等号否定入り文字列制約のStreaming String Transducerを用いた充足可能性判定

東京工業大学 情報理工学院 数理・計算科学系

学籍番号 16B12886

福田大我

指導教員 南出靖彦 教授

令和2年 2月28日
学士論文

概要

文字列制約は主にセキュリティ, ウェブプログラミングの分野においてクロスサイトスクリプティング (XSS) などの脆弱性を検出することに利用される. 文字列制約を解く solver の一つとして Zhu [8] による文字列制約を Streaming String Transducer に変換し, 出力の Parikh Image をとることによって解くものがあるが, これは文字列の等号否定については対応していない. 本論文では Zhu による方法を元に文字列の等号否定入りの制約についても解けるように拡張した. 等号否定を処理する具体的な方法については, 加賀江による Streaming String Transducer の等価性判定の手法 [9] を用いて行い, 出力の Parikh Image を表す Presburger Formula を構成することによって全体の文字列制約を解いた.

目次

第 1 章

序論

プログラミング言語において文字列は基本的なデータ型の一つであり文字列変数が多用される一方で, 文字列の操作はその作用の正当性を決めることが難しく, 意図しない動作を含むことがある. 文字列操作を解析する方法の一つに文字列制約を利用したものがあり, 昨今多くの論文が発表されている. 連接と置換を含むような文字列制約は形を制限しなければポストの対応問題 (PCP) に帰着できてしまい決定不能であるが, straight-line 文字列制約という条件を満たすものについては決定可能である [5]. SLOTH, OSTRICH などの文字列制約のソルバは以下の三つの形で記述された文字列制約に対して制約を満たす文字列が存在するかどうかを判定し, もし存在するなら一つ解を与えるものである.

(1) straight-line な文字列変数の連接, トランスダクションにより各変数の条件が表されている. すなわち, $x_i = w, x_i = x_j, x_i = x_j \cdot x_k$ または $x_i = T(x_j)$ の形で各変数の制約が与えられる. ここで w はある文字列, $j, k < i$ であり, T はトランスダクションである.

(2) 各変数がある正規表現に含まれているか. すなわち, $x_i \in R_i$ の形で各変数の制約が与えられる. R_i は正規言語である.

(3) 文字列変数の長さがある Presburger 算術で表された式を満たすか.

Holik らによるソルバ SLOTH では straight-line な文字列制約から連接を除去した AC という形式に変換し, alternating finite-state automata(AFA) の空判定により解く [4]. Chen らによるソルバ OSTRICH では文字列制約を実用上問題ない範囲のある条件を満たすようなものに限定しているが, SLOTH より早く, より多くのトランスダクションに対応しており, 拡張性も備えている. 一方で, 整数変数を含むような場合に関しては条件を満たさないような場合もある [2]. Zhu によるソルバ [8] では, (1), (2) を Alur らにより提案された Streaming String Transducer(SST) [1] に変換し, SST の出力の Parikh Image をトランスデューサで与えることによって, (3) を満たすような出力が存在するかを SMT ソルバ Z3 [3] を用いて解いている. しかし, このソルバは文字列の等号否定 ($x_i \neq x_j$) を含むような文字列制約には対応していなかった. 本論文では, Zhu によるソルバを文字列の等号否定を解けるように拡張した. 具体的な方法としては加賀江による関数的 SST の等価性判定 [9] を (1), (2) から構成した SST に対して利用した. 関数的 SST の等価性はある入力文字列によって,

i 出力文字列の長さが異なる

ii 出力文字列の p 文字目がそれぞれ相異なる文字になる

のいずれの条件も満たさなければ良い. Zhu によるソルバで構成される SST は決定性であるから, i , ii のい

ずれかの条件を満たすかを調べることにより, 等号否定 ($x_i \neq x_j$) を含む文字列制約を解くことができる．i は (3) の制約として解くことができ，ii はある文字 σ で終わるような出力文字列の部分文字列を出力する非決定性 SST を構成し, その出力の Parikh Image をとることで文字列長の一致可能性問題に帰着することによって解くことができる．また, Zhu によるソルバでは Parikh Image を半線形集合として計算していたが，[7] の定理 4 に示されているような CFG の Parikh Image を表す Presburger formula の構成法を応用してトランスデューサの Parikh Image を表すような Presburger Formula を構成することにより計算し, Presburger Formula を Z3 によって解くことにより文字列制約を解いた.

第 2 章

準備

定義 2.0.1. (モノイド)
集合 M と二項演算 $\cdot$ について以下の条件を満たすとき, $(M, \cdot)$ をモノイドという.

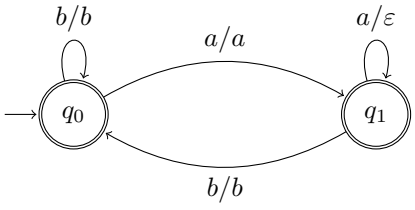
- (結合則) $\forall m_1, m_2, m_3 \in M.\ m_1 \cdot (m_2 \cdot m_3) = (m_1 \cdot m_2) \cdot m_3$
- (単位元) $\exists_1 \mathbb{1} \in M.\ \forall m \in M.\ m \cdot \mathbb{1} = \mathbb{1} \cdot m = m$

また, $\forall m_1, m_2 \in M.\ m_1 \cdot m_2 = m_2 \cdot m_1$ を満たすとき, 可換モノイドという.

定義 2.0.2. (半線形集合)
$(\mathbb{Z}^n, +)$ を可換モノイド, $\{v_1, \ldots, v_m\} \subseteq M$ $(n \geq 1), v_0 \in M$ とする. このとき, $\{v_0 + \lambda_1 v_1 + \cdots + \lambda_m v_m \mid \lambda_1, \ldots, \lambda_m \in \mathbb{N}\}$ を線形集合という. $V_1, V_2, \ldots, V_k$ を線形集合とする. このとき, $\cup_{i=1}^{k} V_i$ を半線形集合という.

定義 2.0.3. (トランスデューサ)
非決定性トランスデューサ T を $\langle \Sigma, \Gamma, Q, \delta, Q_0, F \rangle$ で定義する. Σ は入力文字列, Γ は出力文字列, Q を状態の集合, $Q_0 \subseteq Q$ で開始状態の集合, $F \subseteq Q$ で受理状態の集合を表し, $\delta \subseteq Q \times \Sigma \to Q \times \Gamma$ は遷移関数を表す. $[\![T]\!]$ で T の認識するトランスダクションを表す. すなわち, $w \in \Sigma^*$ に対して $[\![T]\!](w)$ は出力文字列の集合を表す.

