{bin}}{\to} b_1(\pi)) \in R_1$ に対して
  - a_2 を適用して

$$a_2(\varepsilon) \overset{M'_{rev},b_1(\pi)}{\Rightarrow} \langle b_1, a_2\rangle(\pi)$$

    よって $(\langle b_1, a_2\rangle(\pi 2) \overset{\text{bin}}{\to} \langle b_1, a_2\rangle(\pi)) \in R$.
  - b_2 を適用して，$b_2(\varepsilon)$ を $\langle b_1, b_2\rangle(\pi 2)$ に置換し，$(\langle b_1, b_2\rangle(\pi) \overset{\text{bin}}{\to} \langle b_1, b_2\rangle(\pi 2)) \in R$.

$M_{leaves} \fatsemi M_{rev}$ に $\mathbf{T}_{M_{leaves}} \circ \mathbf{T}_{M_{rev}}(\text{bin}(\text{bin}(\text{A},\text{B}),\text{bin}(\text{bin}(\text{B},\text{A}),\text{A}))) = \text{A}(\text{A}(\text{B}(\text{B}(\text{A}(\epsilon)))))$ と同じ入力を与えると，図 3.5に示す簡約で計算される．

ATT の合成アルゴリズムは MTT の合成に応用される．MTT は，蓄積引数を伴う相互再帰関数の集まりで，多引数関数の再帰プログラムの関数定義をモデル化したものである．MTT は ATT と同様に，呼び出す関数と節点のラベルに応じて規則を選択し，関数呼び出しを規則の右辺で置換する．一方で ATT と異なり，木を親節点から子節点へ下向きに巡回する．関数の第 1 引数に入力木の節点のラベルと，その節点の子節点を束縛し，第 2 引数以降には蓄積引数が並ぶ．規則の右辺では再帰呼び出しや蓄積引数の参照が可能であり，蓄積引数には木を渡すことができる．例えば，二分木の葉を並べる再帰プログラムは，1 つの関数 *leaves* を用いて MTT で次のように表現できる:

$$\begin{aligned} \mathit{leaves}(\text{bin}(x_1, x_2), y) &\to \mathit{leaves}(x_1, (\mathit{leaves}(x_2, y))) \\ \mathit{leaves}(\text{A}, y) &\to \text{A}(y) \\ \mathit{leaves}(\text{B}, y) &\to \text{B}(y) \end{aligned}$$

y が蓄積引数である．ATT の継承属性による計算は，MTT において蓄積引数に木構造を蓄積する計算に対応付けることができるので，ATT による計算は MTT で表現できる．そこで，制限を課した MTT を ATT に変換し，変換した 2 つの ATT に対して DC を施し，得られる 1 つの ATT を MTT に逆変換する手法 [16] が提案された．さらにこれを発展させて，ATT を経由せずに MTT を直接合成する手法 [25] が考案されている．

$\langle a_1, a_2\rangle(\varepsilon) \Rightarrow \langle a_1, a_2\rangle(1)$
$\Rightarrow \langle a_1, a_2\rangle(11)$
$\Rightarrow \langle a_1, a_2\rangle(111)$
$\Rightarrow \langle b_1, a_2\rangle(111)$
$\Rightarrow \langle a_1, a_2\rangle(112)$
$\Rightarrow \langle b_1, a_2\rangle(112)$
$\Rightarrow \langle b_1, a_2\rangle(11)$
$\Rightarrow \langle a_1, a_2\rangle(12)$
$\Rightarrow \langle a_1, a_2\rangle(121)$
$\Rightarrow \langle a_1, a_2\rangle(1211)$
$\Rightarrow \langle b_1, a_2\rangle(1211)$
$\Rightarrow \langle a_1, a_2\rangle(1212)$
$\Rightarrow \langle b_1, a_2\rangle(1212)$
$\Rightarrow \langle b_1, a_2\rangle(121)$
$\Rightarrow \langle a_1, a_2\rangle(122)$
$\Rightarrow \langle b_1, a_2\rangle(122)$
$\Rightarrow \langle b_1, a_2\rangle(12)$
$\Rightarrow \langle b_1, a_2\rangle(1)$
$\Rightarrow \langle b_1, b_2\rangle(1)$
$\Rightarrow \langle b_1, b_2\rangle(12)$
$\Rightarrow \langle b_1, b_2\rangle(122)$ （右上に続く）

$\Rightarrow \mathrm{A}(\langle a_1, b_2\rangle(122))$
$\Rightarrow \mathrm{A}(\langle b_1, b_2\rangle(121))$
$\Rightarrow \mathrm{A}(\langle b_1, b_2\rangle(1212))$
$\Rightarrow \mathrm{A}(\mathrm{A}(\langle a_1, b_2\rangle(1212)))$
$\Rightarrow \mathrm{A}(\mathrm{A}(\langle b_1, b_2\rangle(1211)))$
$\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\langle a_1, b_2\rangle(1211))))$
$\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\langle a_1, b_2\rangle(121))))$
$\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\langle a_1, b_2\rangle(12))))$
$\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\langle b_1, b_2\rangle(11))))$
$\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\langle b_1, b_2\rangle(112))))$
$\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\mathrm{B}(\langle a_1, b_2\rangle(112)))))$
$\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\mathrm{B}(\langle b_1, b_2\rangle(111)))))$
$\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(\langle a_1, b_2\rangle(111)))))))$
$\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(\langle a_1, b_2\rangle(11)))))))$
$\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(\langle a_1, b_2\rangle(1)))))))$
$\Rightarrow \mathrm{A}(\mathrm{A}(\mathrm{B}(\mathrm{B}(\mathrm{A}(\epsilon)))))$

図 3.5: ASTT $M_{leaves} \fatsemi M_{rev}$ による簡約例

ATT や MTT では，有限のランク付き文字集合しか扱えないため，例えば M_{leaves} の例では A と B の 2 種類の文字を，2 分木の葉節点のラベルとして用いている．葉節点のラベルとして 2 種類の文字だけでなく，例えば全ての整数値を取り得るような場合を考えると，無限の種類の整数値に対して規則を記述しなければならないため，有限の記述に限定される ATT や MTT では表現できない．

4　記号的木変換器

TDTT を拡張し，従来の木変換器で扱えなかった無限のランク付き文字集合を扱えるようにした**記号的木変換器**（*symbolic tree transducer*, STT） [24] が提案されている．STT では，1 つの規則は 1 種類の文字ではなく文字の集合に対応する．その集合は述語で表現されるので，無限集合でもよい．従って STT では，無限の種類の入力文字に対応する規則を，有限の記述で表現できる．また，規則の右辺のラベルも 1 種類の文字ではなく，入力文字に応じて変えることができる．具体的には，右辺のラベルはランク付き文字集合上の単項の関数である．この関数の像は無限でもよいから，出力文字もまた無限の種類を取り得る．

本章では，記号的木変換器を導入し，その合成アルゴリズムを紹介する．本論文で提案する ASTT の合成アルゴリズムは，STT の合成アルゴリズムの自然な応用である．

4.1　記号的木変換器の定義

X を集合とし，$n \in \mathbb{N}, x_0, \ldots, x_n \in X$ に対して $X_n \stackrel{\text{def}}{=} \{x_0, \ldots, x_n\}$ と定義する．また，X が変数の集合のときは，有限集合 Q に対して，$Q(X) \stackrel{\text{def}}{=} \{q(x) \mid q \in Q, x \in X\}$ と定義する．

まず，STT の構文を定義する．

定義 4.1 (記号的木変換器)**.** 記号的木変換器（STT）とは，次を満たす 6 つ組 $(Q, \Sigma, \Phi, \Delta, q_0, R)$ である:

- Q は有限集合であり，その要素を**状態**と呼ぶ.
- Σ と Δ は $Q \cap (\Sigma \cup \Delta) = \emptyset$ を満たす可算のランク付き文字集合であり，それぞれ入力ランク付き文字集合，出力ランク付き文字集合と呼ぶ．さらに (Σ, Φ) はラベル構造である.
- $q_0 \in Q$ を初期状態と呼ぶ.
- $R \subseteq Q \times \mathrm{BC}(\Phi) \times X^* \times T_{F(\Sigma \to \Delta)}(Q(X))$ は以下を満たす:
  - 任意の $l \in rk(\Sigma), q \in Q$ に対して $(q, \varphi^{(l)}, x_1 \cdots x_l, \eta) \in R$ を満たす $\varphi^{(l)} \in \mathrm{BC}(\Phi)$, $x_1, \ldots, x_l \in X$，$\eta \in T_{F(\Sigma \to \Delta)}(Q(X_l))$ が存在する.

  なお，$\rho = (q, \varphi^{(l)}, x_1 \cdots x_l, \eta) \in R$ を**規則**と呼び，次の形で表す: $q(\varphi(x_1, \ldots, x_l)) \to \eta$. また，$(q, l)$，$\varphi$，$\eta$ をそれぞれ $\mathrm{lhs}(\rho)$，$\mathrm{grd}(\rho)$，$\mathrm{rhs}(\rho)$ で表し，それぞれを規則 ρ の**左辺**，**ガード**，**右辺**と呼ぶ.
- 任意の $\rho_1, \rho_2 \in R$ に対して $\mathrm{lhs}(\rho_1) = \mathrm{lhs}(\rho_2) \implies [\![\mathrm{grd}(\rho_1)]\!] \cap [\![\mathrm{grd}(\rho_2)]\!] = \emptyset$. これは STT が決定的であることを意味する.

- 任意の $l \in rk(\Sigma), q \in Q$ に対して

$$\bigcup_{\substack{\rho \in R \\ \mathrm{lhs}(\rho)=(q,l)}} [\![\mathrm{grd}(\rho)]\!] = \Sigma.$$

  これは STT が全域的であることを意味する．

最後の 2 つの項目により，本論文では専ら決定的かつ全域的な STT を考える．つまり同じ左辺の規則のガードは互いに交わらず，かつそれらの和が入力ランク付き文字集合を網羅する．したがって，明らかに次が成立する．

命題 4.2 (STT の決定性と全域性)**.** 任意の STT $M = (Q, \Sigma, \Phi, \Delta, q_0, R)$ は次を満たす: 任意の $\sigma \in \Sigma, (q, l) \in Q \times rk(\Sigma)$ に対して，$(q, l) = \mathrm{lhs}(\rho), \sigma \in [\![\mathrm{grd}(\rho)]\!]$ を満たす $\rho \in R$ が唯 1 つ存在する．

STT M の任意の規則に対して，各変数 x_i の出現がそれぞれ高々 1 回のとき，M は**線形**であるといい，逆に少なくとも 1 回のとき，M は**非削除的**であるという．

例 4.3. STT $M_{abs} = (\{q_1\}, \mathbb{Z}^{(1)} \cup \{\epsilon^{(0)}\}, \{(<0)^{(1)}\}, \mathbb{N}^{(1)} \cup \{\epsilon^{(0)}\}, q_1, R_1)$ と STT $M_{(\div 2)} = (\{q_2\}, \mathbb{N}^{(1)} \cup \{\epsilon^{(0)}\}, \{even^{(1)}\}, \mathbb{N}^{(1)} \cup \{\epsilon^{(0)}\}, q_2, R_2)$ を次の R_1 と R_2 で定義する．

$$R_1 = \{ \begin{array}{r} q_1((<0)(x)) \to abs(q_1(x)) \\ q_1((\neg(<0))(x)) \to id(q_1(x)) \\ q_1(\top^{(0)}) \to \hat{\epsilon} \end{array} \}$$

$$R_2 = \{ \begin{array}{r} q_2(even(x)) \to (\div 2)(q_2(x)) \\ q_2((\neg even)(x)) \to id(q_2(x)) \\ q_2(\top^{(0)}) \to \hat{\epsilon} \end{array} \}$$

ただし，(<0) は負の整数で真になる述語，$even$ は偶数で真になる述語，abs は整数の絶対値を取る関数，id は恒等関数，$(\div 2)$ は自然数を 2 で割る関数である．M_{abs} はリストを巡回し，負数をその絶対値で置き換える．$M_{(\div 2)}$ はリストを巡回し，偶数をその半分の値で置き換える．

次に，STT の意味論を定義する．

定義 4.4 (STT による導出関係)**.** $M = (Q, \Sigma, \Phi, \Delta, q_0, R)$ を STT として，$\xi \in T_\Sigma$ とする．このとき，M による**導出関係**を，$T_\Delta(Q(T_\Sigma))$ 上の最小の二項関係 $\overset{M}{\Rightarrow}$ で定義する:

- $q(\sigma(\xi_1, \ldots, \xi_l)) \overset{M}{\Rightarrow} \eta'[x_1, \ldots, x_l := \xi_1, \ldots, \xi_l] \iff q \in Q, (q(\varphi(x_1, \ldots, x_l)) \to \eta) \in R, \sigma \in [\![\varphi]\!], \eta' = give_\eta(\sigma)$.
- $\delta(\eta_1, \ldots, \eta_i, \ldots, \eta_l) \overset{M}{\Rightarrow} \delta(\eta_1, \ldots, \eta_i', \ldots, \eta_l) \iff \delta \in \Delta, \eta_i \overset{M}{\Rightarrow} \eta_i', \forall j \in [i-1]. \eta_j \in T_\Delta$.

ただし，導出関係 $\overset{M}{\Rightarrow}$ において M が明らかなときは，それを省略して $\Rightarrow$ と書く．

命題 4.2 より，1 つ目の項目で選ばれる規則は一意である．さらに 2 つ目の項目より，η_i の左側の全ての木が変数を含まない簡約済みの木であるから，この関係による簡約は最外最左簡約である．よって，この導出関係は決定的である．また，規則の右辺に現れる $q(x)$ の x は，簡約す

$q_1(1(-2(-3(4(5(\varepsilon))))))$
$\Rightarrow 1(q_1(-2(-3(4(5(\varepsilon))))))$
$\Rightarrow 1(2(q_1(-3(4(5(\varepsilon))))))$
$\Rightarrow 1(2(3(q_1(4(5(\varepsilon))))))$
$\Rightarrow 1(2(3(4(q_1(5(\varepsilon))))))$
$\Rightarrow 1(2(3(4(5(q_1(\varepsilon))))))$
$\Rightarrow 1(2(3(4(5(\varepsilon)))))$

(a) M_{abs} による簡約例

$q_2(1(2(3(4(5(\varepsilon))))))$
$\Rightarrow 1(q_2(2(3(4(5(\varepsilon))))))$
$\Rightarrow 1(1(q_2(3(4(5(\varepsilon))))))$
$\Rightarrow 1(1(3(q_2(4(5(\varepsilon))))))$
$\Rightarrow 1(1(3(2(q_2(5(\varepsilon))))))$
$\Rightarrow 1(1(3(2(5(q_2(\varepsilon))))))$
$\Rightarrow 1(1(3(2(5(\varepsilon)))))$

(b) $M_{(\div 2)}$ による簡約例

図 4.1: 2 つの STT による簡約例

る木の部分木で置換されるから，この導出関係は必ず停止する．従って，この導出関係による簡約系では，有限の木に対して必ず正規形が存在し，さらにそれは一意である．

定義 4.5. $M = (Q, \Sigma, \Phi, \Delta, q_0, R)$ を STT とすると，M による**木変換**とは，次で定義する関数 $\mathbf{T}_M : T_\Sigma \to T_\Delta$ である: 任意の $\xi \in T_\Sigma$ に対して，$\mathbf{T}_M(\xi) = nf(\overset{M}{\Rightarrow}, q_0(\xi))$.

全ての STT による木変換の集合を STT で表す．

例 4.6. リスト $\xi = 1(-2(-3(4(5(\varepsilon))))) \in T_{\mathbb{Z}\cup\{\epsilon\}}$ に対して，$\mathbf{T}_{M_{abs}}(\xi) = nf(\overset{M_{abs}}{\Rightarrow}, q_1(\xi)) = 1(2(3(4(5(\epsilon))))) \in T_{\mathbb{N}\cup\{\epsilon\}}$ であり，これは図 4.1(a) の簡約で計算される．さらにこの木を ζ とすると，$\mathbf{T}_{M_{even}}(\zeta) = nf(\overset{M_{even}}{\Rightarrow}, q_2(\zeta)) = 1(1(3(2(5(\epsilon))))) \in T_{\mathbb{N}\cup\{\epsilon\}}$ であり，これは図 4.1(b) の簡約で計算される．従って $\mathbf{T}_{M_{abs}} \circ \mathbf{T}_{M_{(\div 2)}}(1(-2(-3(4(5(\varepsilon)))))) = 1(1(3(2(5(\epsilon)))))$ である．

STT は無限のランク付き文字集合を扱えるので，ATT や MTT では表現できないが STT では表現できる計算が存在する．しかし，STT は TDTT の拡張であるから，関数の引数は 1 つだけであり，親節点から子節点へ下向きにしか巡回できない．例えば 2 分木を巡回するなら，中間節点において片方の部分木を巡回して得られる結果を他方の部分木の巡回で利用することができないので，ATT M_{leaves} のように 2 分木の葉節点のラベルを並べることはできない．また，リストの末尾から先頭に向かって巡回したり，蓄積引数に要素を追加したりすることができないので，M_{rev} のようにリストを反転させることもできない．このように，継承属性や蓄積引数を利用しないと表現できない計算が存在する．よって，ATT や MTT で表現できるが STT では表現できない計算もある．これは ATT や MTT と TDTT の関係と同じである．

4.2 記号的木変換器が表現する計算のクラス

STT は TDTT の拡張であるから，TDTT から幾つかの性質を受け継ぐ．TDTT の主な性質の 1 つに**樹高特性**がある．これは，TDTT の出力木の高さが，入力木の高さの定数倍で抑えられるという性質である．STT は TDTT と比べて，ラベルの処理が異なるだけなので，出力木の高さに関しては TDTT と同じ性質を満たす．本節では，STT と ASTT が表現する計算のクラスの比較のために，STT の 樹高特性を示す．証明の方法は，Fül"op ら [8] の方法と同様である．

樹高特性の前に補助的な命題を 1 つ示す．

命題 4.7. $M = (Q, \Sigma, \Phi, \Delta, q_0, R)$ を STT とする．任意の $n \in rk(\Sigma)$, $\eta \in T_{F(\Sigma \to \Delta)}(Q(X_n))$, $\xi_1, \ldots, \xi_n$ に対して，

$$nf(\overset{M}{\Rightarrow}, \eta[x_i := \xi_i]_{i \in [n]}) = \eta[q(x_i) := nf(\overset{M}{\Rightarrow}, q(\xi_i))]_{q(x_i) \in Q(X_n)}$$

証明. η に関する構造帰納法で示す．

$\eta = p(x_j)$ のとき　$p \in Q$，$x_j \in X_n$ とする．

$$\begin{aligned} nf(\overset{M}{\Rightarrow}, \eta[x_i := \xi_i]_{i \in [n]}) &= nf(\overset{M}{\Rightarrow}, p(\xi_j)) \\ &= p(x_j)[q(x_i) := nf(\overset{M}{\Rightarrow}, q(\xi_i))]_{q(x_i) \in Q(X_n)} \qquad \text{(置換の定義より)} \\ &= \eta[q(x_i) := nf(\overset{M}{\Rightarrow}, q(\xi_i))]_{q(x_i) \in Q(X_n)} \end{aligned}$$

$\eta = \delta(\eta_1, \ldots, \eta_l)$ のとき

$$\begin{aligned} nf(\overset{M}{\Rightarrow}, \eta[x_i := \xi_i]_{i \in [n]}) &= \delta(nf(M, \eta_1[x_i := \xi_i]_{i \in [n]}), \ldots, nf(\overset{M}{\Rightarrow}, \eta_l[x_i := \xi_i]_{i \in [n]})) \\ &= \delta(\eta_1[q(x_i) := nf(\overset{M}{\Rightarrow}, q(\xi_i))]_{q(x_i) \in Q(X_n)}, \cdots \\ &\quad \ldots, \eta_l[q(x_i) := nf(\overset{M}{\Rightarrow}, q(\xi_i))]_{q(x_i) \in Q(X_n)}) \qquad \text{(帰納法の仮定より)} \\ &= \delta(\eta_1, \ldots, \eta_l)[q(x_i) := nf(\overset{M}{\Rightarrow}, q(\xi_i))]_{q(x_i) \in Q(X_n)} \\ &= \eta[q(x_i) := nf(\overset{M}{\Rightarrow}, q(\xi_i))]_{q(x_i) \in Q(X_n)} \end{aligned}$$

□

次に，STT の樹高特性を示す．

命題 4.8 (STT の樹高特性)**.** $M = (Q, \Sigma, \Phi, \Delta, q_0, R)$ を STT とする．このとき，$c \in \mathbb{N}$ が存在し，任意の $\xi \in T_\Sigma$ に対して $ht(\mathbf{T}_M(\xi)) \leq c \cdot ht(\xi)$ が成立する．

証明. $c = \max\{ht(\mathrm{rhs}(\rho)) \mid \rho \in R\}$ を M の右辺の高さの最大値とする．このとき，任意の $q \in Q$ と $\xi \in T_\Sigma$ に対して $ht(nf(\Rightarrow M, q(\xi))) \leq c \cdot ht(\xi)$ が成立することを，ξ に関する構造帰納法で示す．

$\xi = \sigma$ のとき　命題 4.2 より，$\sigma \in [\![\mathrm{grd}(\rho)]\!]$ を満たす $\rho \in R$ が唯 1 つ存在し，$ht(nf(\overset{M}{\Rightarrow}, q(\xi))) = ht(\mathrm{rhs}(\rho)) \leq c = c \cdot ht(\xi)$.

$\xi = \sigma(\xi_1, \ldots, \xi_l)$ のとき　命題 4.2 より，$\sigma \in [\![\mathrm{grd}(\rho)]\!]$ を満たす $\rho \in R$ が唯 1 つ存在する．このとき，

$$\begin{aligned}
ht(nf(\overset{M}{\Rightarrow}, q(\xi))) &= ht(\mathrm{rhs}(\rho)[x_i := \xi_i]_{i\in[l]}) \\
&= ht(\mathrm{rhs}(\rho)[q(x_i) := nf(\overset{M}{\Rightarrow}, q(\xi_i))]_{q(x_i)\in Q(X_n)}) && (\text{命題 4.7 より}) \\
&\leq ht(\mathrm{rhs}(\rho)) + \max\{ht(nf(\overset{M}{\Rightarrow}, q(\xi_i))) \mid q(x_i) \in Q(X_l)\} && (ht \text{ の定義より}) \\
&\leq c + \max\{ht(nf(\overset{M}{\Rightarrow}, q(\xi_i))) \mid q(x_i) \in Q(X_l)\} && (c \text{ の定義より}) \\
&\leq c + \max\{c \cdot ht(\xi_i) \mid i \in [l]\} && (\text{帰納法の仮定より}) \\
&= c \cdot (1 + \max\{ht(\xi_i) \mid i \in [l]\}) \\
&= c \cdot ht(\xi). && (ht \text{ の定義より})
\end{aligned}$$

□

4.3　記号的木変換器の合成

STT は TDTT の拡張であるから，合成アルゴリズムも TDTT の合成に基づく．しかし TDTT の合成は ATT の合成の特別な場合であるから，STT の合成アルゴリズムから ASTT の合成に応用すべき要素は，次に示す特別な導出関係である．

定義 4.9 (記号的導出関係)**.** 任意の STT M_1 と STT M_2 に対して，

$$M_1 = (Q_1, \Sigma_1, \Phi_1, \Delta_1, q_1', R_1)$$
$$M_2 = (Q_2, \Sigma_2, \Phi_2, \Delta_2, q_2', R_2)$$

かつ $\Delta_1 \subseteq \Sigma_2$ のとき，M_2 による**記号的導出関係**を，

$$\mathrm{Pred}_0(\Sigma_1) \times T_{F(\Sigma_1\to\Delta_1)}(Q_2(T_{F(\Sigma_2\to\Delta_2)}(Q_1(X))) \cup (Q_1 \times Q_2)(X))$$

の上の最小の二項関係 $\overset{M_2}{\Rightarrow}_S$ で定義する:

- $(\theta, q_2(f_1(\xi_1, \ldots, \xi_l))) \overset{M_2}{\Rightarrow}_S (\theta \wedge f_1 \circ \varphi_2, \zeta'[x_1, \ldots, x_l := \xi_1, \ldots, \xi_l])$
  $\iff \theta \in \mathrm{Pred}_0(\Sigma_1), q_2 \in Q_2, (q_2(\varphi_2(x_1, \ldots, x_l)) \to \zeta) \in R_2, \zeta' = comp_\zeta(f_1)$
- $(\theta, f(\zeta_1, \ldots, \zeta_j, \ldots, \zeta_l)) \overset{M_2}{\Rightarrow}_S (\theta', f(\zeta_1, \ldots, \zeta_j', \ldots, \zeta_l))$
  $\iff f \in F(\Sigma_1 \to \Delta_2), (\theta, \zeta_j) \overset{M_2,\eta}{\Rightarrow}_S (\theta', \zeta_j'), \forall j \in [i-1].\, \eta_j \in T_{F(\Sigma_1\to\Delta_2)}$

- $(\theta, q_2(q_1(x))) \overset{M_2}{\Rightarrow}_S (\theta, \langle q_1, q_2\rangle(x)) \iff \theta \in \mathrm{Pred}_0(\Sigma_1), q_1 \in Q_1, q_2 \in Q_2$

記号的導出関係では，M_1 の規則の右辺を簡約するのに使用する M_2 の規則のガードを追加していくことで，述語で表現されたランク付き文字集合を適切に処理している．ガードを追加するときは，M_1 の右辺の関数と合成してから追加する．

定義 4.10 (STT の合成)**.** 任意の STT M_1 と M_2 に対して，

$$M_1 = (Q_1, \Sigma_1, \Phi_1, \Delta_1, q_1', R_1)$$
$$M_2 = (Q_2, \Sigma_2, \Phi_2, \Delta_2, q_2', R_2)$$

かつ $\Delta_1 \subseteq \Sigma_2$ のとき，まず，述語の集合 Φ を次で定義する:

$$\begin{aligned}\Phi = \Phi_1 \cup \{\, f_1 \circ \varphi_2 \mid & f_1^{(l)} \in F(\Sigma_1 \to \Delta_2), \varphi_2^{(l)} \in \mathrm{BC}(\Phi_2),\\ & \exists \rho_1 \in R_1, w \in \mathit{path}(\mathrm{rhs}(\rho_1)).\, \mathit{lab}_{\mathrm{rhs}(\rho_1)}(w) = f_1\\ & \exists \rho_2 \in R_2.\, \mathrm{grd}(\rho_2) = \varphi_2\}.\end{aligned}$$

次に，$M_1 \fatsemi M_2$ を，次を満たす $(Q, \Sigma_1, \Phi, \Delta_2, \langle q_1', q_2'\rangle, R)$ で定義する: 任意の規則 $(q_1(\varphi_1(x_1, \ldots, x_l)) \to \eta) \in R_1$ と $q_2 \in Q_2$，正規形 $(\varphi_1, q_2(\eta))$ $(\overset{M_2}{\Rightarrow}_S)^*$ (θ, ζ) に対して，$[\![\theta]\!] \neq \emptyset$ ならば，$(\langle q_1, q_2\rangle(\theta(x_1, \ldots, x_l)) \to \zeta) \in R$.

決定性と全域性を仮定しない一般の STT の合成に関して，次が示されている．

定理 4.11 (STT の合成の正当性)**.** 任意の STT M_1 と M_2 に対して，

1. M_1 が決定的または M_2 が線形，かつ
2. M_1 が全域的または M_2 が非削除的

ならば，$M_1 \fatsemi M_2$ は STT であり，$\mathbf{T}_{M_1 \fatsemi M_2} = \mathbf{T}_{M_1} \circ \mathbf{T}_{M_2}$.

定義 4.1 は決定性と全域性を含んでいるので，本論文では任意の 2 つの STT M_1，M_2 に対して上の条件 1 と条件 2 が成立することを仮定している．

例 4.12. STT M_{abs} と $M_{(\div 2)}$ に対して，$M_{abs} \fatsemi M_{(\div 2)} = (\{\langle q_1, q_2\rangle\}, \mathbb{N}^{(1)} \cup \{\epsilon^{(0)}\}, \Phi, \mathbb{N}^{(1)} \cup \{\epsilon^{(0)}\}, \langle q_1, q_2\rangle, R)$ は次を満たす:

$$\begin{aligned}\Phi = {} & \{(<0)\} \cup \{\mathit{abs} \circ \mathit{even}, \mathit{abs} \circ \neg\mathit{even}, \mathit{id} \circ \mathit{even}, \mathit{id} \circ \neg\mathit{even}, \hat{\epsilon} \circ \top\}\\ R = \{ \; & \langle q_1, q_2\rangle(((<0) \wedge \mathit{abs} \circ \mathit{even})(x)) \to \mathit{abs} \circ (\div 2)(\langle q_1, q_2\rangle(x))\\ & \langle q_1, q_2\rangle(((<0) \wedge \mathit{abs} \circ \neg\mathit{even})(x)) \to \mathit{abs} \circ \mathit{id}(\langle q_1, q_2\rangle(x))\\ & \langle q_1, q_2\rangle((\neg(<0) \wedge \mathit{id} \circ \mathit{even})(x)) \to \mathit{id} \circ (\div 2)(\langle q_1, q_2\rangle(x))\\ & \langle q_1, q_2\rangle((\neg(<0) \wedge \mathit{id} \circ \neg\mathit{even})(x)) \to \mathit{id} \circ \mathit{id}(\langle q_1, q_2\rangle(x))\\ & \langle q_1, q_2\rangle(\top^{(0)} \wedge \hat{\epsilon} \circ \top^{(0)}) \to \hat{\epsilon} \circ \hat{\epsilon} \qquad \}.\end{aligned}$$

各規則は記号的導出関係によって次のように導かれる:

- $(q_1((<0)(x)) \to abs(q_1(x))) \in R_1$ に対して，次の 2 つの正規形が得られる:

$$\begin{aligned}((<0), q_2(abs(q_1(x)))) &\overset{M_{(\div 2)}}{\Rightarrow}_S ((<0) \wedge abs \circ even, abs \circ (\div 2)(q_2(q_1(x))))\\ &\qquad (\because (q_2(even(x)) \to (\div 2)(q_2(x))) \in R_2)\\ &\overset{M_{(\div 2)}}{\Rightarrow}_S ((<0) \wedge abs \circ even, abs \circ (\div 2)(\langle q_1, q_2\rangle(x)))\\ ((<0), q_2(abs(q_1(x)))) &\overset{M_{(\div 2)}}{\Rightarrow}_S ((<0) \wedge abs \circ \neg even, abs \circ id(q_2(q_1(x))))\\ &\qquad (\because (q_2((\neg even)(x)) \to id(q_2(x))) \in R_2)\\ &\overset{M_{(\div 2)}}{\Rightarrow}_S ((<0) \wedge abs \circ \neg even, abs \circ id(\langle q_1, q_2\rangle(x)))\end{aligned}$$

  $[\![(<0) \wedge abs \circ even]\!] \neq \emptyset$, $[\![(<0) \wedge abs \circ \neg even]\!] \neq \emptyset$ である.
- $(q_1(\neg(<0)(x)) \to id(q_1(x))) \in R_1$ に対して，次の 2 つの正規形が得られる:

$$\begin{aligned}((<0), q_2(abs(q_1(x)))) &\overset{M_{(\div 2)}}{\Rightarrow}_S (\neg(<0) \wedge id \circ even, id \circ (\div 2)(q_2(q_1(x))))\\ &\qquad (\because (q_2(even(x)) \to (\div 2)(q_2(x))) \in R_2)\\ &\overset{M_{(\div 2)}}{\Rightarrow}_S (\neg(<0) \wedge id \circ even, id \circ (\div 2)(\langle q_1, q_2\rangle(x)))\\ ((<0), q_2(abs(q_1(x)))) &\overset{M_{(\div 2)}}{\Rightarrow}_S (\neg(<0) \wedge id \circ \neg even, id \circ id(q_2(q_1(x))))\\ &\qquad (\because (q_2((\neg even)(x)) \to id(q_2(x))) \in R_2)\\ &\overset{M_{(\div 2)}}{\Rightarrow}_S (\neg(<0) \wedge id \circ \neg even, id \circ id(\langle q_1, q_2\rangle(x)))\end{aligned}$$

  $[\![\neg(<0) \wedge id \circ even]\!] \neq \emptyset$, $[\![\neg(<0) \wedge id \circ \neg even]\!] \neq \emptyset$ である.
- $(q_1(\top^{(0)}) \to \hat{\epsilon}) \in R_1$ に対して，規則 $(q_2(\top^{(0)}) \to \hat{\epsilon}) \in R_2$ を使用して，1 つの正規形 $(\top, q_2(\hat{\epsilon})) \overset{M_{(\div 2)}}{\Rightarrow}_S (\top \wedge \hat{\epsilon} \circ \top, \hat{\epsilon} \circ \hat{\epsilon})$ が得られる．$[\![\top \wedge \hat{\epsilon} \circ \top]\!] \neq \emptyset$ である.

ランクが 1 のガードは互いに交わらず，かつ，それらを真にする文字の集合の和は整数全体の集合に一致する．従って $M_{abs} \fatsemi M_{(\div 2)}$ は STT である.

$M_{abs} \fatsemi M_{(\div 2)}$ はリストの各要素について，それが負数のときに絶対値を取ったら，即座にそれが偶数であるかを判断し，偶数ならば 2 で割った値に置き換える．よってリストの巡回は 1 度だけで済むので，$\mathbf{T}_{M_{abs}} \circ \mathbf{T}_{M_{(\div 2)}}(1(-2(-3(4(5(\varepsilon)))))) = 1(1(3(2(5(\epsilon))))))$ と同じ入力を与えると，$\mathbf{T}_{M_{abs}} \circ \mathbf{T}_{M_{(\div 2)}}$ よりステップの数が少ない次の簡約で計算を完了する:

$$\begin{aligned}&\langle q_1, q_2\rangle(1(-2(-3(4(5(\varepsilon)))))))\\ &\Rightarrow 1(\langle q_1, q_2\rangle(-2(-3(4(5(\varepsilon)))))))\\ &\Rightarrow 1(1(\langle q_1, q_2\rangle(-3(4(5(\varepsilon)))))))\\ &\Rightarrow 1(1(3(\langle q_1, q_2\rangle(4(5(\varepsilon)))))))\\ &\Rightarrow 1(1(3(2(\langle q_1, q_2\rangle(5(\varepsilon)))))))\\ &\Rightarrow 1(1(3(2(5(\langle q_1, q_2\rangle(\varepsilon)))))))\\ &\Rightarrow 1(1(3(2(5(\varepsilon))))))\end{aligned}$$

5　属性付き記号的木変換器

本章では，無限のランク付き文字集合を扱うように ATT を拡張した ASTT を導入する．ASTT における木の巡回や出力の仕方は ATT と同じで，ラベルの処理だけが異なる．従って，ASTT は ATT やその合成に関するいくつかの性質を受け継ぐ．

5.1　属性付き記号的木変換器の定義

まず，記号的木変換器の構文を定義する．規則の形は ATT に近い形を採用している．

定義 5.1 (属性付き記号的木変換器)．**属性付き記号的木変換器**（attributed symbolic tree transducer; ASTT）とは，以下を満たす 9 つ組 $M = (Syn, Inh, \Sigma, \Phi, \Delta, in, \sharp, \flat, R)$ である:

- Syn と Inh は $Syn \cap Inh = \emptyset$ を満たす有限集合であり，その要素をそれぞれ**合成属性**，**継承属性**と呼ぶ．また，$a \in Syn \cup Inh, i \in \mathbb{N} \setminus \{0\}$ のとき (a, π) や $(a, \pi i)$ をそれぞれ $a(\pi)$ や $a(\pi i)$ で表す．
- Σ と Δ は $(Syn \uplus Inh) \cap (\Sigma \cup \Delta) = \emptyset$ を満たす可算のランク付き文字集合であり，それぞれ**入力ランク付き文字集合**，**出力ランク付き文字集合**と呼ぶ．さらに (Σ, Φ) はラベル構造である．
- $in \in Syn$ を**初期属性**と呼ぶ．
- $\sharp^{(1)} \notin \Sigma \cup Syn \uplus Inh$ は文字であり，**初期記号**と呼ぶ．また Σ^+ で $\Sigma \uplus \{\sharp\}$ を表す．
- $\flat^{(1)} \notin \Phi$ は $[\![\flat]\!] = \{\sharp\}$ で定義される単項述語であり，**初期述語**と呼ぶ．また Φ^+ で $\Phi \uplus \{\flat\}$ を表し，$\mathrm{BC}^+(\Phi)$ で $\mathrm{BC}(\Phi) \uplus \{\flat\}$ を表す．
- $R \subseteq LHS(Syn, Inh, \mathrm{maxrk}(\Sigma)) \times \mathrm{BC}^+(\Phi) \times RHS(Syn, Inh, F(\Sigma^+ \to \Delta), \mathrm{maxrk}(\Sigma))$ は以下を満たす:
  - 任意の $l \in rk(\Sigma), a \in Syn$ に対して $(a(\pi), \varphi, \eta) \in R$ を満たす $\varphi^{(l)} \in \mathrm{BC}(\Phi), \eta \in RHS(Syn, Inh, F(\Sigma^{(l)} \to \Delta), l)$ が存在する．
  - 任意の $l \in rk(\Sigma), b \in Inh, i \in [l]$ に対して $(b(\pi i), \varphi, \eta) \in R$ を満たす $\varphi^{(l)} \in \mathrm{BC}(\Phi), \eta \in RHS(Syn, Inh, F(\Sigma^{(l)} \to \Delta), l)$ が存在する．
  - 任意の $b \in Inh, i \in [l]$ に対して $(b(\pi i), \flat, \eta) \in R$ を満たす $\eta \in RHS(Syn, Inh, F(\{\sharp\} \to \Delta), 1)$ が唯 1 つ存在する．
  - $(in(\pi), \flat, \eta) \in R$ を満たす $\eta \in RHS(Syn, Inh, F(\{\sharp\} \to \Delta), 1)$ が唯 1 つ存在する．

  なお，$\rho = (a(w), \varphi^{(l)}, \eta) \in R$ を**属性規則**と呼び，次の形で表す: $a(w) \overset{\varphi, l}{\to} \eta$．また，$(a(w), l)$，$\varphi$，$\eta$ をそれぞれ $\mathrm{lhs}(\rho)$，$\mathrm{grd}(\rho)$，$\mathrm{rhs}(\rho)$ で表し，それぞれを規則 ρ の**左辺**，**ガード**，**右辺**と呼ぶ．$R^{(l)}$ を $\{\rho \in R \mid \mathrm{rhs}(\rho) \in RHS^{(l)}\}$ で定義する．

- 任意の $\rho_1, \rho_2 \in R$ に対して $\mathrm{lhs}(\rho_1) = \mathrm{lhs}(\rho_2) \implies [\![\mathrm{grd}(\rho_1)]\!] \cap [\![\mathrm{grd}(\rho_2)]\!] = \emptyset$.
- 任意の $l \in rk(\Sigma^+), a \in Syn, b \in Inh, i \in [l]$ に対して

$$\bigcup_{\substack{\rho \in R \\ \mathrm{lhs}(\rho) = (a(\pi), l)}} [\![\mathrm{grd}(\rho)]\!] = \bigcup_{\substack{\rho \in R \\ \mathrm{lhs}(\rho) = (b(\pi i), l)}} [\![\mathrm{grd}(\rho)]\!] = \Sigma^+.$$

なお，上記の ASTT M に対して，

$$OCC(M) \stackrel{\mathrm{def}}{=} \bigcup_{l \in rk(\Sigma^+)} \{a(w) \in OCC(Syn, Inh, l) \mid \exists \rho \in R, w' \in path(\mathrm{rhs}(\rho)). \\ lab_{\mathrm{rhs}(\rho)}(w') = a(w)\}$$

$$LHS(M) \stackrel{\mathrm{def}}{=} \bigcup_{l \in rk(\Sigma^+)} \{(a(w), l) \in LHS(Syn, Inh, l) \times [l] \mid \exists \rho \in R. \mathrm{lhs}(\rho) = (a(w), l)\}$$

$$RHS(M) \stackrel{\mathrm{def}}{=} \bigcup_{l \in rk(\Sigma^+)} \{\eta \in RHS(Syn, Inh, F(\Sigma^+ \to \Delta), l) \mid \exists \rho \in R. \mathrm{rhs}(\rho) = \eta\}$$

と定義する．

ASTT の定義より，同じ左辺の規則のガードは互いに交わらず，かつそれらの和が入力ランク付き文字集合を網羅するので，明らかに次が成立する．

命題 5.2 (ASTT の決定性と全域性)**.** 任意の ASTT $M = (Syn, Inh, \Sigma, \Phi, \Delta, in, \sharp, \flat, R)$ は次を満たす: 任意の $\sigma \in \Sigma, (x, l) \in LHS(M)$ に対して，$(x, l) = \mathrm{lhs}(\rho), \sigma \in [\![\mathrm{grd}(\rho)]\!]$ を満たす $\rho \in R$ が唯 1 つ存在する．

例 5.3. ASTT $M_a = (\{a_1\}, \{b_1\}, \{\mathrm{bin}^{(2)}\} \cup \mathbb{Z}^{(0)}, \{(<0)^{(0)}\}, \mathbb{N}^{(1)} \cup \{\epsilon^{(0)}\}, a_1, \sharp_1, \flat_1, R_1)$ と ASTT $M_h = (\{a_2\}, \{b_2\}, \mathbb{N}^{(1)} \cup \{\epsilon^{(0)}\}, \{even^{(1)}\}, \mathbb{N} \cup \{\epsilon\}, a_2, \sharp_2, \flat_2, R_2)$ を次の R_1 と R_2 で定義する．

$$R_1 = \{\ a_1(\pi) \overset{\flat_1, 1}{\to} a_1(\pi 1) \\ a_1(\pi) \overset{\top, 2}{\to} a_1(\pi 1) \\ a_1(\pi) \overset{(<0), 0}{\to} abs(b_1(\pi)) \\ a_1(\pi) \overset{\neg(<0), 0}{\to} id(b_1(\pi)) \\ b_1(\pi 1) \overset{\flat_1, 1}{\to} \hat{\epsilon} \\ b_1(\pi 1) \overset{\top, 2}{\to} a_1(\pi 2) \\ b_1(\pi 2) \overset{\top, 2}{\to} b_1(\pi) \qquad \}$$

$$R_2 = \{\ a_2(\pi) \overset{\flat_2, 1}{\to} a_2(\pi 1) \\ a_2(\pi) \overset{\top, 1}{\to} a_2(\pi 1) \\ a_2(\pi) \overset{\top, 0}{\to} b_2(\pi) \\ b_2(\pi 1) \overset{\flat_2, 1}{\to} \hat{\epsilon} \\ b_2(\pi 1) \overset{even, 1}{\to} (\div 2)(b_2(\pi)) \\ b_2(\pi 1) \overset{\neg even, 1}{\to} id(b_2(\pi)) \quad \}$$

ただし，(<0) は負の整数で真になる述語，$even$ は偶数で真になる述語，abs は整数の絶対値を取る関数，id は恒等関数，$(\div 2)$ は自然数を 2 で割る関数である．

5.2 属性付き記号的木変換器による木変換

次に，ASTT の意味論を定義する．

定義 5.4 (ASTT による導出関係)**.** $M = (Syn, Inh, \Sigma, \Phi, \Delta, in, \sharp, \flat, R)$ を ASTT として，$\xi \in T_{\Sigma^+}$ とする．このとき,ξ における M による**導出関係**を,$T_\Sigma(\{a(w) \mid a \in Syn \cup Inh, w \in path(\xi)\})$ 上の最小の二項関係 $\overset{M,\xi}{\Rightarrow}$ で定義する:

- $a(w) \overset{M,\xi}{\Rightarrow} \eta'[\pi := w] \iff a \in Syn, (a(\pi) \overset{\varphi,l}{\to} \eta) \in R, lab_\xi(w) \in [\![\varphi]\!], \eta' = give_\eta(lab_\xi(w))$
- $b(wi) \overset{M,\xi}{\Rightarrow} \eta'[\pi := w] \iff b \in Inh, (b(\pi i) \overset{\varphi,l}{\to} \eta) \in R, lab_\xi(w) \in [\![\varphi]\!], \eta' = give_\eta(lab_\xi(w))$
- $\delta(\eta_1, \ldots, \eta_i, \ldots, \eta_l) \overset{M,\xi}{\Rightarrow} \delta(\eta_1, \ldots, \eta'_i, \ldots, \eta_l) \iff \delta \in \Delta, \eta_i \overset{M,\xi}{\Rightarrow} \eta'_i, \forall j \in [i-1].\, \eta_j \in T_\Delta$

ただし，導出関係 $\overset{M,\xi}{\Rightarrow}$ において M や ξ が明らかなときは，それを省略して $\Rightarrow$ と書く．

命題 5.2 より，初めの 2 つの項目で選ばれる規則は一意である．さらに 3 つ目の項目より，最外最左簡約を用いるため，この導出関係は決定的である．従って，この導出関係による簡約系では，正規形が存在するとき，それは一意である．ただし，導出が循環して停止しないときは正規形が存在しないため，以降の ASTT には次の定義可能性を仮定する．

定義 5.5. $M = (Syn, Inh, \Sigma, \Phi, \Delta, in, \sharp, \flat, R)$ を ASTT とすると，M による**木変換**とは，次で定義する関数 $\mathbf{T}_M : T_\Sigma \to T_\Delta$ である: 任意の $\xi \in T_\Sigma$ に対して $\mathbf{T}_M(\xi) = nf(\overset{M,\sharp(\xi)}{\Rightarrow}, in(\varepsilon))$.

これが定義可能（well-defined）であるためには，次が成立すれば十分である: 任意の $\xi \in T_\Sigma, a \in Syn \cup Inh, w \in path(\sharp(\xi))$ に対して，$\zeta = nf(\overset{M,\sharp(\xi)}{\Rightarrow}, a(w)), (a, w) \notin Inh \times \{\varepsilon\}$ を満たす $\zeta \in T_\Delta$ が存在する．

これが成立するとき M は**定義可能**（*well-defined*）であるという．

以降の ASTT は指定のない限り定義可能とする．全ての定義可能な ASTT による木変換の集合を $ASTT$ で表す．また，継承属性を持たない ATT が TDTT であるのと同じ理由で，継承属性を持たない ASTT は STT と等価である．

例 5.6. 木 $\xi = \mathrm{bin}(\mathrm{bin}(1,-2), \mathrm{bin}(\mathrm{bin}(-3,4),5)) \in T_{\mathbb{Z}\cup\{\mathrm{bin}\}}$ に対して，$\mathbf{T}_{M_a}(\xi) = nf(\overset{M_a,\sharp_1(\xi)}{\Rightarrow}, a_1(\varepsilon)) = 1(2(3(4(5(\epsilon))))) \in T_{\mathbb{N}\cup\{\epsilon\}}$ であり，これは図 5.1(a) の簡約で計算される．さらにこの木を ζ とすると，$\mathbf{T}_{M_h}(\zeta) = nf(\overset{M_h,\sharp_2(\zeta)}{\Rightarrow}, a_2(\varepsilon)) = 5(2(3(1(1(\epsilon))))) \in T_{\mathbb{N}\cup\{\epsilon\}}$ であり，これは図 5.1(b) の簡約で計算される．従って $\mathbf{T}_{M_a} \circ \mathbf{T}_{M_h}(\mathrm{bin}(\mathrm{bin}(1,-2), \mathrm{bin}(\mathrm{bin}(-3,4),5))) = 5(2(3(1(1(\epsilon)))))$ である．

$$
\begin{aligned}
a_1(\varepsilon) &\Rightarrow a_1(1)\\
&\Rightarrow a_1(11)\\
&\Rightarrow a_1(111)\\
&\Rightarrow 1(b_1(111))\\
&\Rightarrow 1(a_1(112))\\
&\Rightarrow 1(2(b_1(112)))\\
&\Rightarrow 1(2(b_1(11)))\\
&\Rightarrow 1(2(a_1(12)))\\
&\Rightarrow 1(2(a_1(121)))\\
&\Rightarrow 1(2(a_1(1211)))\\
&\Rightarrow 1(2(3(b_1(1211))))\\
&\Rightarrow 1(2(3(a_1(1212))))\\
&\Rightarrow 1(2(3(4(b_1(1212)))))\\
&\Rightarrow 1(2(3(4(b_1(121)))))\\
&\Rightarrow 1(2(3(4(a_1(122)))))\\
&\Rightarrow 1(2(3(4(5(b_1(122))))))\\
&\Rightarrow 1(2(3(4(5(b_1(12))))))\\
&\Rightarrow 1(2(3(4(5(b_1(1))))))\\
&\Rightarrow 1(2(3(4(5(\epsilon)))))
\end{aligned}
$$

(a) M_a による簡約例

$$
\begin{aligned}
a_2(\varepsilon) &\Rightarrow a_2(1)\\
&\Rightarrow^5 a_2(111111)\\
&\Rightarrow b_2(111111)\\
&\Rightarrow 5(b_2(11111))\\
&\Rightarrow 5(4(b_2(1111)))\\
&\Rightarrow 5(2(3(b_2(111))))\\
&\Rightarrow 5(2(3(1(b_2(11)))))\\
&\Rightarrow 5(2(3(1(1(b_2(1)))))))\\
&\Rightarrow 5(2(3(1(1(\epsilon)))))
\end{aligned}
$$

(b) M_h による簡約例

図 5.1: ASTT による簡約例

5.3 属性付き記号的木変換器が表現する計算のクラス

ASTT は STT を自然に拡張した木変換器であるから，ASTT と STT の関係は ATT と TDTT の関係と同様に捉えることができる．命題 4.8 で STT の樹高特性を示したので，$TDTT \subsetneq ATT$ であることと同じ理由で，次が成立する．

命題 5.7 (ASTT が表現する計算のクラス)**.** $STT \subsetneq ASTT$.

証明. 任意の STT $M = (Q, \Sigma, \Phi, \Delta, q_0, R)$ に対して，ASTT $M' = (Q, \emptyset, \Sigma, \Phi, \Delta, q_0, \sharp, \flat, R')$ を，次の規則を含む R' で定義する:

- $q_0(\pi) \xrightarrow{\flat,1} q_0(\pi 1)$.

- 任意の $\rho \in R$ に対して $q(\pi) \overset{\mathrm{grd}(\rho),l}{\to} \mathrm{rhs}(\rho)[x_i := \pi i]_{i\in[l]}$．ただし $(x,l) = \mathrm{lhs}(\rho)$.

このように，継承属性を持たない ASTT で STT を表現することができるので，$STT \subseteq ASTT$.

次に，STT で表現できない ASTT による木変換の存在を示す．まず，葉節点が自然数でラベル付けされた二分木を表すランク付き文字集合 $\Delta_{\mathrm{bin}} = \{\mathrm{bin}^{(2)}\} \cup \mathbb{N}^{(0)}$ に対して，高さが n の平衡二分木の集合 $T_n \in T_\Delta$ を次で定義する:

- $T_1 = \mathbb{N}$.
- 任意の $n \geq 1$ と $\xi \in T_n$ に対して $T_{n+1} = \mathrm{bin}(\xi, \xi)$.

このとき，ASTT $M_{\mathrm{bin}} = (\{a\}, \{b\}, \Delta_{\mathrm{bin}}, \emptyset, \Delta_{\mathrm{bin}}, a, \sharp, \flat, R)$ を，次の規則を含む R で定義する:

$$
\begin{aligned}
&a(\pi) \overset{\flat,1}{\to} a(\pi 1) && b(\pi 1) \overset{\flat,1}{\to} \hat{0}\\
&a(\pi) \overset{\top,2}{\to} \hat{\mathrm{bin}}(a(\pi 1), a(\pi 1)) && b(\pi 1) \overset{\top,2}{\to} a(\pi 2)\\
&a(\pi) \overset{\top,0}{\to} \hat{\mathrm{bin}}(b(\pi), b(\pi)) && b(\pi 2) \overset{\top,2}{\to} b(\pi)
\end{aligned}
$$

次に，任意の高さ n に対して，$\mathbf{T}_{M_{\mathrm{bin}}}(T_n) = T_{2^n}$ を，n に関する帰納法で示す.