以下は $(a \cup b)^*$ を入力として受け取り, 連続した a を一つの a で置き換えるトランスデューサの例である. 例えば, $[\![T]\!](aaabbbab) = abbab$ である.



定義 2.0.4. (SST)
(deterministic) Streaming String Transducer (SST)S を $\langle \Sigma, \Gamma, Q, X, \delta, \eta, q_0, F \rangle$ で定義する. Σ は入力文字列, Γ は出力文字列, Q を状態の集合, X は文字列変数の集合, q_0 は開始状態である. $\delta \subseteq Q \times \Sigma \to Q$ は遷移関数, $\eta \subseteq Q \times \Sigma \to M_{X,\Gamma}$ は変数更新関数である. $F \subseteq Q \hookrightarrow (X \cup \Gamma)^*$ は出力関数で, 受理状態に対して出力

文字列を返す. ここで, $M_{X,\Gamma}$ は変数 $x \in X$ に対して X と Γ の文字列を返す関数 $\alpha : X \to (X \cup \Gamma)^*$ の集合である. $M_{X,\Gamma}$ は関数の合成 $\circ$ と恒等写像 $\mathbb{1}_{X,\Gamma}(x) = x$ についてモノイドをなす. SST の意味 $[\![S]\!]$ を以下のように定義する. $q_0 \xrightarrow{w/\alpha} q_f,\ q_f \in dom(F)$ のとき、

$$[\![S]\!](w) = \hat{\varepsilon}(\alpha(F(q_f)))$$

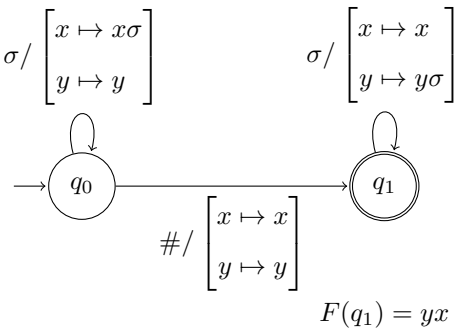
ここで、$\hat{\varepsilon} \in (\Gamma \cup X)^* \to \Gamma^*$ は変数を空文字列で置き換える関数である.

文字列 w 中の文字 $c \in (X \cup \Gamma)$ の数を $|w|_c$ と書くことにする. $\alpha \in M_{X,\Gamma}$ に対し, $\underset{x \in X}{max}|\alpha|_x \leq K (K \in \mathbb{N})$ であるとき, α は K-copy であるという.

$\hat{\eta}$ を $\hat{\eta}(q, w) = \begin{cases} \mathbb{1}_{X,\Gamma}(x) & (w = \epsilon) \\ \eta(q, \sigma) \circ \hat{\eta}(\delta(q, \sigma), w') & (w = \sigma w') \end{cases}$ として, 文字列に対して変数更新関数を返すよう η を拡張したものとする. 任意の文字列 $w \in \Sigma^*$ に対し, $\hat{\eta}(q_0, w)$ が K-copy であるとき, SSTS は K-bounded copy であるという.

任意の入力文字列 w に対して, $[\![S]\!](w) = w'$ なる出力が唯一つに決まるような SST を関数的 SST という. 関数的 SST については等価性判定が可能であり, 具体的な手法が [9] により与えられている.

以下は入力 $w_0 \# w_1$ に対し $w_1 w_0$ を出力する SST の $ab\#ba$ での動作である. 動作のわかりやすさのため, 状態, 文字列変数の中身を $\langle q, [x, y] \rangle$ の組みで表す.



- $\langle q, [x, y] \rangle = \langle q_0, [\varepsilon, \varepsilon] \rangle$ から開始.
- $\langle q_0, [\varepsilon, \varepsilon] \rangle \xrightarrow{a} \langle q_0, [a, \varepsilon] \rangle$
- $\langle q_0, [a, \varepsilon] \rangle \xrightarrow{b} \langle q_0, [ab, \varepsilon] \rangle$
- $\langle q_0, [ab, \varepsilon] \rangle \xrightarrow{\#} \langle q_1, [ab, \varepsilon] \rangle$
- $\langle q_1, [ab, \varepsilon] \rangle \xrightarrow{b} \langle q_1, [ab, b] \rangle$
- $\langle q_1, [ab, b] \rangle \xrightarrow{a} \langle q_1, [ab, ba] \rangle$
- $F(q_1) = yx$ であるから出力は $baab$ となる.

定義 2.0.5. 文字列 $w \in \Gamma^*$ に対して, Parikh Image $\Psi(w) \in \mathbb{N}^\Gamma$ を

$$\Psi(w)_\gamma = (\text{w 中の } \gamma \text{ の数})$$

とする. 通常の意味での Parikh Image に加えて, transition の列 $t \in \delta^*$ に対し, $\Psi_\delta(t) \in \mathbb{N}^\delta$ を $(\Psi(t)_\delta)_d = (t$ 中の d の適用回数), $\Psi_Q(t) \in \mathbb{N}^Q$ を $(\Psi_Q(t))_q = (t$ 中の q の出現数) として定義する.

定義 2.0.6. 以下のような命題 ϕ を existential Presburger Formula φ という.

$$\begin{cases} t \coloneqq 0 \mid 1 \mid x \mid t_1 + t_2 \\ \phi \coloneqq t_1 = t_2 \mid t_1 < t_2 \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \exists x.\ \phi \end{cases}$$

2.1 文字列制約

$X = \{x_0, x_1, \ldots, x_{n-1}\}$ を文字列変数の集合とする.

定義 2.1.1. $w \in \Sigma^*$ をある文字列, $j, k < i$, T をトランスデューサとする. 以下のような e_i について, $x_i = e_i$ を原子制約という.

$$e_i \coloneqq w \mid x_j \mid x_j \cdot x_k \mid T(x_j)$$

$m < n$ に対して,

$$\varphi_{sl} \coloneqq (x_m = e_m) \wedge (x_{m+1} = e_{m+1}) \wedge \cdots \wedge (x_{n-1} = e_{n-1})$$

なる φ_{sl} を基礎直線制約という.