$n = 1$ のとき　任意の $\xi \in T_1 = \mathbb{N}$ に対して，

$$
\begin{aligned}
\mathbf{T}_{M_{\mathrm{bin}}}(\xi) &= nf(\overset{M_{\mathrm{bin}},\sharp(\xi)}{\Rightarrow}, a(\varepsilon))\\
&= nf(\overset{M_{\mathrm{bin}},\sharp(\xi)}{\Rightarrow}, a(1))\\
&= nf(\overset{M_{\mathrm{bin}},\sharp(\xi)}{\Rightarrow}, \mathrm{bin}(b(1), b(1)))\\
&= nf(\overset{M_{\mathrm{bin}},\sharp(\xi)}{\Rightarrow}, \mathrm{bin}(0,0))\\
&= \mathrm{bin}(0,0) \in T_{2^1}.
\end{aligned}
$$

$n = k \geq 1$ のとき　任意の $\xi = \mathrm{bin}(\xi_1, \xi_2) \in T_{k+1}$ に対して，ξ_1，$\xi_2 \in T_k$ に注意して，

$$
\begin{aligned}
\mathbf{T}_{M_{\mathrm{bin}}}(\xi) &= nf(\overset{M_{\mathrm{bin}},\sharp(\mathrm{bin}(\xi_1,\xi_2))}{\Rightarrow}, a(\varepsilon))\\
&= nf(\overset{M_{\mathrm{bin}},\sharp(\mathrm{bin}(\xi_1,\xi_2))}{\Rightarrow}, a(1))\\
&= nf(\overset{M_{\mathrm{bin}},\sharp(\mathrm{bin}(\xi_1,\xi_2))}{\Rightarrow}, \mathrm{bin}(a(11), a(11)))\\
&= nf(\overset{M_{\mathrm{bin}},\xi_2}{\Rightarrow}, nf(\overset{M_{\mathrm{bin}},\xi_1}{\Rightarrow}, \mathrm{bin}(a(\varepsilon), a(\varepsilon)))[b(\varepsilon) := a(\varepsilon)])[b(\varepsilon) := 0]\\
&= \mathrm{bin}(nf(\overset{M_{\mathrm{bin}},\xi_2}{\Rightarrow}, nf(\overset{M_{\mathrm{bin}},\xi_1}{\Rightarrow}, a(\varepsilon))[b(\varepsilon) := a(\varepsilon)])[b(\varepsilon) := 0],\\
&\qquad nf(\overset{M_{\mathrm{bin}},\xi_2}{\Rightarrow}, nf(\overset{M_{\mathrm{bin}},\xi_1}{\Rightarrow}, a(\varepsilon))[b(\varepsilon) := a(\varepsilon)])[b(\varepsilon) := 0])
\end{aligned}
$$

ここで，$\eta_1 = nf(\overset{M_{\mathrm{bin}},\xi_1}{\Rightarrow}, a(\varepsilon))$，$\eta_2 = nf(\overset{M_{\mathrm{bin}},\xi_2}{\Rightarrow}, \eta_1[b(\varepsilon) := a(\varepsilon)])$ とおく．帰納法の仮定より

$$
\eta_1[b(\varepsilon) := 0] = nf(\overset{M_{\mathrm{bin}},\sharp(\xi_1)}{\Rightarrow}, a(\varepsilon)) = \mathbf{T}_{M_{\mathrm{bin}}}(\xi_1) \in T_{2^k}
$$

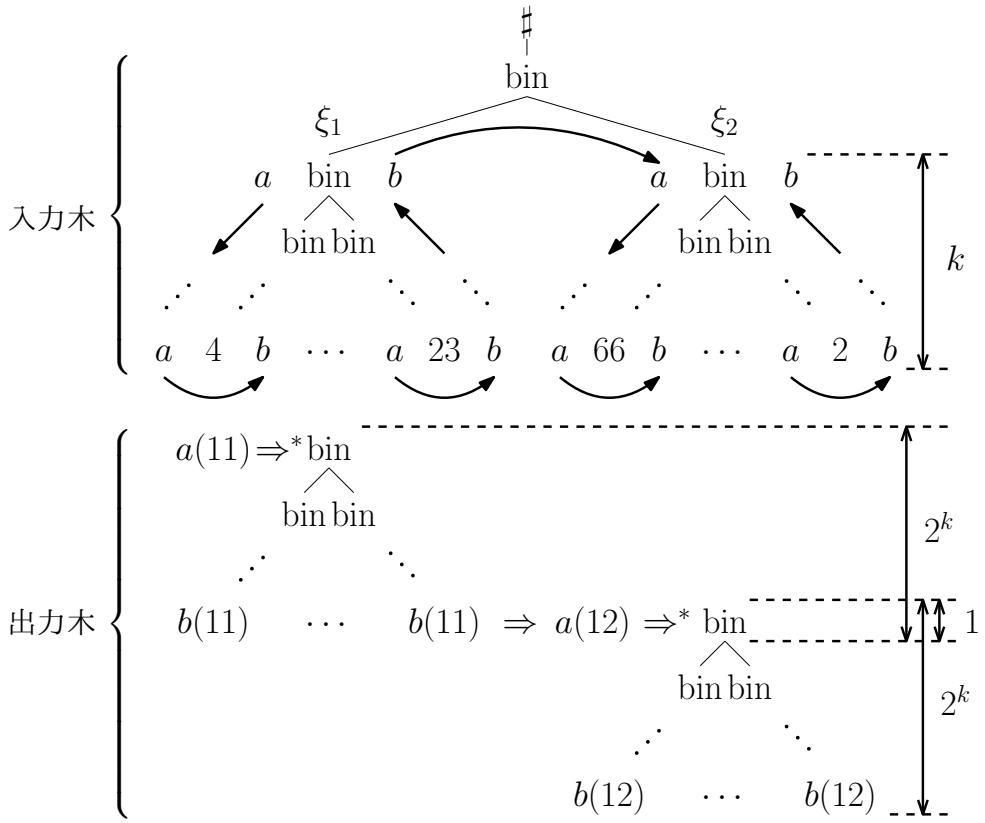


図 5.2: M_{bin} の計算

$$nf(\overset{M_{\mathrm{bin}},\xi_2}{\Rightarrow}, a(\varepsilon))[b(\varepsilon) := 0] = nf(\overset{M_{\mathrm{bin}},\sharp(\xi_2)}{\Rightarrow}, a(\varepsilon)) = \mathbf{T}_{M_{\mathrm{bin}}}(\xi_2) \in T_{2^k}$$

である．図 5.2より $\eta_2[b(\varepsilon) := 0] \in T_{2^k+2^k-1}$ であるから，

$$\begin{aligned} ht(\mathbf{T}_{M_{\mathrm{bin}}}(\xi)) &= ht(\mathbf{T}_{M_{\mathrm{bin}}}(\mathrm{bin}(\eta_2[b(\varepsilon) := 0], \eta_2[b(\varepsilon) := 0]))) \\ &= 1 + \max\{2^k + 2^k - 1, 2^k + 2^k - 1\} = 2^{k+1}. \end{aligned}$$

以上より，任意の $c \in \mathbb{N}$ に対して，$ht(\mathbf{T}_M(\xi)) = 2^{c+1} > c \cdot ht(\xi) = c \cdot (c+1)$ を満たす $\xi \in T_{c+1}$ が存在する．従って命題 4.8 より $\mathbf{T}_M \notin STT$．つまり $STT \subsetneq ASTT$ が成立する． □

上の証明では，M_{bin} は ASTT で表現できるが，STT で表現できないことを示している．ATT や ASTT は継承属性を持つため，2 分木の中間節点において，一方の部分木の巡回による出力木に対して，他方の部分木に対する巡回による出力木を追加することができる．従って，平衡二分木では中間節点の数だけ出力木の高さを増やすことができる．

6　属性付き記号的木変換器の合成

本章では，ASTT の合成アルゴリズムを導入し，その正当性を証明する．まず，ATT の SUR を ASTT に適応させ，記号的導出関係を使って DC を拡張する．正当性の証明では，合成した ASTT を正規化し，それを基にして合成前の ASTT を ATT に符号化する．最後にその ATT の入力文字や出力文字を，特定の入力木の有限個のラベルを基にして置き換えることで，元の ASTT と同じ計算を行うように構成する．これによって，ASTT の合成の正当性を ATT の合成の正当性に帰着する．

述語 θ や関数 f，像 $f([\![\theta]\!])$ をそのまま文字として扱うときは，区別のため，それぞれ $\overline{\theta}$ や $\overline{f}$，$\overline{f([\![\theta]\!])}$ で表す．

6.1　属性付き記号的木変換器の単一使用制約

まず，単一使用制約（SUR）を ASTT に適応させる．

定義 6.1 (ASTT の SUR)**.** ASTT $M = (Syn, Inh, \Sigma, \Phi, \Delta, in, \sharp, \flat, R)$ に対して，次が成立するとき，M は **単一使用制約**（*single-use requirement*, SUR）を満たすという: 任意の $a(w) \in OCC(M)$ に対して，

$$\forall w_1 \in path(\eta_1), w_2 \in path(\eta_2).\, [\![\varphi_1]\!] \cap [\![\varphi_2]\!] \neq \emptyset,$$
$$(x_1, w_1) \neq (x_2, w_2), sub_{\eta_1}(w_1) = sub_{\eta_2}(w_2) = a(w)$$

を満たす $(x_1 \overset{\varphi_1;^l}{\to} \eta_1), (x_2 \overset{\varphi_2;^l}{\to} \eta_2) \in R$ が存在しない．

なお，SUR を満たす全ての定義可能な ASTT による木変換の集合を $ASTT_{su}$ で表す．

ATT の SUR では，同じ文字に対する規則の中で同じ属性の適用が 2 度以上出現しないことを述べている．ASTT の SUR は「同じ文字に対する規則」を「同じ文字が満たすガードを持つ規則」と言い換えているだけなので，ATT の SUR と等価である．つまり SUR を満たす ASTT について，任意の入力文字 σ に対応する全ての規則を集めると，その右辺全体で，同じ属性の出現が丁度 1 回である．

6.2　属性付き記号的木変換器の記述的合成

次に，DC に記号的導出関係を取り入れることで，ASTT の合成アルゴリズムを導入する．2 つの ASTT M_1，M_2 を構文的に合成して得られる 1 つの ASTT を $M_1 \fatsemi M_2$ で表す．

定義 6.2 (ASTT の記号的導出関係)**.** 任意の ASTT M_1 と ASTT M_2 に対して，

$$M_1 = (Syn_1, Inh_1, \Sigma_1, \Phi_1, \Delta_1, in_1, \sharp_1, \flat_1, R_1)$$
$$M_2 = (Syn_2, Inh_2, \Sigma_2, \Phi_2, \Delta_2, in_2, \sharp_2, \flat_2, R_2)$$

かつ $\Delta_1 \subseteq \Sigma_2$ のとき，木 $\eta \in RHS(M_1)$ における M_2 による**記号的導出関係**を，$\mathrm{Pred}_0(\Sigma_1) \times T_{F(\Sigma_1 \to \Delta_2)}(OCC(M_1) \cup OCC(Syn_1 \times Syn_2, Syn_1 \times Inh_2, \mathrm{maxrk}(\Delta_2)))$ 上の最小の二項関係 $\overset{M_2,\eta}{\Rightarrow}_S$ で定義する:

- $(\theta, a_2(w)) \overset{M_2,\eta}{\Rightarrow}_S (\theta \wedge lab_\eta(w)^{(l)} \circ \varphi_2^{(l)}, \zeta'[\pi := w])$
  $\iff \theta \in \mathrm{Pred}_0(\Sigma_1), a_2 \in Syn_2, a_2(\pi) \overset{\varphi_2,l}{\Rightarrow} \zeta \in R_2, \zeta' = comp_\zeta(lab_\eta(w))$
- $(\theta, b_2(wj)) \overset{M_2,\eta}{\Rightarrow}_S (\theta \wedge lab_\eta(w)^{(l)} \circ \varphi_2^{(l)}, \zeta'[\pi := w])$
  $\iff \theta \in \mathrm{Pred}_0(\Sigma_1), b_2 \in Inh_2, b_2(\pi j) \overset{\varphi_2,l}{\Rightarrow} \zeta \in R_2, \zeta' = comp_\zeta(lab_\eta(w))$
- $(\theta, f(\zeta_1, \ldots, \zeta_j, \ldots, \zeta_l)) \overset{M_2,\eta}{\Rightarrow}_S (\theta', f(\zeta_1, \ldots, \zeta'_j, \ldots, \zeta_l))$
  $\iff f \in F(\Sigma_1 \to \Delta_2), (\theta, \zeta_j) \overset{M_2,\eta}{\Rightarrow}_S (\theta', \zeta'_j), \forall j \in [i-1].\, \eta_j \in T_{F(\Sigma_1 \to \Delta_2)}$
- $(\theta, a_2(w)) \overset{M_2,\eta}{\Rightarrow}_S (\theta, \langle a_1, a_2 \rangle(\pi j))$
  $\iff \theta \in \mathrm{Pred}_0(\Sigma_1), a_1 \in Syn_1, a_2 \in Syn_2, j \in [\mathrm{maxrk}(\Delta_1)], lab_\eta(w) = a_1(\pi j)$
- $(\theta, a_2(w)) \overset{M_2,\eta}{\Rightarrow}_S (\theta, \langle b_1, a_2 \rangle(\pi))$
  $\iff \theta \in \mathrm{Pred}_0(\Sigma_1), b_1 \in Inh_1, a_2 \in Syn_2, j \in [\mathrm{maxrk}(\Delta_1)], lab_\eta(w) = b_1(\pi)$

この導出関係は，ATT の合成（定義 3.8）で M'_2 が M_1 の規則の右辺を簡約するという手続きを，STT の記号的導出関係と統合して ASTT に適応させたものである．初めの 2 つの項目では，適用した属性と，その節点のランクに応じて規則を選択し，その右辺 ζ を簡約のために用いる．しかし，ASTT の規則の右辺のラベルは関数であるため，選択される規則は一意ではない．つまり選択する規則のガードに指定がないので，$(\theta, a_2(w))$ や $(\theta, b_2(w))$ の正規形は一意ではない．また，ζ に対しては，経路の変数を置換するだけでなく，ζ の各ラベル（関数）を，属性を適用した節点のラベル（関数）と合成する．これによって，M_1 の入力ラベルは直接 M_2 の出力ラベルに写される．そして述語に関しても，選択した規則のガードを，属性を適用した節点のラベルと合成し，さらに簡約前の述語との積を取る．従って，簡約で得られる正規形を (θ, ζ) とすると，それまでの簡約で選択されてきた全ての M_2 の規則のガードが θ に蓄えられている．つまり，この簡約系で導出された木を，$M_1 \fatsemi M_2$ の規則の右辺として用いるとき，その規則が適用されるべきラベルは全て，この述語を真にする．

3 つ目の項目より，η_i より左側の全て木は属性を含まない簡約済みの木である．よってこの導出関係による簡約は最外最差簡約である．また，最後の 2 つの項目では簡約前の $a_2(w)$ に対して簡約が一意である．従って正規形は，初めの 2 つの項目でのガードの選び方の数だけ存在する．なお，最後の 2 つの項目では，簡約によって述語は変化しておらず，木の簡約は ATT の合成で R_2 に導入した規則による簡約と等価である．

次に，記号的導出関係を用いて DC を拡張する．

定義 6.3 (ASTT の DC)**.** 任意の ASTT M_1 と ASTT M_2 に対して，

$$M_1 = (Syn_1, Inh_1, \Sigma_1, \Phi_1, \Delta_1, in_1, \sharp_1, \flat_1, R_1)$$
$$M_2 = (Syn_2, Inh_2, \Sigma_2, \Phi_2, \Delta_2, in_2, \sharp_2, \flat_2, R_2)$$

かつ，$\Delta_1 \subseteq \Sigma_2$ であり，$Inh_2 = \emptyset$ または M_1 が SUR を満たすとき，$M_1 \mathbin{⨟} M_2$ を，次を満たす $(Syn, Inh, \Sigma_1, \Phi, \Delta_2, \langle in_1, in_2\rangle, \sharp_1, \flat_1, R)$ で定義する:

$$\begin{aligned}
Syn &= (Syn_1 \times Syn_2) \cup (Inh_1 \times Inh_2)\\
Inh &= (Syn_1 \times Inh_2) \cup (Inh_1 \times Syn_2)\\
\Phi = \Phi_1 \cup \{ f_1 \circ \varphi_2 \mid & f_1^{(l)} \in F(\Sigma_1 \to \Delta_1), \varphi_2^{(l)} \in \mathrm{BC}(\Phi_2),\\
& \exists \rho_1 \in R_1, w \in path(\mathrm{rhs}(\rho_1)).\, lab_{\mathrm{rhs}(\rho_1)}(w) = f_1,\\
& \exists \rho_2 \in R_2.\, \mathrm{grd}(\rho_2) = \varphi_2\}.
\end{aligned}$$

ただし，R は次を満たす:

- 任意の $l \in rk(\Sigma_1)$ に対して，$R^{(l)}$ は次の属性規則を含む:
  - 任意の $a_1 \in Syn_1, a_1(\pi) \overset{\varphi,l}{\to} \eta \in R_1$ に対して，$\varphi \neq \flat_1$ のとき:
    - 任意の $a_2 \in Syn_2$ と正規形 $(\varphi, a_2(\varepsilon))$ $(\overset{M_2,\eta}{\Rightarrow}_S)^*$ (θ, ζ) に対して，$[\![\theta]\!] \neq \emptyset$ ならば，

$$\langle a_1, a_2\rangle(\pi) \overset{\theta,l}{\to} \zeta[b_2'(\varepsilon) := \langle a_1, b_2'\rangle(\pi)]_{b_2' \in Inh_2}.$$

    - 任意の $b_2 \in Inh_2, w \in path(\eta), j' \in [l], a_1'(\pi j') = sub_\eta(w), a_1' \in Syn_1$ と正規形 $(\varphi, b_2(w))$ $(\overset{M_2,\eta}{\Rightarrow}_S)^*$ (θ, ζ) に対して，$[\![\theta]\!] \neq \emptyset$ ならば，

$$\langle a_1', b_2\rangle(\pi j') \overset{\theta,l}{\to} \zeta[b_2'(\varepsilon) := \langle a_1, b_2'\rangle(\pi)]_{b_2' \in Inh_2}.$$

    - 任意の $b_2 \in Inh_2, w \in path(\eta), b_1'(\pi) = sub_\eta(w), b_1' \in Inh_1$ と正規形 $(\varphi, b_2(w))$ $(\overset{M_2,\eta}{\Rightarrow}_S)^*$ (θ, ζ) に対して，$[\![\theta]\!] \neq \emptyset$ ならば，

$$\langle b_1', b_2\rangle(\pi) \overset{\theta,l}{\to} \zeta[b_2'(\varepsilon) := \langle a_1, b_2'\rangle(\pi)]_{b_2' \in Inh_2}.$$

  - 任意の $b_1 \in Inh_1, j \in [l], b_1(\pi j) \overset{l,\varphi}{\to} \eta \in R_1$ に対して，$\varphi \neq \flat_1$ のとき:
    - 任意の $a_2 \in Syn_2$ と正規形 $(\varphi, a_2(\varepsilon))$ $(\overset{M_2,\eta}{\Rightarrow}_S)^*$ (θ, ζ) に対して，$[\![\theta]\!] \neq \emptyset$ ならば，

$$\langle b_1, a_2\rangle(\pi j) \overset{\theta,l}{\to} \zeta[b_2'(\varepsilon) := \langle b_1, b_2'\rangle(\pi j)]_{b_2' \in Inh_2}.$$

    - 任意の $b_2 \in Inh_2, w \in path(\eta), j' \in [l]$，$a_1'(\pi j') = sub_\eta(w), a_1' \in Syn_1$ と正規形 $(\varphi, b_2(w))$ $(\overset{M_2,\eta}{\Rightarrow}_S)^*$ (θ, ζ) に対して，$[\![\theta]\!] \neq \emptyset$ ならば，

$$\langle a_1', b_2\rangle(\pi j') \overset{\theta,l}{\to} \zeta[b_2'(\varepsilon) := \langle b_1, b_2'\rangle(\pi j)]_{b_2' \in Inh_2}.$$