定義 2.1.2. $R_i (0 \leq i \leq n-1)$ を正規言語とする. このとき,

$$\varphi_{reg} \coloneqq \bigwedge (x_i \in R_i)$$

なる φ_{reg} を正規制約という.

定義 2.1.3. $c \in \mathbb{N}$ を定数, $u \in \mathbb{Z}$ を変数, $|x|$ を文字列 x の長さとする. このとき,

$$t \coloneqq c \mid u \mid |x| \mid t + t \mid t - t$$

を整数表現という. また, 整数制約 φ_{int} を

$$\varphi_{int} \coloneqq t = t \mid t < t \mid \varphi \wedge \varphi \mid \neg\varphi$$

とする.

以上の制約を合わせた直線制約 φ を以下で定義する.

$$\varphi \coloneqq \varphi_{sl} \wedge \varphi_{reg} \wedge \varphi_{int}$$

定義 2.1.4. 文字列変数の集合 X に対して, 文字列変数の割り当て θ_{str} を

$$\theta_{str} \coloneqq [x_0 \to w_0, x_1 \to w_1, \ldots, x_{n-1} \to w_{n-1}]$$

として, $\theta_{str}(\gamma) = \gamma$ $(\gamma \in \Gamma)$, $\theta_{str}(x_i) = w_i$ $(x_i \in X)$ とする. また, $\theta_{str}(x_j \cdot x_k) = \theta_{str}(x_j) \cdot \theta_{str}(x_k)$, $\theta_{str}(T(x_j)) = T(\theta(x_j))$, 整数変数に対する割り当てを θ_{int} とする. このとき, $\theta = \theta_{str} \cup \theta_{int}$ を以下のように定義する.

$$\begin{aligned} \theta(c) &= c \ (c \in \mathbb{Z}), \\ \theta(|x_i|) &= |\theta(x_i)| \ (x_i \in X), \\ \theta(t_1 \pm t_2) &= \theta(t_1) \pm \theta(t_2) \end{aligned}$$

とする.

定義 2.1.5. 変数割り当て θ に対し, 制約 φ の真偽値を val_θ により以下のように定める.

$$\begin{aligned} val_\theta(\varphi_1 \wedge \varphi_2) &= val_\theta(\varphi_1) \wedge val_\theta(\varphi_2), \\ val_\theta(\varphi_1 \vee \varphi_2) &= val_\theta(\varphi_1) \vee val_\theta(\varphi_2), \\ val_\theta(x = e) &= (\theta(x) = \theta(e)), \\ val_\theta(x \in R) &= (\theta(x) \in R), \\ val_\theta(t_1 = t_2) &= (\theta(t_1) = \theta(t_2)), \\ val_\theta(t_1 < t_2) &= (\theta(t_1) < \theta(t_2)), \end{aligned}$$

制約 φ に対して, $val_\theta(\varphi) =$ 真となるような θ が存在するとき, φ は充足可能という. そのような θ が存在しないとき, φ は充足不能という.

定義 2.1.6. 文字列変数の集合 X に対して, 変数の割り当て $\theta = [x_0 \to w_0, x_1 \to w_1, \ldots, x_{n-1} \to w_{n-1}]$ が制約 φ を充足可能なとき, $\theta \models \varphi$ とする.

(基礎直線制約でない制約を含むような) 一般の直線制約については決定不可能である. これは $x = T(x)$ なる制約が書けてしまうことで PCP に帰着できるためである [6]. 一方で, 直線文字列制約については決定可能である [5].

以下は直線文字列制約の例で, $x_0 = aba, x1 = ba, x_2 = ababa, x_3 = cba$ が充足可能な解の一つになる.

$$
\begin{aligned}
x_2 &= x_0 \cdot x_1 \\
x_3 &= x_2.replaceAll(aba, c) \\
x_0 &\in (a \cup b)^* \\
|x_2| &= 5
\end{aligned}
$$

第 3 章

出力文字列の Parikh image を出力するトランスデューサの構成

bounded-copy SST $S = \langle \Sigma, \Gamma, Q, X, \delta, \eta, q_0, F \rangle$ に対し, 入力 w のとき, $|S(w)|$ を出力とするようなトランスデューサ $T = \langle \Sigma, \mathbb{N}, Q^A, \delta^A, q_0^A, F^A \rangle$ を構成することができる. ただし, $Q^A = \{q_\perp\} \cup (Q' \times \mathbb{N}^X), Q_0^A = \{(q_0, B) \mid (q_0, B) \in Q^A\}, F^A = q_\perp$ とする. $\delta^A \subseteq Q \times \Sigma \to Q \times \mathbb{N}$ に関して,

- $\forall q_f \in dom(F), F(q_f) = w_f$ のとき, $(q_f, \Psi_X(w_f)) \xrightarrow{\varepsilon/|\Psi_\Gamma(w_f)|} q_\perp \in \delta^A$
- $\forall (q, B) \in Q^A, \delta(q', \sigma) = q, \eta(q', \sigma) = \alpha$ のとき, $(q', \Psi'_X(\alpha) \cdot B) \xrightarrow{\sigma/|\Psi'_\Gamma(\alpha)\cdot B|} (q, B) \in \delta^A$

である.

ここで, Ψ_X, Ψ_Γ はそれぞれ $(X \cup \Gamma)^* \to \mathbb{N}^X, (X \cup \Gamma)^* \to \mathbb{N}^\Gamma$ で, X, Γ に関する入力 v の parikh image を表す. ($\Psi_X(v)$ の x_i には $|v|_{x_i}$ が入っている.) Ψ'_X, Ψ'_Γ はそれぞれ $M_{X,\Gamma} \to \mathbb{N}^{X\times X}, M_{X,\Gamma} \to \mathbb{N}^{X\times\Gamma}$ で, X, Γ に関する, $\alpha(x)$ の parikh image を表す ($\Psi_X(\alpha)$ の x_i, x_j には $|\alpha(x_j)|_{x_i}$ が入っている). Q^A は S が bounded-copy であるために有限となる. 以下では, このように構成されたトランスデューサが実際に文字列長を出力することを示す. また, S が deterministic か non-deterministic かに関わらず同様の構成法で良いため, non-deterministic な場合, すなわち, $S(w) = \{w_0, w_1, \ldots, w_n\}$ のときに, $T(w) = \{|w_0|, |w_1|, \ldots, |w_n|\}$ を示す.