* 任意の $b_2 \in Inh_2, w \in path(\eta)$, $b_1'(\pi) = sub_\eta(w), b_1' \in Inh_1$ と正規形 $(\varphi, b_2(w))$ $(\overset{M_2,\eta}{\Rightarrow}_S)^*$ (θ, ζ) に対して，$[\![\theta]\!] \neq \emptyset$ ならば，

$$\langle b_1', b_2\rangle(\pi) \overset{\theta,l}{\to} \zeta[b_2'(\varepsilon) := \langle b_1, b_2'\rangle(\pi j)]_{b_2' \in Inh_2}.$$

- さらに $R^{(1)}$ は次の規則を含む．
  - $in_1(\pi) \overset{\flat_1,1}{\to} \eta \in R_1$ に対して:
    * $(\flat_1, in_2(\varepsilon))$ $(\overset{M_2,\hat{\sharp_1}(\eta)}{\Rightarrow}_S)^*$ (θ, ζ) が正規形であり，$[\![\theta]\!] \neq \emptyset$ ならば，

$$\langle in_1, in_2\rangle(\pi) \overset{\flat_1,1}{\to} \zeta$$

    * 任意の $w \in path(\eta), a_1' \in Syn_1, b_2 \in Inh_2, a_1'(\pi 1) = sub_\eta(w)$ と正規形 $(\flat_1, b_2(1w))$ $(\overset{M_2,\hat{\sharp_1}(\eta)}{\Rightarrow}_S)^*$ (θ, ζ) に対して，$[\![\theta]\!] \neq \emptyset$ ならば，

$$\langle a_1', b_2\rangle(\pi 1) \overset{\flat_1,1}{\to} \zeta.$$

  - 任意の $b_1 \in Inh_1$ と $b_1(\pi 1) \overset{\flat_1,1}{\to} \eta \in R_1$ に対して:
    * 任意の $a_2 \in Syn_2$ と正規形 $(\flat_1, a_2(\varepsilon))$ $(\overset{M_2,\eta}{\Rightarrow}_S)^*$ (θ, ζ) に対して $[\![\theta]\!] \neq \emptyset$ ならば，

$$\langle b_1, a_2\rangle(\pi 1) \overset{\flat_1,1}{\to} \zeta[b_2(\varepsilon) := \langle b_1, b_2'\rangle(\pi 1)]_{b_2' \in Inh_2}.$$

    * 任意の $w \in path(\eta), a_1' \in Syn_1, b_2 \in Inh_2, a_1'(\pi 1) = sub_\eta(w)$ と正規形 $(\flat_1, b_2(w))$ $(\overset{M_2,\eta}{\Rightarrow}_S)^*$ (θ, ζ) に対して，$[\![\theta]\!] \neq \emptyset$ のとき，

$$\langle a_1', b_2\rangle(\pi 1) \overset{\flat_1,1}{\to} \zeta[b_2'(\varepsilon) := \langle b_1, b_2'\rangle(\pi 1)]_{b_2' \in Inh_2}$$

M_1 の右辺の簡約に記号的導出関係を用いていることを除いては，ATT の合成と同様である．述語の簡約は M_1 の規則のガードから始めており，簡約して得られた述語を満たすラベルが存在しない場合は，それをガードとする規則は無駄であるから，$M_1 \fatsemi M_2$ の規則に含めない．

合成の定義より，明らかに $M_1 \fatsemi M_2$ は ASTT の構文で表現でき，さらに次の命題が成立するので，$M_1 \fatsemi M_2$ は ASTT である．

命題 6.4. ASTT M_1 と ASTT M_2 に対して，

$$M_1 \fatsemi M_2 = (Syn, Inh, \Sigma_1, \Phi, \Delta_2, \langle in_1, in_2\rangle, \sharp_1, \flat_1, R)$$

が存在するとき，次が成立する:

- 任意の $\rho_1, \rho_2 \in R$ に対して，$\mathrm{lhs}(\rho_1) = \mathrm{lhs}(\rho_2)$ ならば $[\![\mathrm{grd}(\rho_1)]\!] \cap [\![\mathrm{grd}(\rho_2)]\!] = \emptyset$.
- 任意の $l \in rk(\Sigma_1), a \in Syn, b \in Inh, i \in [l]$ に対して

$$\bigcup_{\substack{\rho \in R \\ \mathrm{lhs}(\rho) = (a(\pi), l)}} [\![\mathrm{grd}(\rho)]\!] = \bigcup_{\substack{\rho \in R \\ \mathrm{lhs}(\rho) = (b(\pi i), l)}} [\![\mathrm{grd}(\rho)]\!] = \Sigma_1.$$

証明.

$$M_1 = (Syn_1, Inh_1, \Sigma_1, \Phi_1, \Delta_1, in_1, \sharp_1, \flat_1, R_1)$$
$$M_2 = (Syn_2, Inh_2, \Sigma_2, \Phi_2, \Delta_2, in_2, \sharp_2, \flat_2, R_2)$$

とする．合成の定義より，M_2 によるガードの簡約は M_1 の 1 つの規則のガードから始めていて，かつ，$Inh_2 = \emptyset$ または M_1 が SUR を満たす．M_1 は ASTT だから，任意の $(\langle a_1, a_2\rangle(w) \overset{\theta,l}{\to} \eta) \in R$ に対して，$[\![\theta]\!] \subseteq [\![\varphi_1]\!]$ を満たす $(a_1(w) \overset{\varphi_1,l}{\to} \eta_1) \in R_1$ が唯 1 つ存在する．従って，任意の $(a_1(w) \overset{\varphi_1,l}{\to} \eta_1) \in R_1$ に対して次の 2 つを示せばよい．

1. 任意の $(\langle a_1, a_2\rangle(w) \overset{\theta_1,l}{\to} \zeta_1), (\langle a_1, a_2\rangle(w) \overset{\theta_2,l}{\to} \zeta_2) \in R$ に対して $[\![\theta_1]\!] \subseteq [\![\varphi_1]\!], [\![\theta_2]\!] \subseteq [\![\varphi_1]\!]$ ならば $[\![\theta_1]\!] \cap [\![\theta_2]\!] = \emptyset$.
2. 任意の $(\langle a_1, a_2\rangle(w), l) \in LHS(M_1 \fatsemi M_2)$ に対して $\bigcup_{\substack{(\langle a_1,a_2\rangle(w)\overset{\theta,l}{\to}\zeta)\in R \\ [\![\theta]\!]\subseteq[\![\varphi_1]\!]}} [\![\theta]\!] = [\![\varphi_1]\!]$.

1 の証明　記号的導出関係では，M_2 の規則のガードを蓄えている．任意の異なる $(\langle a_1, a_2\rangle(w) \overset{\theta,l}{\to} \zeta), (\langle a_1, a_2\rangle(w) \overset{\theta',l}{\to} \zeta') \in R$ に対して，$(\varphi_1, a_2(w'))\ (\overset{M_2,\eta_1}{\Rightarrow}_S)^*\ (\theta, \zeta), (\varphi_1, a_2(w'))\ (\overset{M_2,\eta_1}{\Rightarrow}_S)^*\ (\theta', \zeta')$ の簡約列で，i 番目（$i < n$）の簡約までは同じ列であり，次の $i+1$ 番目の簡約では異なる結果が得られるとする．$i+1$ 番目の簡約では，同じ節点に適用した同じ属性を簡約する．なぜなら，それ以外の場合では簡約結果が等しくなるからである．適用した節点の文字（関数）を f_{i+1} とし，このときの簡約をそれぞれ $(\theta_i, a_2'(w)) \overset{M_2,\eta_1}{\Rightarrow}_S (\theta_i \wedge f_{i+1}^{(l)} \circ \psi_{i+1}^{(l)}, \eta_2')$, $(\theta_i, a_2'(w)) \overset{M_2,\eta_1}{\Rightarrow}_S (\theta_i \wedge f_{i+1}^{(l)} \circ (\psi_{i+1}')^{(l)}, \eta_2'')$ とする．同じ節点に同じ属性を適用したので，ψ_{i+1} と ψ_{i+1}' は同じ左辺 (x, l) の規則の異なるガードであるから，M_2 が ASTT であることより，$[\![\psi_{i+1}]\!] \cap [\![\psi_{i+1}']\!] = \emptyset$ である．従って，$[\![\theta_1]\!] \cap [\![\theta_2]\!] = \emptyset$ が成立する．

2 の証明　上の証明と同様に，$(\varphi_1^{(k)}, a_2(w))$ の正規形を簡約する全ての簡約列のうち，i 番目（$i < n$）までの簡約列は等しく，$i+1$ 番目の簡約は異なる任意の n 個の簡約列をそれぞれ $(\varphi_1, a_2(w))\ (\overset{M_1,\eta_1}{\Rightarrow}_S)^*\ (\theta_1, \zeta_1), \ldots, (\varphi_1, a_2(w))\ (\overset{M_1,\eta_1}{\Rightarrow}_S)^*\ (\theta_n, \zeta_n)$ とする．$i+1$ 番目の簡約をそれぞれ $(\theta, x) \overset{M_1,\eta_1}{\Rightarrow}_S (\theta \wedge f^{(l)} \circ \psi_1^{(l)}, \eta_1'), \ldots, (\theta, x) \overset{M_1,\eta_1}{\Rightarrow}_S (\theta \wedge f^{(l)} \circ \psi_n^{(l)}, \eta_n')$ とする．このとき，記号的導出関係の定義より，$\psi_1, \ldots, \psi_n$ は同じ左辺 (x, l) に当てはまる全ての規則のガードであるから，$\{\rho \in R_2 \mid \mathrm{lhs}(\rho) = (x, l)\} = \{\rho \in R_2 \mid \mathrm{grd}(\rho) \in \{\psi_1, \ldots, \psi_n\}\}$．$M_2$ は ASTT であるから，$\bigcup_{i\in[n]} [\![\psi_i]\!] = \Sigma_2^{(l)}$．$f : \Sigma_1^{(k)} \to \Delta_1^{(l)}$ である．したがって $\Delta_1 \subseteq \Sigma_2$ より $\bigcup_{i\in[n]} [\![f \circ \psi_i]\!] = \Sigma_1^{(k)}$．このような n のうち最大の値を m とおくと，

$$\bigcup_{\substack{(\langle a_1,a_2\rangle(w)\overset{\theta,l}{\to}\zeta)\in R \\ [\![\theta]\!]\subseteq[\![\varphi_1]\!]}} [\![\theta]\!] = \bigcup_{\substack{(\langle a_1,a_2\rangle(w)\overset{\theta,l}{\to}\zeta)\in R \\ [\![\theta]\!]\subseteq[\![\varphi_1]\!]}} [\![\varphi_1 \wedge f_1 \circ \psi_1 \wedge \cdots \wedge f_n \circ \psi_n]\!]$$
$$= [\![\varphi_1]\!] \cap \bigcup_{i\in[1]} [\![f_1 \circ \psi_i]\!] \cap \cdots \cap \bigcup_{i\in[m]} [\![f_m \circ \psi_i]\!]$$

$$= [\![\varphi_1^{(k)}]\!] \cap \Sigma_1^{(k)} \cap \cdots \cap \Sigma^{(k)} = [\![\varphi_1]\!].$$

□

例 6.5. M_{abs_leaves} は SUR を満たすので，$M_{abs\_leaves} \fatsemi M_{half\_rev} = (Syn, Inh, \{\mathrm{bin}^{(2)}\} \cup \mathbb{Z}^{(0)}, \Phi, \mathbb{N} \cup \{\epsilon\})$ が存在し，次を満たす:

$$\begin{aligned}
Syn &= \{a_1\} \times \{a_2\} \cup \{b_1\} \times \{b_2\} = \{\langle a_1, a_2\rangle, \langle b_1, b_2\rangle\} \\
Inh &= \{a_1\} \times \{b_2\} \cup \{b_1\} \times \{a_2\} = \{\langle a_1, b_2\rangle, \langle b_1, a_2\rangle\} \\
\Phi &= \{(<0)\} \cup \{abs \circ even, abs \circ \neg even, abs \circ \top, id \circ even, id \circ \neg even, id \circ \top\} \\
R = \{ & \ \langle a_1, a_2\rangle(\pi) \overset{\flat_1, 1}{\to} \langle a_1, a_2\rangle(\pi 1), \\
& \langle a_1, b_2\rangle(\pi 1) \overset{\flat_1, 1}{\to} \hat{\epsilon}, \\
& \langle a_1, a_2\rangle(\pi) \overset{\top, 2}{\to} \langle a_1, a_2\rangle(\pi 1), \\
& \langle a_1, b_2\rangle(\pi 1) \overset{\top, 2}{\to} \langle a_1, b_2\rangle(\pi), \\
& \langle a_1, a_2\rangle(\pi) \overset{(<0)\wedge abs\circ\top, 0}{\to} \langle b_1, a_2\rangle(\pi), \\
& \langle b_1, b_2\rangle(\pi) \overset{(<0)\wedge abs\circ even, 0}{\to} (abs \circ (\div 2))(\langle a_1, b_2\rangle(\pi)), \\
& \langle b_1, b_2\rangle(\pi) \overset{(<0)\wedge abs\circ\neg even, 0}{\to} (abs \circ id)(\langle a_1, b_2\rangle(\pi)), \\
& \langle a_1, a_2\rangle(\pi) \overset{\neg(<0)\wedge id\circ\top, 0}{\to} \langle b_1, a_2\rangle(\pi), \\
& \langle b_1, b_2\rangle(\pi) \overset{\neg(<0)\wedge id\circ even, 0}{\to} (id \circ (\div 2))(\langle a_1, b_2\rangle(\pi)), \\
& \langle b_1, b_2\rangle(\pi) \overset{\neg(<0)\wedge id\circ\neg even, 0}{\to} (id \circ id)(\langle a_1, b_2\rangle(\pi)), \\
& \langle b_1, a_2\rangle(\pi 1) \overset{\flat_1, 1}{\to} \langle b_1, b_2\rangle(\pi 1), \\
& \langle b_1, a_2\rangle(\pi 1) \overset{\top, 2}{\to} \langle a_1, a_2\rangle(\pi 2), \\
& \langle a_1, b_2\rangle(\pi 2) \overset{\top, 2}{\to} \langle b_1, b_2\rangle(\pi 1), \\
& \langle b_1, a_2\rangle(\pi 2) \overset{\top, 2}{\to} \langle b_1, a_2\rangle(\pi), \\
& \langle b_1, b_2\rangle(\pi) \overset{\top, 2}{\to} \langle b_1, b_2\rangle(\pi 2) \qquad\qquad \}
\end{aligned}$$

各規則は次のように導かれる:

- $(a_1(\pi) \overset{\flat_1, 1}{\to} a_1(\pi 1)) \in R_1$ に対して
  - a_2 を適用して

$$\begin{aligned}
(\flat_1, a_2(\varepsilon)) &\overset{M_{rev}, \hat{\sharp_2}(a_1(\pi 1))}{\Rightarrow_S} (\flat_1 \wedge \hat{\sharp_2} \circ \flat_2, a_2(1)) \qquad (\because (a_2(\pi) \overset{\flat_2}{\to} a_2(\pi 1)) \in R_2) \\
&\overset{M_{rev}, \hat{\sharp_2}(a_1(\pi 1))}{\Rightarrow_S} (\flat_1 \wedge \hat{\sharp_2} \circ \flat_2, \langle a_1, a_2\rangle(\pi 1))
\end{aligned}$$

  よって $(\langle a_1, a_2\rangle(\pi) \overset{\flat_1, 1}{\to} \langle a_1, a_2\rangle(\pi 1)) \in R$.

– b_2 を適用して

$$(\flat_1, b_2(1)) \overset{M_{rev}, \hat{\sharp_2}(a_1(\pi 1))}{\Rightarrow_S} (\flat_1 \wedge \hat{\sharp_2} \circ \flat_2, \hat{\epsilon}) \qquad (\because (b_2(\pi 1) \overset{\flat_2}{\to} \hat{\epsilon}) \in R_2)$$

よって $(\langle a_1, b_2\rangle(\pi 1) \overset{\flat_1,1}{\to} \hat{\epsilon}) \in R$.

- $(a_1(\pi) \overset{\top,2}{\to} a_1(\pi 2)) \in R_1$ に対して，
  – a_2 を適用して

$$(\top, a_2(\varepsilon)) \overset{M_{rev}, a_1(\pi 2)}{\Rightarrow_S} (\top, \langle a_1, a_2\rangle(\pi 2))$$

よって $(\langle a_1, a_2\rangle(\pi) \overset{\top,2}{\to} \langle a_1, a_2\rangle(\pi 2)) \in R$.

– b_2 を適用して，$b_2(\varepsilon)$ を $\langle a_1, b_2\rangle(\pi)$ に置換し，$(\langle a_1, b_2\rangle(\pi) \overset{\top,2}{\to} \langle a_1, b_2\rangle(\pi)) \in R$.

- $(a_1(\pi) \overset{(<0),0}{\to} abs(b_1(\pi))) \in R_1$ に対して
  – a_2 を適用して

$$((<0), a_2(\varepsilon)) \overset{M_{rev}, abs(b_1(\pi))}{\Rightarrow_S} ((<0) \wedge abs \circ \top, a_2(1)) \quad (\because (a_2(\pi) \overset{\top,1}{\to} a_2(\pi 1)) \in R_2)$$
$$\overset{M_{rev}, abs(b_1(\pi))}{\Rightarrow_S} ((<0) \wedge abs \circ \top, \langle b_1, a_2\rangle(\pi))$$

よって，$(\langle a_1, a_2\rangle(\pi) \overset{(<0)\wedge abs\circ\top,0}{\to} \langle b_1, a_2\rangle(\pi)) \in R$.

– b_2 を適用して

$$((<0), b_2(1)) \overset{M_{rev}, abs(b_1(\pi))}{\Rightarrow_S} ((<0) \wedge abs \circ even, abs \circ (\div 2)(b_2(\varepsilon)))$$
$$(\because (b_2(\pi 1) \overset{even,1}{\to} (\div 2)(b_2(\pi))) \in R_2)$$
$$((<0), b_2(1)) \overset{M_{rev}, abs(b_1(\pi))}{\Rightarrow_S} ((<0) \wedge abs \circ \neg even, abs \circ id(b_2(\varepsilon)))$$
$$(\because (b_2(\pi 1) \overset{\neg even,1}{\to} id(b_2(\pi))) \in R_2)$$

$b_2(\varepsilon)$ を $\langle a_1, b_2\rangle(\pi)$ に置換し，

  * $(\langle b_1, b_2\rangle(\pi) \overset{(<0)\wedge abs\circ even,0}{\to} abs \circ (\div 2)(\langle a_1, b_2\rangle(\pi))) \in R$.
  * $(\langle b_1, b_2\rangle(\pi) \overset{(<0)\wedge abs\circ\neg even,0}{\to} abs \circ id(\langle a_1, b_2\rangle(\pi))) \in R$.

- $(a_1(\pi) \overset{\neg(<0),0}{\to} id(b_1(\pi))) \in R_1$ に対して
  – a_2 を適用して

$$(\neg(<0), a_2(\varepsilon)) \overset{M_{rev}, id(b_1(\pi))}{\Rightarrow_S} (\neg(<0) \wedge id \circ \top, a_2(1)) \quad (\because (a_2(\pi) \overset{\top,1}{\to} a_2(\pi 1)) \in R_2)$$
$$\overset{M_{rev}, id(b_1(\pi))}{\Rightarrow_S} (\neg(<0) \wedge id \circ \top, \langle b_1, a_2\rangle(\pi))$$

よって，$(\langle a_1, a_2\rangle(\pi) \overset{\neg(<0)\wedge id\circ\top,0}{\to} \langle b_1, a_2\rangle(\pi)) \in R$.

– b_2 を適用して

$$(\neg(<0), b_2(1)) \overset{M_{rev}, id(b_1(\pi))}{\Rightarrow_S} (\neg(<0) \wedge id \circ even, id \circ (\div 2)(b_2(\varepsilon)))$$
$$(\because (b_2(\pi 1) \overset{even,1}{\to} (\div 2)(b_2(\pi))) \in R_2)$$

$$(\neg(<0), b_2(1)) \overset{M_{rev}, id(b_1(\pi))}{\Rightarrow_S} (\neg(<0) \wedge id \circ \neg even, id \circ id(b_2(\varepsilon))) \quad (\because (b_2(\pi 1) \overset{\neg even, 1}{\to} id(b_2(\pi))) \in R_2)$$

$b_2(\varepsilon)$ を $\langle a_1, b_2\rangle(\pi)$ に置換し，

- $(\langle b_1, b_2\rangle(\pi) \overset{\neg(<0)\wedge id\circ even, 0}{\to} id \circ (\div 2)(\langle a_1, b_2\rangle(\pi))) \in R$.
- $(\langle b_1, b_2\rangle(\pi) \overset{\neg(<0)\wedge id\circ\neg even, 0}{\to} id \circ id(\langle a_1, b_2\rangle(\pi))) \in R$.

- $(b_1(\pi 1) \overset{\flat_1, 1}{\to} \hat{\epsilon}) \in R_1$ に対して a_2 を適用して

$$(\flat_1, a_2(\varepsilon)) \overset{M_{rev}, \hat{\epsilon}}{\Rightarrow_S} (\flat_1 \wedge \hat{\epsilon} \circ \top, b_2(\varepsilon)) \quad (\because (a_2(\pi) \overset{\top, 0}{\to} b_2(\pi)) \in R_2)$$

  $b_2(\varepsilon)$ を $\langle b_1, b_2\rangle(\pi 1)$ に置換し，$(\langle b_1, a_2\rangle(\pi 1) \overset{\flat_1, 1}{\to} \langle b_1, b_2\rangle(\pi 1)) \in R$.
- $(b_1(\pi 1) \overset{\top, 2}{\to} a_1(\pi 2)) \in R_1$ に対して
  - a_2 を適用して

$$(\top, a_2(\varepsilon)) \overset{M_{rev}, a_1(\pi 2)}{\Rightarrow_S} (\top, \langle a_1, a_2\rangle(\pi 2))$$

    よって $(\langle b_1, a_2\rangle(\pi 1) \overset{\top, 2}{\to} \langle a_1, a_2\rangle(\pi 2)) \in R$.
  - b_2 を適用して，$b_2(\varepsilon)$ を $\langle b_1, b_2\rangle(\pi 1)$ に置換し，$(\langle a_1, b_2\rangle(\pi 2) \overset{\top, 2}{\to} \langle b_1, b_2\rangle(\pi 1)) \in R$.
- $(b_1(\pi 2) \overset{\top, 2}{\to} b_1(\pi)) \in R_1$ に対して
  - a_2 を適用して

$$(\top, a_2(\varepsilon)) \overset{M_{rev}, b_1(\pi)}{\Rightarrow_S} (\top, \langle b_1, a_2\rangle(\pi))$$

    よって $(\langle b_1, a_2\rangle(\pi 2) \overset{\top, 2}{\to} \langle b_1, a_2\rangle(\pi)) \in R$.
  - b_2 を適用して，$b_2(\varepsilon)$ を $\langle b_1, b_2\rangle(\pi 2)$ に置換し，$(\langle b_1, b_2\rangle(\pi) \overset{\top, 2}{\to} \langle b_1, b_2\rangle(\pi 2)) \in R$.

木 $\xi = \mathrm{bin}(\mathrm{bin}(1, -2), \mathrm{bin}(\mathrm{bin}(-3, 4), 5)) \in T_{\mathbb{Z}\cup\{\mathrm{bin}\}}$ に対して，$\mathbf{T}_{M_{abs\_leaves} \S M_{half\_rev}}(\xi) = nf(\overset{M_{abs\_leaves} \S M_{half\_rev}, \sharp_1(\xi)}{\Rightarrow}, \langle a_1, a_2\rangle(\varepsilon)) = 5(2(3(1(1(\epsilon))))) = \mathbf{T}_{M_{abs\_leaves}} \circ \mathbf{T}_{M_{half\_rev}}(\xi)$ であり，図 6.1に示す簡約で計算される．

6.3 属性付き記号的木変換器の合成の正当性

本節では，合成アルゴリズムの正当性の証明について述べる．まず，合成した ASTT を正規化することで，各規則のガードを揃える．

定義 6.6 (ASTT の正規化)．ASTT $M = (Syn, Inh, \Sigma, \Phi, \Delta, in, \sharp, \flat, R)$ に対して，$\|M\| = (Syn, Inh, \Sigma, \Phi, \Delta, in, \sharp, \flat, R')$ を M の正規化と呼ぶ．ここで R' は次で定義する:

任意の $ch \in CH, l \in rk(\Sigma)$ に対して，述語

$$\varphi' = \bigwedge_{\rho \in ch(LHS(M)) \cap R^{(l)}} \mathrm{grd}(\rho).$$

$\langle a_1, a_2\rangle(\varepsilon) \Rightarrow \langle a_1, a_2\rangle(1)$
$\Rightarrow \langle a_1, a_2\rangle(11)$
$\Rightarrow \langle a_1, a_2\rangle(111)$
$\Rightarrow \langle b_1, a_2\rangle(111)$
$\Rightarrow \langle a_1, a_2\rangle(112)$
$\Rightarrow \langle b_1, a_2\rangle(112)$
$\Rightarrow \langle b_1, a_2\rangle(11)$
$\Rightarrow \langle a_1, a_2\rangle(12)$
$\Rightarrow \langle a_1, a_2\rangle(121)$
$\Rightarrow \langle a_1, a_2\rangle(1211)$
$\Rightarrow \langle b_1, a_2\rangle(1211)$
$\Rightarrow \langle a_1, a_2\rangle(1212)$
$\Rightarrow \langle b_1, a_2\rangle(1212)$
$\Rightarrow \langle b_1, a_2\rangle(121)$
$\Rightarrow \langle a_1, a_2\rangle(122)$
$\Rightarrow \langle b_1, a_2\rangle(122)$
$\Rightarrow \langle b_1, a_2\rangle(12)$
$\Rightarrow \langle b_1, a_2\rangle(1)$
$\Rightarrow \langle b_1, b_2\rangle(1)$
$\Rightarrow \langle b_1, b_2\rangle(12)$
$\Rightarrow \langle b_1, b_2\rangle(122)$（右上に続く）

$\Rightarrow 5(\langle a_1, b_2\rangle(122))$
$\Rightarrow 5(\langle b_1, b_2\rangle(121))$
$\Rightarrow 5(\langle b_1, b_2\rangle(1212))$
$\Rightarrow 5(2(\langle a_1, b_2\rangle(1212)))$
$\Rightarrow 5(2(\langle b_1, b_2\rangle(1211)))$
$\Rightarrow 5(2(3(\langle a_1, b_2\rangle(1211))))$
$\Rightarrow 5(2(3(\langle a_1, b_2\rangle(121))))$
$\Rightarrow 5(2(3(\langle a_1, b_2\rangle(12))))$
$\Rightarrow 5(2(3(\langle b_1, b_2\rangle(11))))$
$\Rightarrow 5(2(3(\langle b_1, b_2\rangle(112))))$
$\Rightarrow 5(2(3(1(\langle a_1, b_2\rangle(112)))))$
$\Rightarrow 5(2(3(1(\langle b_1, b_2\rangle(111)))))$
$\Rightarrow 5(2(3(1(1(\langle a_1, b_2\rangle(111))))))$
$\Rightarrow 5(2(3(1(1(\langle a_1, b_2\rangle(11))))))$
$\Rightarrow 5(2(3(1(1(\langle a_1, b_2\rangle(1))))))$
$\Rightarrow 5(2(3(1(1(\epsilon)))))$

図 6.1: ASTT $M_{abs} \fatsemi M_{even}$ による簡約例

が $[\![\varphi']\!] \neq \emptyset$ ならば，$(a(w) \overset{\varphi', l}{\to} \mathrm{rhs}(ch((a(w), l)))) \in R'$．ただし CH は，$LHS(M)$ から R への任意の関数 ch のうち，

$$\forall (a(w), l) \in LHS(M). \exists \rho \in R. \mathrm{lhs}(\rho) = (a(w), l), \rho = ch((a(w), l))$$

を満たすもの全ての集合である．

上の正規化 $\|M\|$ の定義では，任意のランク l に対して，$R^{(l)}$ のうち互いに異なる全ての左辺の規則を，ch によって 1 つずつ取り出して，それらのガードの積を新しいガードとして扱っている．任意の ch による取り出し方でガードの積を取っているので，命題 5.2 より，任意の $\sigma \in \Sigma$ に対して選ばれる規則のガードの積が，$\|M\|$ の規則のガードとして含まれる．つまり，任意の $\sigma^{(l)} \in \Sigma$ に対して，そのランク l に対応する任意の $(x, l) \in LHS(M)$ に対して $\sigma \in [\![\mathrm{grd}(ch((x, l)))]\!]$，$ch((x, l)) \in R$，$\sigma \in [\![\varphi']\!]$，$(x \overset{\varphi', l}{\to} \mathrm{rhs}(ch(x, l))) \in R'$ が成立する．従っ

て $\|M\|$ は定義可能な ASTT である．さらに，選ばれた規則の左辺と右辺はそのまま $\|M\|$ の規則に用いているから，$\mathbf{T}_M = \mathbf{T}_{\|M\|}$ である．

次に，正規化した ASTT のガードや右辺の関数を用いて，合成前の 2 つの ASTT を ATT に符号化する．

定義 6.7 (ASTT の符号化)**.** 任意の 2 つの ASTT

$$M_1 = (Syn_1, Inh_1, \Sigma_1, \Phi_1, \Delta_1, in_1, \sharp_1, \flat_1, R_1)$$
$$M_2 = (Syn_2, Inh_2, \Sigma_2, \Phi_2, \Delta_2, in_2, \sharp_2, \flat_2, R_2)$$

に対して，

$$\|M_1 \fatsemi M_2\| = (Syn, Inh, \Sigma_1, \Phi, \Delta_2, in, \sharp_1, \flat_1, R')$$

が存在するとき，(M_1, M_2) の ATT への符号化 (M_1', M_2') を $E(M_1, M_2)$ と表し，次で定義する:

$$M_1' = (Syn_1, Inh_1, \Sigma', \Delta', in_1, \overline{\flat_1}^{(1)}, R_1')$$
$$M_2' = (Syn_2, Inh_2, \Delta', \Omega', in_2, \overline{\flat_2}^{(1)}, R_2')$$

ただし，

- 任意の $(a_1(w_1) \overset{\varphi_1, l_1}{\to} \eta_1) \in R_1$ と $\varphi'^{(l_1)} \in \mathrm{BC}(\Phi)$ に対して，$\mathrm{grd}(\rho') = \varphi'$ を満たす $\rho' \in R'$ が存在し，かつ $[\![\varphi']\!] \subseteq [\![\varphi_1]\!]$ ならば，$(a_1(w_1) \overset{\overline{\varphi'}^{(l_1)}}{\to} give_{\eta_1}([\![\varphi']\!])) \in R_1'$ かつ $\overline{\varphi'}^{(l_1)} \in \Sigma'$.
- 任意の $\rho_1' \in R_1', w \in path(\mathrm{rhs}(\rho_1'))$ に対して $(lab_{\mathrm{rhs}(\rho_1')}(w))^{(l)} \in F(\Sigma_1^+ \to \Delta_1^{(l)})$ ならば $(lab_{\mathrm{rhs}(\rho_1')}(w))^{(l)} \in \Delta'$.
- 任意の $\overline{f_1([\![\varphi']\!])}^{(l_2)} \in \Delta', (x_2 \overset{\varphi_2, l_2}{\to} \eta_2) \in R_2$ に対して

$$\begin{cases} f_1([\![\varphi']\!]) \not\subseteq [\![\varphi_2]\!] \text{ のとき} (x_2 \overset{\overline{f_1([\![\varphi']\!])}^{(l_2)}}{\to} nil) \in R_2'. \\ f_1([\![\varphi']\!]) \subseteq [\![\varphi_2]\!] \text{ のとき} (x_2 \overset{\overline{f_1([\![\varphi']\!])}^{(l_2)}}{\to} comp_{\eta_2}(f_1)) \in R_2'. \end{cases}$$

  ただし $nil^{(0)} \notin \Delta_2$ を文字とする．
- 任意の $(x_2 \overset{\flat_2, 1}{\to} \eta_2) \in R_2$ に対して $(x_2 \overset{\overline{\flat_2}^{(1)}}{\to} comp_{\eta_2}(\hat{\sharp_2})) \in R_2'$.
- 任意の $\rho_2' \in R_2', w \in path(\mathrm{rhs}(\rho_2'))$ に対して $lab_{\mathrm{rhs}(\rho_2')}(w) \notin OCC(M_2)$ ならば $(lab_{\mathrm{rhs}(\rho_2')}(w))^{(rk_{\mathrm{rhs}(\rho_2')}(w))} \in \Omega'$.

正規化の定義より，任意の規則 $(x \overset{\varphi', l}{\to} \eta') \in R'$ に対して，同じガードを持つ規則 $(x' \overset{\varphi', l}{\to} \eta'') \in R'$ が，ランクが同じ全ての左辺 $(x', l) \in LHS(\|M_1 \fatsemi M_2\|)$ に対して存在する．従って $[\![\varphi']\!] \subseteq [\![\varphi_1]\!]$ を満たすなら，任意の左辺 $(x, l) \in LHS(M_1)$ に対して規則 $(a_1(w_1) \overset{\overline{\varphi'}^{(l_1)}}{\to}$

$give_{\eta_1}([\![\varphi']\!])) \in R'_1$ かつ $\overline{\varphi'}^{(l_1)} \in \Sigma'$ が存在する．さらに次が成立するので，M_1 が定義可能な ASTT なら，M'_1 は定義可能な ATT であり，さらに M_1 が SUR を満たすなら，M'_1 は SUR を満たす．

命題 6.8. 任意の 2 つの ASTT M_1, M_2 に対して，$\|M_1 \fatsemi M_2\|$ が存在し，その規則の集合を R' とする．このとき，任意の $(x \overset{\varphi_1,l_1}{\to} \eta_1) \in R_1, a_2 \in Syn_2 \cup Inh_2$ に対して，

$$\bigcup_{\substack{(x' \overset{\varphi',l_1}{\to} \zeta) \in R' \\ [\![\varphi']\!] \subseteq [\![\varphi_1]\!]}} [\![\varphi']\!] = [\![\varphi_1]\!].$$

証明. $M_1 \fatsemi M_2$ の規則の集合を R とする．命題 6.4 の証明より，

$$\bigcup_{\substack{(x \overset{\varphi,l_1}{\to} \zeta) \in R \\ [\![\varphi]\!] \subseteq [\![\varphi_1]\!]}} [\![\varphi]\!] = [\![\varphi_1]\!].$$

正規化の定義より，

$$\bigcup_{\substack{(x' \overset{\varphi',l_1}{\to} \zeta) \in R' \\ [\![\varphi']\!] \subseteq [\![\varphi]\!]}} [\![\varphi']\!] = [\![\varphi]\!].$$

以上より，

$$\bigcup_{\substack{(x' \overset{\varphi',l_1}{\to} \zeta) \in R' \\ [\![\varphi']\!] \subseteq [\![\varphi_1]\!]}} [\![\varphi']\!] = \bigcup_{\substack{(x \overset{\varphi,l_1}{\to} \zeta) \in R \\ [\![\varphi]\!] \subseteq [\![\varphi_1]\!]}} \left(\bigcup_{\substack{(x' \overset{\varphi',l_1}{\to} \zeta) \in R' \\ [\![\varphi']\!] \subseteq [\![\varphi]\!]}} [\![\varphi']\!] \right) = [\![\varphi_1]\!].$$

□

また，Δ' は R'_1 が含む全ての規則の右辺に現れるラベルを含んでいるから，符号化の定義より M_2 が定義可能な ASTT なら，M'_2 は定義可能な ATT である．よって $M_1 \fatsemi M_2$ が定義可能な ASTT なら，$M'_1 \fatsemi M'_2$ は定義可能な ATT である．

そして，次の命題より，符号化で得た 2 つの ATT を合成すると，もとの ASTT の合成と見かけ上等価な ATT が得られる．

命題 6.9. 任意の ASTT M_1 と ASTT M_2 に対して，$E(M_1, M_2) = (M'_1, M'_2)$ が存在し，

$$\|M_1 \fatsemi M_2\| = (Syn, Inh, \Sigma, \Phi, \Omega, in, \flat, R')$$
$$M'_1 \fatsemi M'_2 = (Syn', Inh', \Sigma', \Omega', in', \overline{\flat}^{(1)}, R'')$$

とするとき，次が成立する:

1. $Syn' = Syn, Inh' = Inh, in' = in$.
2. 任意の $\sigma \in \Sigma$ に対して，$\sigma \in [\![\varphi']\!]$ を満たす $(x \overset{\varphi',l}{\to} \zeta) \in R', (x \overset{\overline{\varphi'}^{(l)}}{\to} \zeta) \in R''$ がそれぞれ唯 1 つ存在する．

証明.

$$M_1' = (Syn_1, Inh_1, \Sigma', \Delta', in_1, \overline{\flat_1}^{(1)}, R_1')$$
$$M_2' = (Syn_2, Inh_2, \Delta', \Omega', in_2, \overline{\flat_2}^{(1)}, R_2')$$

とする.