補題 3.0.1. $\forall \alpha, \beta \in M_{X,\Gamma}$ に対して,

$$\begin{cases} \Psi'_X(\alpha \circ \beta) & = \Psi'_X(\alpha) \cdot \Psi'_X(\beta) \\ \Psi'_\Gamma(\alpha \circ \beta) & = \Psi'_\Gamma(\alpha) \cdot \Psi'_\Gamma(\beta) + \Psi'_\Gamma(\beta) \end{cases}$$

証明. $\Psi'_X(\beta)$ の $y \in X$ に対応する列には, $\beta(y)$ に含まれる各変数の parikh image が入っている. $\Psi'_X(\alpha)$ の $x \in X$ に対応する列には, 各 $z \in X$ に対する $\alpha(z)$ 中の x の個数が入っている. $\Psi'_X(\alpha)$ の x に対応する行と $\Psi'_X(\beta)$ の y に対応する列の積は, $\Psi'_X(\alpha \circ \beta(y))$ の x の個数を表す. これは, $\Psi'_X(\alpha \circ \beta)$ の x, y 成分に他ならない. 故に, 一つ目の式が成り立つ.
同様にして二つ目の式も示すことができる. □

補題 3.0.2. $q \xrightarrow[S]{w/\alpha} q'$ であるとき, $(q', B) \in (Q', \mathbb{N}^X)$ に対して $(q, \Psi'_X(\alpha) \cdot B) \xrightarrow[T]{w/|\Psi_\Gamma(\alpha)\cdot B|} (q', B)$.

証明. w の長さに関する帰納法で示す. $|w| = 0$ のとき, $(q, B) \xrightarrow[T]{\varepsilon/\mathbb{0}} (q, B)$ は明らかに成り立つ. $(q', B) \in$

$(Q', \mathbb{N}^X)$ に対して, $q \xrightarrow[S]{\varepsilon/\alpha} q'$ であるときは, T の構成法より, $(q, \Psi'_X(\alpha) \cdot B) \xrightarrow[T]{\varepsilon/|\Psi_\Gamma(\alpha)\cdot B|} (q', B)$. $|w| = k$ のとき成り立つと仮定し, $|w| = k+1$ のときに成り立つことを示す. $q \xrightarrow[S]{\sigma/\alpha} q'' \xrightarrow[S]{w/\beta} q'$ と仮定する. $q'' \xrightarrow[S]{w/\beta} q'$ より, 帰納法の仮定から $(q'', \Psi'_X(\beta) \cdot B) \xrightarrow[T]{w/|\Psi_\Gamma(\beta)\cdot B|} (q', B)$ が成り立つ. また, $q \xrightarrow[S]{\sigma/\alpha} q''$ より, T の構成法から $\forall (q'', \Psi'_X(\beta) \cdot B) \in Q^A$ について $(q, \Psi'_X(\alpha) \cdot \Psi'_X(\beta) \cdot B) \xrightarrow{\sigma/|\Psi'_\Gamma(\alpha)\cdot\Psi'_X(\beta)\cdot B|} (q'', \Psi'_X(\beta) \cdot B) \in \delta^A$ である. よって, $(q, \Psi'_X(\alpha) \cdot \Psi'_X(\beta) \cdot B) \xrightarrow{\sigma/|\Psi'_\Gamma(\alpha)\cdot\Psi'_X(\beta)\cdot B|+|\Psi_\Gamma(\beta)\cdot B|} (q', B)$ が成り立つ. 補題 3.0.1 より, $(q, \Psi'_X(\alpha \circ \beta) \cdot B) \xrightarrow{\sigma w/|\Psi'_\Gamma(\alpha\circ\beta)\cdot B|} (q', B)$ が成り立つ. □

定理 3.0.1. $q \xrightarrow{w/\alpha} q_f, F(q_f) = w_f$ であるとき, $(q, \Psi_X(\alpha) \cdot \Psi_X(w_f)) \xrightarrow{w/|\Psi_\Gamma(\alpha)\cdot\Psi_X(w_f)+\Psi_\Gamma(w_f)|} q_\perp$.

補題 3.0.2 により示すことができる. 同様にして以下も示すことができる.

定理 3.0.2. $(q, B) \xrightarrow{w/n} q_\perp$ であるとき, $\exists q_f \in dom(F).\ q \xrightarrow{w/\alpha} q_f, n = |\Psi_\Gamma(\alpha) \cdot \Psi_X(w_f) + \Psi_\Gamma(w_f)|, B = \Psi_X(\alpha) \cdot \Psi_X(w_f), w_f = F(q_f)$ が成り立つ.

定理 3.0.1, 3.0.2 より, 次が成り立つ.

定理 3.0.3. $S(w) \ni w'$ であるとき, $T(w) \ni |w'|$.

定理 3.0.4. $T(w) \ni n$ であるとき, ある文字列 w' が存在して, $S(w) \ni w', |w'| = n$ を満たす.

第 4 章

部分文字列を出力する SST の構成

4.1　部分文字列を出力する SST の構成法

決定性 bounded-copy SST $S = \langle \Sigma, \Gamma, Q, X, \delta, q_0, F \rangle$ として $\gamma \in \Gamma$ をとる. $w \in \Sigma^*$ に対し, $S_\gamma(w) = \{w_L\gamma \mid \exists w_R \in (X \cup \Gamma)^*.\ [\![S]\!](w) = w_L\gamma w_R\}$ となるような非決定性 SSTS_γ の構成法を以下に示す.

$\langle \Sigma, \Gamma, Q^A, X^A, q_0^A, F^A, \Delta \rangle$ の各要素を以下のように定義すると、求めたい SST と一致する.
$Q^A \subseteq Q \times (\{\bot\} \cup X)$ (Q^A は右辺全てとしてしまっても良いが, 実際に全要素が q_0^A から到達可能とは限らない), $X^A = X \cup \{x_0\}$(ただし, $x_0 \notin X$), $q_0^A = (q_0, \bot)$, $F(q_f) = w$ のとき, $F^A((q_f, \bot)) = \{w_L\gamma \mid \exists w_R \in (X \cup \Gamma)^*.\ w_L\gamma w_R = w\}$, $F^A((q_f, x)) = \{w_L x_0 \mid \exists w_R \in (X \cup \Gamma)^*.\ w_L x w_R = w\}$.
$\Delta \subseteq Q \times \Sigma \to Q \times M_{X^A,\Gamma}$ は以下のような三つの遷移からなる.