1 の証明　まず，M_1', M_2' はそれぞれ M_1, M_2 の属性をそのまま受け継ぐ．また，ASTT の合成では，記号的導出関係の定義より，M_1 の規則の右辺の属性に M_2 の属性を適用したときの簡約が ATT の合成と等価なので，$Syn' = Syn, Inh' = Inh, in' = in$.

2 の証明　M_1, M_2, $M_1 \fatsemi M_2$ の規則の集合をそれぞれ R_1, R_2, R とする．$\|M_1 \fatsemi M_2\|$ は ASTT であるから，$\sigma \in [\![\varphi']\!]$ を満たす $(x \overset{\varphi', l}{\to} \zeta) \in R'$ が唯 1 つ存在する．合成の定義と正規化の定義より，φ' が $\varphi' = \theta_1 \wedge \theta_2 \wedge \cdots \wedge \theta_n$ のように構成されていて，任意の左辺 $(x', l) \in LHS(M_1 \fatsemi M_2)$ に対して，$(x' \overset{\theta_i, l}{\to} \zeta) \in R$ を満たす $i \in [n]$ が存在する．命題 6.8 より，$[\![\varphi']\!] \subseteq [\![\varphi_1]\!]$ を満たす $(a_1'(w_1') \overset{\varphi_1, l}{\to} \eta_1) \in R_1$ が唯 1 つ存在する．$x = \langle a_1, a_2 \rangle(w)$ とおくと，$a_2 \in Syn_2$ のときは $a_1 = a_1'$ かつ $w = w_1'$ であり，$(\varphi_1, a_2(\varepsilon))\ (\overset{M_2, \eta_1}{\Rightarrow_S})^*\ (\theta_i, \zeta')$ である．$a_2 \in Inh_2$ のときは $lab_{\eta_1}(w') = a_1(w)$ を満たす $w' \in path(\eta_1)$ が存在し，$(\varphi_1, a_2(w'))\ (\overset{M_2, \eta_1}{\Rightarrow_S})^*\ (\theta_i, \zeta')$ である．ここで ζ' に出現する全ての継承属性 $b_2'(\varepsilon)$ を合成の手続きに従って置き換えると ζ が得られる．

各ガード θ_i は $\theta_i = \varphi_1 \wedge f_1^{(l_1)} \circ \psi_1^{(l_1)} \wedge f_2^{(l_2)} \circ \psi_2^{(l_2)} \wedge \cdots \wedge f_m^{(l_m)} \circ \psi_m^{(l_m)}$ のように構成されている．ただし $f_1, \ldots, f_m$ は R_1 の右辺のラベルであり，$\psi_1, \ldots, \psi_m$ は R_2 のガードである．また $m \geq 1$ とする．各 $f_j^{(l_j)} \circ \psi_j^{(l_j)}$ に対して，$f_j([\![\theta_i]\!]) = f_j([\![\varphi_1]\!]) \cap f_j([\![f_1 \circ \psi_1]\!]) \cap \cdots \cap [\![\psi_j]\!] \cap \cdots \cap f_j([\![f_m \circ \psi_m]\!])$ となる．よって $f_j([\![\theta_i]\!]) \subseteq [\![\psi_j]\!]$．$M_2$ は ASTT であるから，$f_j([\![\theta_i]\!]) \subseteq [\![\psi]\!]$ を満たす R_2 のガード ψ は高々 1 つしかないので，$\psi = \psi_j$ である．従って，$(\varphi_1, a_2(\varepsilon))\ (\overset{M_2, \eta_1}{\Rightarrow_S})^*\ (\theta_i, \zeta')$ または $(\varphi_1, a_2(w'))\ (\overset{M_2, \eta_1}{\Rightarrow_S})^*\ (\theta_i, \zeta')$ の簡約列の中で，η_1 の上の f_j でラベル付けされた節点における簡約では，規則 $(a_2(w_2) \overset{\psi_j, l_j}{\to} \eta_2) \in R_2$ が使われる．

$f_j([\![\theta_i]\!]) \subseteq [\![\psi_j]\!]$ より $f_j([\![\theta_i]\!]) \subseteq [\![\varphi']\!]$ だから，符号化の定義より，2 つの規則 $(a_1(w_1) \overset{\overline{\varphi'}^{(l)}}{\to} give_{\eta_1}([\![\varphi']\!])) \in R_1'$ と $\rho_2 = (a_2(w_2) \overset{\overline{f_j([\![\varphi']\!])}^{(l)}}{\to} comp_{\eta_2}(f_j)) \in R_2'$ の対が唯 1 つ存在する．ρ_2 は $give_{\eta_1}([\![\varphi']\!])$ の上の $f_j([\![\varphi']\!])$ でラベル付けされた節点において，$a_2(w_2')$ のような属性の出現を $comp_{\eta_2}(f_j)$ に簡約する．この結果は記号的導出関係による簡約の結果に等しい．以上のことは任意の i と j に対して成立するから，$\zeta' = nf(\overset{M_2, \eta_1}{\Rightarrow}, a_2(\varepsilon))$ または $\zeta' = nf(\overset{M_2, \eta_1}{\Rightarrow}, a_2(w'))$．すなわち，規則 $(x \overset{\overline{\varphi'}^{(l)}}{\to} \zeta) \in R''$ が唯 1 つ存在する．　□

例 6.10. $E(M_{abs\_leaves}, M_{half\_rev}) = (M'_{abs}, M'_{even})$ の規則の集合 R'_{abs}, R'_{even} は次を満たす:

$$
\begin{aligned}
R'_{abs} = \{\ & a_1(\pi) \overset{\overline{b_1}^{(1)}}{\to} a_1(\pi 1), && ,\\
& a_1(\pi) \overset{\overline{\top}^{(2)}}{\to} a_1(\pi 1),\\
& a_1(\pi) \overset{\overline{\top\wedge(<0)\wedge abs\circ even}^{(0)}}{\to} \overline{abs([\![\top \wedge (<0) \wedge abs \circ even]\!])}(b_1(\pi)),\\
& a_1(\pi) \overset{\overline{\top\wedge(<0)\wedge abs\circ\neg even}^{(0)}}{\to} \overline{abs([\![\top \wedge (<0) \wedge abs \circ \neg even]\!])}(b_1(\pi)),\\
& a_1(\pi) \overset{\overline{\top\wedge\neg(<0)\wedge id\circ even}^{(0)}}{\to} \overline{id([\![\top \wedge \neg(<0) \wedge id \circ even]\!])}(b_1(\pi)),\\
& a_1(\pi) \overset{\overline{\top\wedge\neg(<0)\wedge id\circ\neg even}^{(0)}}{\to} \overline{id([\![\top \wedge \neg(<0) \wedge id \circ \neg even]\!])}(b_1(\pi)),\\
& b_1(\pi 1) \overset{\overline{b_1}^{(1)}}{\to} \overline{\hat{\epsilon}([\![b_1]\!])},\\
& b_1(\pi 1) \overset{\overline{\top}^{(2)}}{\to} a_1(\pi 2),\\
& b_1(\pi 2) \overset{\overline{\top}^{(2)}}{\to} b_1(\pi) && \}\\
R'_{even} = \{\ & a_2(\pi) \overset{\overline{b_2}^{(1)}}{\to} a_2(\pi 1),\\
& a_1(\pi) \overset{\overline{abs([\![\top\wedge(<0)\wedge abs\circ even]\!])}^{(1)}}{\to} a_2(\pi 1),\\
& a_1(\pi) \overset{\overline{abs([\![\top\wedge(<0)\wedge abs\circ\neg even]\!])}^{(1)}}{\to} a_2(\pi 1),\\
& a_1(\pi) \overset{\overline{id([\![\top\wedge\neg(<0)\wedge id\circ even]\!])}^{(1)}}{\to} a_2(\pi 1),\\
& a_1(\pi) \overset{\overline{id([\![\top\wedge\neg(<0)\wedge id\circ\neg even]\!])}^{(1)}}{\to} a_2(\pi 1),\\
& a_2(\pi) \overset{\overline{\hat{\epsilon}([\![b_1]\!])}^{(0)}}{\to} b_2(\pi),\\
& b_2(\pi 1) \overset{\overline{b_2}^{(1)}}{\to} \hat{\epsilon},\\
& b_2(\pi 1) \overset{\overline{abs([\![\top\wedge(<0)\wedge abs\circ even]\!])}^{(1)}}{\to} \overline{abs \circ (\div 2)}(b_2(\pi)),\\
& b_2(\pi 1) \overset{\overline{id([\![\top\wedge\neg(<0)\wedge id\circ even]\!])}^{(1)}}{\to} \overline{id \circ (\div 2)}(b_2(\pi)),\\
& b_2(\pi 1) \overset{\overline{abs([\![\top\wedge(<0)\wedge abs\circ\neg even]\!])}^{(1)}}{\to} \overline{abs \circ id}(b_2(\pi)),\\
& b_2(\pi 1) \overset{\overline{id([\![\top\wedge\neg(<0)\wedge id\circ\neg even]\!])}^{(1)}}{\to} \overline{id \circ id}(b_2(\pi)) && \}
\end{aligned}
$$

ただし右辺が nil である規則は省略した．ATT $M'_{abs} \fatsemi M'_{even}$ の規則は，ASTT $M_{abs\_leaves} \fatsemi M_{half\_rev}$ の規則と見かけ上同等になる．

定義 6.11. 任意の 2 つの ASTT M_1, M_2 に対して，$E(M, M_2) = (M'_1, M'_2)$ とし，M_1，M_2

の出力ランク付き文字集合をそれぞれ Δ_1，Δ_2 とする．

$$M_1' = (Syn_1, Inh_1, \Sigma', \Delta', in_1, \overline{\flat_1}^{(1)}, R_1')$$
$$M_2' = (Syn_2, Inh_2, \Delta', \Omega', in_2, \overline{\flat_2}^{(1)}, R_2')$$

のとき，任意の木 $\xi \in T_{\Sigma_1}$ に対して**インスタンス** $Ins_\xi(M_1, M_2) = (M_{1,\xi}', M_{2,\xi}')$ を次の ATT の対で定義する:

$$M_{1,\xi}' = (Syn_1, Inh_1, \Sigma_\xi, \Delta_\xi, in_1, \sharp_1^{(1)}, R_{1,\xi}')$$
$$M_{2,\xi}' = (Syn_2, Inh_2, \Delta_\xi, \Omega_\xi, in_2, \sharp_2^{(1)}, R_{2,\xi}')$$

ただし:

- 任意の $w \in path(\xi)$ に対して $lab_\xi(w) \in \Sigma_\xi$.
- 任意の $\sigma^{(l_1)} \in \Sigma_\xi^+$ と $(x_1 \overset{\overline{\varphi'}^{(l_1)}}{\to} \eta_1') \in R_1'$ に対して $\sigma \in [\![\varphi']\!]$ ならば $(x_1 \overset{\sigma^{(l_1)}}{\to} \eta_{1,\xi}') \in R_{1,\xi}'$. ただし，$\eta_{1,\xi}'$ は η_1' の上の各ラベル $f_1([\![\varphi']\!])$ を $f_1(\sigma)$ で置き換えたものである．すなわち，
  - $path(\eta_1') = path(\eta_{1,\xi}')$.
  - 任意の $w \in path(\eta_1')$ に対して，$lab_{\eta_1'}(w) = \overline{f_1([\![\varphi']\!])}, lab_{\eta_{1,\xi}'}(w) = f_1(\sigma)$ を満たす $\overline{f_1([\![\varphi']\!])} \in \Delta'$ が存在する．
- 任意の $\rho_{1,\xi}' \in R_{1,\xi}', w \in path(\mathrm{rhs}(\rho_{1,\xi}'))$ に対して，$lab_{\mathrm{rhs}(\rho_{1,\xi}')}(w) \in \Delta_1$ ならば $lab_{\mathrm{rhs}(\rho_{1,\xi}')}(w) \in \Delta_\xi$.
- 任意の $\sigma^{(l_1)} \in \Sigma_\xi$ と $(x_2 \overset{\overline{f_1([\![\varphi']\!])}^{(l_2)}}{\to} \eta_2') \in R_2'$ に対して $\sigma \in [\![\varphi']\!]$ ならば $(x_2 \overset{f_1(\sigma)^{(l_2)}}{\to} \eta_{2,\xi}') \in R_{2,\xi}'$. ただし，$\eta_{2,\xi}'$ は η_2' の上の各ラベル $f_1 \circ f_2$ を $(f_1 \circ f_2)(\sigma)$ で置き換えたものである．すなわち，
  - $path(\eta_2') = path(\eta_{2,\xi}')$.
  - 任意の $w \in path(\eta_2')$ に対して，$lab_{\eta_2'}(w) = \overline{f_1 \circ f_2}, lab_{\eta_{2,\xi}'}(w) = (f_1 \circ f_2)(\sigma)$ を満たす $\overline{f_1 \circ f_2} \in \Omega'$ が存在する．
- 任意の $\rho_{2,\xi}' \in R_{2,\xi}', w \in path(\mathrm{rhs}(\rho_{2,\xi}'))$ に対して，$lab_{\mathrm{rhs}(\rho_{2,\xi}')}(w) \in \Delta_2$ ならば $lab_{\mathrm{rhs}(\rho_{2,\xi}')}(w) \in \Omega_\xi$.

$Ins_\xi(M_1, M_2)$ の定義では，$E(M_1, M_2) = (M_1', M_2')$ の規則に現れるガードや像，合成関数を，ξ 中の文字をもとに，具体的な文字で置き換える．次を満たすので，インスタンスは ξ に対してのみ，もとの ASTT と意味論が等しい．

命題 6.12. 任意の ASTT M_1 と ASTT M_2 に対して，これらの合成が存在するものとする．また，M_1 の入力ランク付き文字集合を Σ_1 とする．このとき，任意の木 $\xi \in T_{\Sigma_1}$ に対して，$Ins_\xi(M_1, M_2) = (M_{1,\xi}', M_{2,\xi}')$ が存在し，いずれも定義可能である．そして M_1 が SUR を満た

すなら，$M'_{1,\xi}$ も SUR を満たし，

$$(\mathbf{T}_{M'_{1,\xi}} \circ \mathbf{T}_{M'_{2,\xi}})(\xi) = (\mathbf{T}_{M_1} \circ \mathbf{T}_{M_2})(\xi).$$

証明. $E(M_1, M_2) = (M'_1, M'_2)$ とする．ξ の上の全てのラベルの集合は有限なので，Σ_ξ，Δ_ξ，Ω_ξ も全て有限である．

命題 6.8 より，任意の $\sigma \in \Sigma_{1,\xi}$ と左辺 $(x_1, l_1) \in LHS(M_1)$ に対して，$\sigma \in [\![\varphi_1]\!]$ を満たす M_1 の規則 $(x_1 \overset{\varphi_1, l_1}{\to} \eta_1)$ と $M'_{1,\xi}$ の規則 $(x_1 \overset{\sigma^{(l_1)}}{\to} \eta'_{1,\xi})$ がそれぞれ唯 1 つ存在する．インスタンスの定義より $\eta_{1,\xi} = give_{\eta_1}(\sigma)$ だから，$M_{1,\xi}$ は定義可能な ATT であり，M_1 が SUR を満たすなら $M_{1,\xi}$ も SUR を満たす．そして意味論が等しいから $\mathbf{T}_{M_1}(\xi) = \mathbf{T}_{M'_{1,\xi}}(\xi)$.