- $q \xrightarrow{\sigma/\alpha} q' \in \delta$ であり, ある w_L, w_R が存在して, $\alpha(x') = w_L x w_R$ であるとき,

$$(q, x) \xrightarrow{\sigma/\alpha'} (q', x') \in \Delta \quad \text{ただし, } \alpha'(z) = \begin{cases} w_L x_0 & z = x_0 \\ \alpha(z) & z \in X \end{cases}$$

- $q \xrightarrow{\sigma/\alpha} q' \in \delta$ であり, ある w_L, w_R が存在して, $\alpha(x) = w_L b w_R$ であるとき,

$$(q, \bot) \xrightarrow{\sigma/\alpha'} (q', x) \in \Delta \quad \text{ただし, } \alpha'(z) = \begin{cases} w_L\gamma & z = x_0 \\ \alpha(z) & z \in X \end{cases}$$

- $q \xrightarrow{\sigma/\alpha} q' \in \delta$ であるとき,

$$(q, \bot) \xrightarrow{\sigma/\alpha'} (q', \bot) \in \Delta \quad \text{ただし, } \alpha'(z) = \begin{cases} x_0 & z = x_0 \\ \alpha(z) & z \in X \end{cases}$$

4.2　上記の SST が部分文字列を出力することの証明

S_γ が $S_\gamma(w) = \{w_L\gamma \mid \exists w_R \in (X \cup \Gamma)^*.\ S(w) = w_L\gamma w_R\}$ であることを示すためにいくつかの補題を示す.

補題 4.2.1. $(q, x) \xrightarrow{w/\alpha'} (q', y)$, $(x, y \in X)$ であるとき, ある $w_L \in (X \cup \Gamma)^*$ が存在して, $\alpha'(x_0) = w_L x_0$ が成り立つ.

証明. w の長さに関する帰納法で示す. $|w| = 0$ のとき, $(q, x) \overset{\epsilon/\mathbb{1}_{M_{X^A,\Gamma}}}{\longrightarrow} (q', y)$ なる Δ の遷移は存在しないので満たす. $|w| = k$ で成立すると仮定し, $|w| = k+1$ で成り立つことを示す. $(q, x) \overset{w/\alpha'}{\longrightarrow} (q'', u) \overset{\sigma/\beta'}{\longrightarrow} (q', y)$ であるとする. 帰納法の仮定より, $\alpha'(x_0) = w_L x_0$ が成り立つ. ここで, $(q'', u) \overset{\sigma/\beta'}{\longrightarrow} (q', x)$ であり, S_γ の構成法からある文字列 w'_L, w'_R について $\beta(x) = w'_L u w'_R$, $\beta'(x_0) = w'_L x_0$ を満たす. よって, $\alpha' \circ \beta'(x) = (\alpha'(w'_L x_0)) = \alpha'(w'_L) w_L x_0$ である. □

同様にして以下も成り立つ. 証明は省略する.

補題 4.2.2. $(q, \bot) \overset{w/\alpha'}{\longrightarrow} (q', \bot)$ であるとき, $\alpha'(x_0) = x_0$ が成り立つ.

補題 4.2.3. $(q_0, \bot) \overset{w/\alpha'}{\longrightarrow} (q', x), (x \in X)$ であるとき, ある $w_L \in (X \cup \Gamma)^*$ が存在して $\alpha'(x_0) = w_L \gamma$ が成り立つ.

証明. ある文字列 $w_1, w_2 \in (X \cup \Gamma)^*$ と $\sigma \in \Sigma$ が存在して, $w = w_1 \sigma w_2$, $(q_0, \bot) \overset{w_1/\alpha'}{\longrightarrow} (q_1, \bot) \overset{\sigma/\beta'}{\longrightarrow} (q_2, y) \overset{w_2/\gamma'}{\longrightarrow} (q, x)$ としてよい. 補題 4.2.1, 4.2.2 よりある文字列 w_L について $\alpha'(x_0) = x_0, \gamma'(x_0) = w_L x_0$ が成り立つ. S_γ の構成法より, $\beta'(x_0) = w'_L \gamma$ が成り立つ. よって,

$$\begin{aligned}(\alpha' \circ \beta') \circ \gamma'(x_0) &= \alpha' \circ \beta'(w_L x_0) \\ &= \alpha'(\beta'(w_L) w'_L \gamma) \\ &= \alpha' \circ \beta'(w_L) \alpha'(w'_L) \gamma\end{aligned}$$

であるから成り立つ. □

補題 4.2.4. $q \overset{w/\alpha}{\longrightarrow} q'$ かつ $\exists w_L, w_R.\ \alpha(y) = w_L x w_R$ であれば,

$$(q, x) \overset{w/\alpha'}{\longrightarrow} (q', y) \quad \text{ただし, } \alpha'(z) = \begin{cases} w_L x_0 & z = x_0 \\ \alpha(z) & z \in X \end{cases}$$

証明. 入力文字列 w の長さに関する帰納法で示す. $|w| = 0$ のとき, $q \overset{\epsilon/\mathbb{1}_{M_{X,\Gamma}}}{\longrightarrow} q$ であり, 任意の $x \in X$ に対して, $(q, x) \overset{\epsilon/\mathbb{1}_{M_{X^A,\Gamma}}}{\longrightarrow} (q, x)$ を満たす. $|w| = k$ で成り立つと仮定し, $|w| = k+1$ でも成り立つ事を示す. $q \overset{\sigma/\alpha}{\longrightarrow} q'' \overset{w/\beta}{\longrightarrow} q'$ かつある $x, u, y \in X$ に対して $\alpha(u) = w_L x w_R$, $\beta(y) = w'_L u w'_R$ であるとする. このとき,

$$\begin{aligned}\alpha \circ \beta(y) &= \alpha(w'_L u w'_R) \\ &= \alpha(w'_L) w_L x w_R \alpha(w'_R)\end{aligned}$$

を満たす. S_γ の構成法から, $(q, x) \overset{\sigma/\alpha'}{\longrightarrow} (q'', u)$. 帰納法の仮定より, $(q'', u) \overset{w/\beta'}{\longrightarrow} (q', y)$ である. ただし, $\alpha'(x_0) = w_L x_0$, $\beta'(x_0) = w'_L x_0$. 故に, $(q, x) \overset{\sigma w/\alpha\circ\beta}{\longrightarrow} (q', y)$ であり,

$$\begin{aligned}\alpha' \circ \beta'(x_0) &= \alpha'(w'_L x_0) \\ &= \alpha(w') w_L x_0\end{aligned}$$

を満たす. □

この補題は逆に当たるものも成り立つ.