任意の $\delta \in \Delta_\xi$ と左辺 $(x_2, l_2) \in LHS(M_2)$ に対して，唯 1 つの M_2 の規則 $(x_2 \overset{\varphi_2, l_2}{\to} \eta_2)$ が存在し，入力文字 $\sigma \in \Sigma_\xi$ と M'_2 の規則 $(x_2 \overset{\overline{f_1([\![\varphi']\!])}^{(l_2)}}{\to} comp_{\eta_2}(f_1))$ の対がいくつか存在して，それらは全て $f_1([\![\varphi']\!]) \subseteq [\![\varphi_2]\!]$，$\sigma \in [\![\varphi']\!]$，$f_1(\sigma) = \delta$ を満たす．インスタンスの定義により，任意のこのような文字と規則の対に対して，$M'_{2,\xi}$ の規則 $(x_2 \overset{\delta^{(l_2)}}{\to} \eta'_{2,\xi})$ の右辺は必ず同じ $\eta'_{2,\xi} = give_{comp_{\eta_2}(f_1)}(\sigma) = give_{\eta_2}(\delta)$ となる．従って，$\delta \in [\![\varphi_2]\!]$ を満たす $M'_{2,\xi}$ の規則 $(x_2 \overset{\delta^{(l_2)}}{\to} give_{\eta_2}(\delta))$ は唯 1 つ存在する．以上より $M'_{2,\xi}$ は定義可能な ATT であり，意味論が等しいから $\mathbf{T}_{M_2}(\mathbf{T}_{M_1}(\xi)) = \mathbf{T}_{M'_{2,\xi}}(\mathbf{T}_{M'_{1,\xi}}(\xi))$. □

命題 6.9 より，符号化された ATT を合成すると，もとの ASTT の合成に対応する規則で構成された ATT が得られる．$Ins_\xi(M_1, M_2)$ では符号化された ATT の規則の文字（述語や像，関数）を具体的に置き換えているだけなので，この合成ももとの ASTT の合成に対応する規則で構成される．さらに，$M'_{1,\xi} \fatsemi M'_{2,\xi}$ の入力ランク付き文字集合や出力ランク付き文字集合はそれぞれ $M_1 \fatsemi M_2$ の入力ランク付き文字集合や出力ランク付き文字集合に含まれているから，$M'_{1,\xi} \fatsemi M'_{2,\xi}$ も命題 6.9 と同じ性質を受け継ぐ．したがって次が成立する．

命題 6.13. 任意の ASTT M_1 と ASTT M_2 に対して，これらの合成が存在するものとする．また，M_1 の入力ランク付き文字集合を Σ_1 とする．このとき，任意の木 $\xi \in T_{\Sigma_1}$ に対して，$Ins_\xi(M_1, M_2) = (M'_{1,\xi}, M'_{2,\xi})$ が存在し，

$$\mathbf{T}_{M'_{1,\xi} \fatsemi M'_{2,\xi}}(\xi) = \mathbf{T}_{M_1 \fatsemi M_2}(\xi).$$

系 3.11 より，$Ins_\xi(M_1, M_2)$ の合成については正当性が保証されるので，次が成立する．

系 6.14. 任意の ASTT M_1 と ASTT M_2 に対して，これらの合成が存在するものとする．また，M_1 の入力ランク付き文字集合を Σ_1 とする．このとき，任意の木 $\xi \in T_{\Sigma_1}$ に対して，$Ins_\xi(M_1, M_2) = (M'_{1,\xi}, M'_{2,\xi})$ が存在し，

$$\mathbf{T}_{M'_{1,\xi} \fatsemi M'_{2,\xi}} = \mathbf{T}_{M'_{1,\xi}} \circ \mathbf{T}_{M'_{2,\xi}}.$$

以上の 6.12,6.13,6.14より，次が成立する．

定理 6.15. 任意の ASTT M_1, M_2 に対して，それらの合成が存在するとき，

$$\mathbf{T}_{M_1 \fatsemi M_2} = \mathbf{T}_{M_1} \circ \mathbf{T}_{M_2}.$$

証明. M_1 を ASTT とし，その入力ランク付き文字集合を Σ_1 とする．このとき，任意の木 $\xi \in T_{\Sigma_1}$ に対して，

$$\begin{aligned}
\mathbf{T}_{M_1 \fatsemi M_2}(\xi) &= \mathbf{T}_{M'_{1,\xi} \fatsemi M'_{2,\xi}}(\xi) && (\because 6.13)\\
&= (\mathbf{T}_{M'_{1,\xi}} \circ \mathbf{T}_{M'_{2,\xi}})(\xi) && (\because 6.14)\\
&= (\mathbf{T}_{M_1} \circ \mathbf{T}_{M_2})(\xi) && (\because 6.12)
\end{aligned}$$

□

定義 6.3 より，$M_1 \fatsemi M_2$ が定義されるのは，M_2 が継承属性を持たないか，M_1 が SUR を満たすときなので，次が即座に導かれる．

系 6.16.

1. $ASTT_{su} \circ ASTT = ASTT$.
2. $ASTT \circ STT = ASTT$.

7　関連研究

本章では，本研究に関連する 2 つの既存研究について紹介し，本研究と比較する．

7.1　条件分岐付き属性文法

ASTT はガード式を持つパターンマッチを扱う再帰プログラムを表現できる．パターンマッチにおけるガード式は if 式に置き換えることもできる．**条件分岐付き属性文法**（*conditional attribute grammar*, CAG）[2] は，属性文法（*attribute grammar*, AG）[14] を拡張し，if 式を扱えるようにしたものである．AG とは，文脈自由言語の意味論を定める形式手法であり，それを，構文木を変換する計算のモデルとして一般化した**属性カップリング**（AC）[11, 12] を構文的に合成するアルゴリズムである記述的合成（DC）[10, 11] が考案された．ATT は実は AC を木変換器として形式化したものであり，ATT の合成アルゴリズムは DC に基づいている．

AC は ATT と同様な規則を持っている．例えば，M_{abs} の規則

$$a_1(\pi) \overset{\sharp}{\to} b_1(\pi 1) \qquad a_1(\pi) \overset{\mathrm{bin}}{\to} b_1(\pi 1)$$
$$b_1(\pi 1) \overset{\sharp}{\to} \epsilon \qquad b_1(\pi 1) \overset{\mathrm{bin}}{\to} a_1(\pi 2)$$
$$a_1(\pi) \overset{\mathrm{A}}{\to} \mathrm{A}(b_1(\pi)) \qquad b_1(\pi 2) \overset{\mathrm{bin}}{\to} b_1(\pi)$$

は AC の規則で次のように表現できる

$$\begin{aligned} \sharp \to \mathrm{bin}: \\ \sharp.a_1 &= \mathrm{bin}.a_1 \\ \mathrm{bin}.b_1 &= \epsilon \\ \mathrm{A} \to \varepsilon: \\ \mathrm{A}.a_1 &= \mathrm{A}(\mathrm{A}.b_1) \end{aligned} \qquad \begin{aligned} \mathrm{bin} \to \mathrm{A}_1\mathrm{A}_2: \\ \mathrm{bin}.a_1 &= \mathrm{A}_1.a_1 \\ \mathrm{A}_1.b_1 &= \mathrm{A}_2.a_1 \\ \mathrm{A}_2.b_1 &= \mathrm{bin}.b_1 \end{aligned}$$

ここで，簡単のため，葉節点のラベルは A のみに限定している．AC では，属性とは木のラベルに関連付けられた変数であり，等式によって左辺の変数の値を右辺の式の値で定義する．$X_0 \to X_1X_2\cdots X_n:$ という形の記述によって，X_0 でラベル付けされた節点に到達するときに使用する規則を，コロン以下にまとめて記述することを意味する．$X_1X_2\cdots X_n$ は子節点のラベルの列である．子節点がない場合は空列を記述する．$X.a = e$ で，記号 X に紐付いた属性 a の値を式 e の値で定義することを意味する．

CAG では，図 7.1(a) の形で if 式の then 節と else 節に規則を記述することで，条件式 c の真偽に応じて規則を選択する．if 式は入れ子にすることができる．ただし，その場合に出現する複数の条件式 c の真偽は，真偽の判断を行う計算の順序によらないから，条件式の出現を，

$$
\begin{aligned}
&X_0 \to X_1 X_2 \cdots X_n : \\
&\qquad \text{if } c \text{ then } X_0.a_0 = e_0 \\
&\qquad\qquad \vdots \\
&\qquad\qquad X_0.a_m = e_m \\
&\qquad \text{else } X_0.a_0 = e'_0 \\
&\qquad\qquad \vdots \\
&\qquad\qquad X_0.a_m = e'_m
\end{aligned}
$$

(a) CAG における規則の記述

$$
\begin{aligned}
&\text{if } c_0 \text{ then } \cdots \\
&\qquad \text{else if } c_1 \text{ then } \cdots \\
&\qquad\qquad \vdots \\
&\qquad\qquad \text{else if } c_n \text{ then } \cdots \\
&\qquad\qquad\qquad \text{else}
\end{aligned}
$$

(b) if 式を先頭に集める

図 7.1(b) のように記述全体の先頭部分に移動することができる．これによって，CAG の条件式 $c_0, \ldots, c_n$ とパターンマッチのガード式を対応させることができる．2 つの AC を組み合わせるとき，先に計算を行う AC を α_1，その後に α_1 の計算結果を受け取って計算を行う AC を α_2 とする．α_1 が生産関数，α_2 が消費関数に相当する．Boyland ら [1] は DC を拡張し，α_1 が if 式を含む CAG で表現され，α_2 が if 式を伴わない場合に，α_1 と α_2 の合成が可能なアルゴリズムを考案した．このアルゴリズムでは，α_1 の if 式の全ての then 節と，最後の else 節に対して通常の DC の手続きを施し，if 式の構造や条件式はほぼそのまま流用する．

STT や ASTT の合成では，$c_0, \ldots, c_n$ に相当するガードに，消費関数側の α_2 に相当する木変換器のガードを蓄えていくことで，双方の関数がガード式を扱う場合の関数融合も可能にしている．なお，消費関数側の木変換器が実質的にガードを持たない，すなわち全てのガードが常真の述語 $\top$ であるような場合には，CAG の合成と同様の手続きとなる．ただし，CAG の条件式には，到達する節点のラベルに対する条件ではなく，属性の値に対する条件を記述するので，条件分岐の用途は異なる．

7.2 記号的木変換器の合成による関数融合

D'Antoni ら [5] は，STT を記述してそれらによる木変換を実行するための領域特化言語 FAST を開発した．FAST では STT の合成アルゴリズムも実装されており，FAST 上で記述した 2 つの STT を合成することができる．D'Antoni らは，その応用例の一つとして，関数融合を挙げている．そして，整数のリストの各要素 x を $(x+5)\,\%\,26$ に置き換える関数 *map_caeser* を FAST 上の STT で記述し，それを繰り返し適用したプログラムに対して，STT の合成の機能を用いて関数融合を施した．関数は 512 回まで繰り返し適用し，そのプログラムを融合する前後でその実行時間を比較した．その結果，融合前のプログラムでは，繰り返しの回数にほぼ比例して実行時間が増加していたのに対し，融合後のプログラムでは，繰り返しが増えても実行時

間がほとんど増加しなかった．融合前のプログラムでは，繰り返しの回数だけリストの先頭から末尾まで巡回するため，繰り返すたびにその分のオーバーヘッドが生じる．しかし融合を施すと，STT をいくつ合成していても，その分だけリストの各要素について $(x + 5)\,\%\,26$ の計算を繰り返すのみで，リストの巡回は 1 度だけで済む．

STT や ASTT の合成では，ガードの積を取る処理や，ランク付き文字集合上の関数を合成する処理を行うが，この実験結果より，合成によって得られるガードや関数が新たなオーバーヘッドとして問題になることはないと思われる．従って，有限のランク付き文字集合を扱う木変換器の合成による関数融合と同様の結果が，無限のランク付き文字集合を扱う木変換器でも得られると予想できる．つまり，ATT の合成による関数融合と同様に，ASTT の合成による関数融合の有用性が期待できる．

8 おわりに

本研究では，はじめに，TDTT を STT に拡張する方法を自然に応用して，ATT を ASTT に拡張した．つまり，ランク付き文字集合上の述語や関数を用いて ATT の規則を拡張して ASTT を定義した．また，$TDTT \subsetneq ATT$ の証明と同様の方法で $STT \subsetneq ASTT$ を証明した．

次に，ASTT の合成アルゴリズムを考案した．これも STT の手法を応用し，記号的導出関係を ASTT に適応した．また，ATT の SUR を ASTT に適応した．そして，合成した ASTT を正規化してガードが表す文字の集合を細かく分割し，それらをもとに，合成前の 2 つの ASTT を ATT に符号化することで，それら 2 つの ATT の合成結果が，合成された ASTT の正規化と見かけ上同等になることを示した．これを利用して ATT の合成の正当性に帰着することで，ASTT の合成アルゴリズムの正当性を証明した．

ASTT は，蓄積引数とパターンマッチのガード式を伴う再帰プログラムのモデルであるから，ASTT の合成アルゴリズムは，そのようなプログラムに対する関数融合に応用することができる．そのためには，再帰プログラムと ASTT の相互変換を行う必要がある．これは，再帰呼び出しにおける蓄積引数に関する処理と継承属性の計算を対応させることで実現できる．ASTT では，合成属性と継承属性の規則のガードが異なることがあるが，再帰プログラムのパターンマッチでは，1 つのガード式に対して再帰呼び出しと蓄積引数の処理が同時に行われる．つまり，再帰プログラムを ASTT M に変換すると，$\|M\| = M$ であると考えられる．逆に言えば，一般の ASTT から再帰プログラムへの変換は，ASTT を正規化することで自然に行うことができると期待できる．

ATT の合成を応用した MTT の合成 [25] では，MTT と ATT との相互変換を考えず，2 つの MTT を直接合成するアルゴリズムを与えている．蓄積引数を伴う再帰プログラムの直接的なモデルである MTT を，ATT を介さずに合成することで，実装上の有用性を向上させている．さらに理論面での結果として，ATT の合成の適用範囲より広い範囲に相当する MTT の合成に成功している．STT で行われた拡張を MTT に適用し，さらに ASTT の合成を発展させることで，ASTT の合成より広い範囲の再帰プログラムに対する，ASTT を介さない関数融合を実現することが期待される．

ATT の合成の応用例として，**スタック属性付き木変換器**（SATT）[18] の合成も挙げられる．SATT は，ATT の属性をスタックが扱えるように拡張した木変換器であり，ATT による木変換の範囲より真に広い範囲の木変換を表現することができる．これについても，SATT やその合成で行った拡張を応用することができると思われる．

謝辞

本研究にあたって，終始変わらぬ御指導を賜りました中野圭介准教授，岩崎英哉教授に深く感謝致します．また，修士研究の方向性について助言をして頂いた久野靖教授に感謝致します．そして，本研究に関して積極的に議論し，本論文の執筆に関して助言や指摘をして頂いた，中野研究室の阿部和敬氏，高橋祐多氏，岩崎研究室の西山舜氏に感謝致します．また，研究生活でお世話になった研究室の皆様に感謝致します．最後に，学生生活を支えて頂いた全ての方々に感謝致します．

参考文献

[1] John Boyland and Susan L. Graham. Composing tree attributions. In *Proceedings of the 21st ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '94*, pages 375–388, New York, New York, USA, 1994. ACM Press.

[2] John Tang Boyland and John Tang. Conditional attribute grammars. *ACM Transactions on Programming Languages and Systems*, 18(1):73–108, jan 1996.

[3] R. M. Burstall and John Darlington. A Transformation System for Developing Recursive Programs. *Journal of the ACM*, 24(1):44–67, jan 1977.

[4] Duncan Coutts, Roman Leshchinskiy, and Don Stewart. Stream fusion. In *Proceedings of the 2007 ACM SIGPLAN international conference on Functional programming - ICFP '07*, volume 42, page 315, New York, New York, USA, oct 2007. ACM Press.

[5] Loris D'antoni, Margus Veanes, Benjamin Livshits, and David Molnar. FAST: A Transducer-Based Language for Tree Manipulation. *ACM Transactions on Programming Languages and Systems*, 38(1):1–32, oct 2015.

[6] Joost Engelfriet and Heiko Vogler. Macro tree transducers. *Journal of Computer and System Sciences*, 31(1):71–146, aug 1985.

[7] Zoltán Fülöp. On attributed tree transducers. *Acta Cybernetica*, 5:261–279, 1981.

[8] Zoltan Fulop and H. Vogler. *Syntax-Directed Semantics: Formal Models Based on Tree Transducers*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1998.

[9] Zoltán Fülöp and Heiko Vogler. Forward and backward application of symbolic tree transducers. *Acta Informatica*, 51(5):297–325, aug 2014.

[10] H Ganzinger. Increasing modularity and language-independency in automatically generated compilers. *Science of Computer Programming*, 3(3):223–278, dec 1983.

[11] H Ganzinger and R Giegerich. Attribute coupled grammars. In *ACM SIGPLAN Notices*, volume 19, pages 157–170, New York, New York, USA, 1984. ACM Press.

[12] Robert Giegerich. Composition and evaluation of attribute coupled grammars. *Acta Informatica*, 25(4):355–423, may 1988.

[13] Andrew Gill, John Launchbury, and Simon Peyton Jones. A short cut to deforestation. In *Proceedings of the conference on Functional programming languages and computer architecture - FPCA '93*, number Section 4, pages 223–232, New York, New York, USA, 1993. ACM Press.

[14] Donald E. Knuth. Semantics of context-free languages. *Mathematical Systems Theory*, 2(2):127–145, jun 1968.

[15] Armin Kühnemann. *Berechnungsstärken von Teilklassen primitiv-rekursiver Programmschemata*. PhD thesis, Technischen Universität Dresden, 1997.

[16] Armin Kühnemann. *Attribute Grammars and Program Optimization: Summer Colloquium Waseda University, Tokyo, Japan July 25-August 2, 2000*. School of Science and Engineering, Institute for Software Production Technology, Waseda University, 2001.

[17] John Launchbury and Tim Sheard. Warm fusion. In *Proceedings of the seventh international conference on Functional programming languages and computer architecture - FPCA '95*, pages 314–323, New York, New York, USA, 1995. ACM Press.

[18] Keisuke Nakano. Composing Stack-Attributed Tree Transducers. *Theory of Computing Systems*, 44(1):1–38, jan 2009.

[19] Atsushi Ohori and Isao Sasano. Lightweight Fusion by Fixed Point Promotion. In *Proceedings of the 34th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '07, pages 143–154, New York, NY, USA, 2007. ACM.

[20] William C. Rounds. Mappings and grammars on trees. *Mathematical systems theory*, 4(3):257–287, Sep 1970.

[21] M. H. Sørensen, R. Glück, and N D Jones. A positive supercompiler. *Journal of Functional Programming*, 6(06):811–838, nov 1996.

[22] James W. Thatcher. Generalized2 sequential machine maps. *Journal of Computer and System Sciences*, 4(4):339–367, 1970.

[23] Valentin F. Turchin. The concept of a supercompiler. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 8(3):292, jun 1986.

[24] Margus Veanes and Nikolaj Bjørner. Symbolic Tree Transducers. In *Perspectives of Systems Informatics 2011*, pages 377–393. Springer, Berlin, Heidelberg, 2012.

[25] Janis Voigtländer and Armin Kühnemann. Composition of functions with accumulating parameters. *Journal of functional Programming*, 14(3):317–363, may 2004.

[26] Philip Wadler. Deforestation: transforming programs to eliminate trees. *Theoretical Computer Science*, 73(2):231–248, 1990.