補題 4.2.5. $(q, x) \overset{w/\alpha'}{\longrightarrow} (q', y)$ であるとき,

$$q \xrightarrow{w/\alpha} q' \quad \text{ただし, } \alpha \in M_{X,\Gamma} \text{ で, } \alpha(z) = \alpha'(z)\ (z \in X)$$

また, $\alpha'(x_0) = w_L x_0$ のとき, α はある $w_R \in (X \cup \Gamma)^*$ について $\alpha(y) = w_L x w_R$ を満たす.

証明. 同様に入力文字列 w の長さに関する帰納法で示すことができる. $|w| = 0$ のとき, $(q, x) \xrightarrow{\epsilon/\mathbb{1}_{M_{X^A,\Gamma}}} (q, x)$ であり, このとき $q \xrightarrow{\epsilon/\mathbb{1}_{M_{X,\Gamma}}} q$ を満たす. $|w| = k$ で成り立つと仮定し, $|w| = k+1$ でも成り立つことを示す. $(q, x) \xrightarrow{\sigma/\alpha'} (q'', u) \xrightarrow{w/\beta'} (q', y)$ であるとする. S_γ の構成法から, $(q, x) \xrightarrow{\sigma/\alpha'} (q'', u) \in \Delta$ のとき, $q \xrightarrow{\sigma/\alpha} q'' \in \delta$ であり, ある文字列 w_L, w_R に対し $\alpha(u) = w_L x w_R$ を満たす. ただし α は $\alpha(z) = \alpha'(z)\ (z \in X)$ を満たし, α' は $x \in X$ に対して $\alpha'(x) \in (X \cup \Gamma)^*$ であるから, $\alpha \in M_{X,\Gamma}$. 帰納法の仮定から, $q'' \xrightarrow{w/\beta} q'$ かつ $\beta \in M_{X,\Gamma}$, $\beta(z) = \beta'(z)\ (z \in X)$ であり, ある文字列 w'_L, w'_R に対し $\beta(y) = w'_L u w'_R$ が成り立つ. 故に, $q \xrightarrow{\sigma w/\alpha\circ\beta} q'$ である. さらに,

$$\begin{aligned} \alpha \circ \beta(z) &= \alpha'(\beta'(z)) \\ &= \alpha' \circ \beta'(z)\ (z \in X) \end{aligned}$$

を満たすから $\alpha \circ \beta \in M_{X,\Gamma}$ であり,

$$\begin{aligned} \alpha \circ \beta(y) &= \alpha(w'_L u w'_R) \\ &= \alpha(w'_L) w_L x w_R \alpha(w'_R) \end{aligned}$$

以上より, $|w| = k+1$ でも成り立つことが示された. □

同様の入力文字列 w の長さに関する帰納法により, 以下の補題も示すことができる.

補題 4.2.6. $q \xrightarrow{w/\alpha} q'$ であるとき, $(q, \bot) \xrightarrow{w/\alpha'} (q', \bot)$ かつ $\alpha'(z) = \begin{cases} x_0 & (z = x_0) \\ \alpha(z) & (z \in X) \end{cases}$

補題 4.2.7. $(q, \bot) \xrightarrow{w/\alpha'} (q', \bot)$ であるとき, $q \xrightarrow{w/\alpha} q'$ かつ $\alpha(z) = \alpha'(z)\ (z \in X)$ を満たす.

補題 4.2.8. $q \xrightarrow{w/\alpha} q'$ かつある $w_L, w_R \in (X \cup \Gamma)^*$ について $\alpha(x) = w_L \gamma w_R$ であれば,

$$(q, \bot) \xrightarrow{w/\alpha'} (q', x) \quad \text{ただし, } \alpha'(z) = \begin{cases} w_L \gamma & (z = x_0) \\ \alpha(z) & (z \in X) \end{cases}$$

補題 4.2.9. $(q, \bot) \xrightarrow{w/\alpha'} (q', x)$, $\alpha'(x_0) = w_L \gamma$ であれば, $q \xrightarrow{w/\alpha} q'$ ただし, α は $\alpha(z) = \alpha'(z)\ (z \in X)$ を満たし, ある $w_R \in (\Gamma \cup X)^*$ が存在して $\alpha(x) = w_L \gamma w_R$.

これらの補題から目的の命題を示すことができる.

定理 4.2.1. $[\![S_\gamma(w)]\!] = \{w_L \gamma \mid \exists w_R \in (\Gamma \cup X)^*.\ [\![S]\!](w) = w_L \gamma w_R\}$

証明. 見やすさのため, $\{w_L \gamma \mid \exists w_R \in (X \cup \Gamma)^*.\ S(w) = w_L \gamma w_R\} = B$ と書くことにする.

まず, $[\![S_\gamma(w)]\!] \subseteq B$ を示す. $w' \in S_\gamma(w)$, $(q_0, \bot) \xrightarrow{w/\alpha'} q_f^A$, $q_f^A \in dom(F^A)$ であるとする. $q_f^A = (q_f, \bot)$ $(q_f \in dom(F))$ のとき, 補題 4.2.7 より, S は $q_0 \xrightarrow{w/\alpha} q_f$ を満たす. w' はある $u \in F^A((q_f, \bot))$ により, $w' = \hat{\epsilon}(\alpha'(u))$ とかける. ここで, F^A の定義より, ある $w_L, w_R \in (\Gamma \cup X)^*$ に対して $u = w_L \gamma$, $F(q_f) = w_L \gamma w_R$ である. 故に, $w' = \hat{\epsilon}(\alpha'(w_L))\gamma$ であり, $[\![S]\!](w) = \hat{\epsilon}(\alpha(w_L))\gamma\hat{\epsilon}(\alpha(w_R))$ を満たす. x_0 を含

まない文字列 $v \in (X \cup \Gamma)^*$ に対して, $\alpha(v) = \alpha'(v)$ であるから $w' \in B$.
$q_f^A = (q_f, x)$, $(q_f \in dom(F))$, $\alpha'(x_0) = w'_L\gamma$ のとき, 補題 4.2.9 より, S は $q \xrightarrow{w/\alpha} q_f$ を満たし, ある $w'_R \in (\Gamma \cup X)^*$ について $\alpha(x) = w'_L\gamma w'_R$. また, F^A の定義よりある $w_L, w_R \in (X \cup \Gamma)^*$ に対して $u = w_L x_0, F(q_f) = w_L x w_R$ である. 故に, $w' = \hat{\epsilon}(\alpha'(w_L x_0)) = \hat{\epsilon}(\alpha'(w_L)w'_L)\gamma$ であり,

$$\begin{aligned} [\![S]\!](w) &= \hat{\epsilon}(\alpha(w_L x w_R)) \\ &= \hat{\epsilon}(\alpha(w_L)w'_L)\gamma\hat{\epsilon}(w'_R\alpha(w_R)) \end{aligned}$$

を満たす. よって, $w' \in B$.

次に, $B \subseteq [\![S_\gamma]\!](w)$ を示す. $w' \in B$ とする. ある文字列 $w_L, w_R \in \Gamma^*$ によって $w' = w_L\gamma, S(w) = w_L\gamma w_R$ と書ける. このとき, $q_0 \xrightarrow{w/\alpha} q_f$ $(q_f \in dom(F))$ とすると, $[\![S]\!](w) = \hat{\epsilon}(\alpha(F(q_f)))$ である. $w' \in B$ であるから, $w_L = \hat{\epsilon}(\alpha(w'_L))$, $w_R = \hat{\epsilon}(\alpha(w'_R))$, $F(q_f) = w'_L\gamma w'_R$, または, $\alpha(x) = w''_L\gamma w''_R$ $(w''_L, w''_R \in (\Gamma \cup X)^*)$ かつ $w_L = \hat{\epsilon}(\alpha(w'_L)w''_L)$, $w_R = \hat{\epsilon}(w''_R\alpha(w'_R))$, $F(q_f) = w'_L x w'_R$ のいずれかが成り立つような w'_L, w'_R が存在する. 前者が成り立つ場合, 補題 4.2.9 より $(q_0, \perp) \xrightarrow{w/\alpha'} (q_f, \perp)$ が成り立ち, F^A の定義より, $w'_L\gamma \in F^A((q_f, \perp))$ であるから, $\hat{\epsilon}(\alpha'(w'_L\gamma)) = w'$ より, $w' \in [\![S_\gamma]\!](w)$. 後者の場合, 補題 4.2.9 より, $(q_0, \perp) \xrightarrow{w/\alpha'} (q_f, x)$, $\alpha'(x_0) = w''\gamma$ が成り立ち, F^A の定義より, $w'_L x_0 \in F^A((q_f, x))$ であるから,

$$\begin{aligned} \hat{\epsilon}(\alpha'(w'_L x_0)) &= \hat{\epsilon}(\alpha(w'_L)w''_L)\gamma \\ &= w' \end{aligned}$$

より, $w' \in [\![S_\gamma]\!](w)$.
以上より, $[\![S_\gamma]\!](w) = B$ が示された. □

第 5 章

トランスデューサの出力文字列の Parikh Image を表す Presburger Formula の構成

トランスデューサ $T = \langle \Sigma, \Gamma, Q, \delta, Q_0, F \rangle$ に対し, 出力文字列の Parikh Image となりうる変数割り当てに対して真となる Presburger Formulaϕ を構成することができる. すなわち,

$$\phi(x_{\gamma_1}, \ldots, x_{\gamma_n}) = true \Leftrightarrow \exists w \in \Sigma^*.\ \exists w' \in [\![T]\!](w).\ (\Psi(w'))_{\gamma_i} = x_{\gamma_i}$$

である.

5.1 Parikh Image を表す Presburger Formula の構成法

見やすさのため以下のような記号を定義する.

定義 5.1.1. $\delta \ni d : p \xrightarrow{\sigma/\gamma} q$ に対して,

$$\begin{cases} source(d) & = p \\ target(d) & = q \\ v_d & = \Psi(\gamma) \end{cases}$$

とする.

$T = \langle \Sigma, \Gamma, Q, \delta, Q_0, F \rangle$ に対して T の出力する Parikh Image に一致する Presburger Formula を構成する.
$\gamma \in \Gamma$ に対し, 出力中の γ の数を x_γ, $d \in \delta$ に対し, 対応する transition 中の d の出現回数を $y_d \in \mathbb{N}$, $q \in Q$ に対し, q の transition 中で開始状態 q_0 からの距離 $z_q \in \mathbb{N}$, transition 中での使用回数を $n_q \in \mathbb{N}$, q が開始状態であるかを表す変数 $s_q \in \{1, 0\}$, 最終状態であるかを表す変数 $r_q \in \{1, 0\}$ とする.
ただし, z_q について transition 中に q が出現しないとき -1, 開始状態であるとき 0, それ以外の状態に関しては再帰的に定義するものとする.
$q \in Q$ に対して, $D(q) = (\sum_{\delta \ni d : target(d) = q} y_d - \sum_{\delta \ni d : source(d) = q} y_d) + (s_q - r_q)$ とする.
以下の四つの existential Presburger Formula を満たしていれば良い.

i . (Euler condition)

$\forall q \in Q$ に対し, $D(q) = 0$. すなわち,

$$\phi_1 = (\bigwedge_{q \in Q} D(q) = 0)$$

を満たしていれば良い.

ii. (Connectivity)
$\forall q \in Q$ に対して以下が成り立つ.
まず, q の出現回数について,

$$n_q = \sum_{\delta \ni d: target(d)=q} y_d + s_q \tag{5.1}$$

$$= \sum_{\delta \ni d: source(d)=q} y_d + r_q \tag{5.2}$$

である.
q が transition で到達不可能ならば $\phi_q^A = (n_q = 0) \wedge (z_q = -1)$
q が transition の開始状態ならば $(n_q > 0) \wedge ((z_q = 0) \wedge (s_q = 1))$
q が transition で到達可能ならば

$$(n_q > 0) \wedge (\bigvee_{\delta \ni d: target(d)=q} (z_q = z_{source(d)} + 1) \wedge (y_d > 0) \wedge (z_{source(d)} \geq 0))$$

故に, q が transition で使用されているならば

$$\phi_q^B = (n_q > 0) \wedge (((z_q = 0) \wedge (s_q = 1)) \vee (\bigvee_{\delta \ni d: target(d)=q} (z_q = z_{source(d)} + 1) \wedge (y_d > 0) \wedge (z_{source(d)} \geq 0)))$$

いずれかを満たすから,

$$\phi_2 = (\bigwedge_{q \in Q} (\phi_q^A \vee \phi_q^B)) \wedge (\bigwedge_{q \in Q} n_q = \sum_{\delta \ni d: source(d)=q} y_d + r_q)$$

を満たす.

iii. (Parikh Image)
$\forall \gamma \in \Gamma$ に対し, $x_\gamma = \sum_{d \in \delta} v_d(\gamma) \cdot y_d$. すなわち,

$$\phi_3 = \bigwedge_{\gamma \in \Gamma} (x_\gamma = \sum_{d \in \delta} v_d(\gamma) \cdot y_d)$$

を満たす.

iv. (開始状態が Q_0, 終了状態が F)
$\forall q \in Q$ に対し, s_q, r_q は

$$\begin{cases} s_q = 0 & (q \notin Q_0) \\ \sum_{q \in Q_0} s_q = 1 & (q \in Q_0) \end{cases} \text{かつ} \begin{cases} r_q = 0 & (q \notin F) \\ \sum_{q \in F} r_q = 1 & (q \in F) \end{cases} \text{を満たす.}$$

すなわち,

$$\phi_4 = (\sum_{q \in Q_0} s_q = 1) \wedge (\sum_{q \in F} r_q = 1) \wedge (\bigwedge_{q \notin Q_0} (s_q = 0)) \wedge (\bigwedge_{q \notin F} (r_q = 0))$$

を満たす.

Presburger Formula $\phi(x_{\gamma_1}, \ldots, x_{\gamma_n}) = \exists \mathrm{y} \in \mathbb{N}^{\delta}, \mathrm{z}, \mathrm{r}, \mathrm{s}, \mathrm{n} \in \mathbb{N}^{Q}. (\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4)$ は T の出力の Parikh Image を表す. ここで, x_γ 以外の変数は束縛されている.

5.2 上記の Presburger formula の正当性

まず, T の出力が Presburger formula を満たすことを示す. transition の列 t に対して, y_d を t 中で $d \in \delta$ が使われた回数として $T_t(q) = \sum_{\delta \ni d : target(d) = q} y_d$, $S_t(q) = \sum_{\delta \ni d : source(d) = q} y_d$ とする. このとき, $D(q) = (T_t(q) - S_t(q)) + (s_q - r_q)$ である. transition の列 t に対し $t = d_{i_0}, d_{i_1} \cdots d_{i_{n-1}}$ であるとき $|t| = n$ である.

i について示す.

補題 5.2.1. $q_1 \longrightarrow q_2$ ならば y_d を各 transition の Parikh Image, s_q, r_q を開始状態, 終了状態を表す変数とするとき, $\forall q \in Q$. $D(q) = 0$.

証明. $s_q = \begin{cases} 1 & (q = q_1) \\ 0 & (q \neq q_1) \end{cases}$, $r_q = \begin{cases} 1 & (q = q_2) \\ 0 & (q \neq q_2) \end{cases}$ である.

$D(q) = 0$ を transition の長さの帰納法で示す. $q_1 \xrightarrow[t]{} q_2$ とする. $|t| = 0$ のとき, $q_1 \neq q_2$ であれば, 明らかに成り立つ. $q_1 = q_2 = p$ のとき, $D(p) = 0 - (1 - 1) = 0$, $\forall q \in Q \setminus \{p\}$. $D(q) = 0$ であるから成り立つ. $|t| = n$ で成り立つとして, $q_1 \xrightarrow[t]{} q_3 \xrightarrow[d_{i_n}]{} q_2$ のときに示す.

まず, $q_2 = q_3 = p$ であるとき, 新しく使われる遷移は $p \xrightarrow[d_{i_n}]{} p$ であり, 帰納法の仮定より, $\forall q \in Q \setminus \{p\}$ に対しては, $D(q) = 0$ である.

$$D(p) = ((T_t(p) + 1) - (S_t(p) + 1)) + (s_p - (r_p - 1 + 1)) \tag{5.3}$$
$$= (T_t(p) - S_t(p)) + (s_p - r_p) = 0 \tag{5.4}$$

より成り立つ.

次に, $q_2 \neq q_3$ であるとき, 新しく使われる遷移は $q_3 \xrightarrow[d_{i_n}]{} q_2$ であり, 帰納法の仮定より, $\forall q \in Q \setminus \{q_3, q_2\}$ に対しては, $D(q) = 0$ である.

$$D(q_3) = (T_t(q_3) - (S_t(q_3) + 1)) + (s_{q_3} - (r_{q_3} - 1)) \tag{5.5}$$
$$= (T_t(q_3) - S_t(q_3)) + (s_{q_3} - r_{q_3}) = 0 \tag{5.6}$$

$$D(q_2) = ((T_t(q_2) + 1) - S_t(q_2)) + (s_{q_2} - (r_{q_2} + 1)) \tag{5.7}$$
$$= (T_t(q_2) - S_t(q_2)) + (s_{q_2} - r_{q_2}) = 0 \tag{5.8}$$

より成り立つ. □

補題 5.2.1 より次が示せる.

定理 5.2.1. $q_0 \in Q_0, q_f \in F$ に対して $q_0 \longrightarrow q_f$ であるとき, y_d を各 transition の Parikh Image, s_q, r_q を開始状態, 終了状態を表す変数とすると, ϕ_1 を満たす.

ii について示す.

補題 5.2.2. $q_1 \longrightarrow q_2$ ならば n_q, y_d, s_q をそれぞれ q の transition 中の出現回数, transition の Parikh Image, 開始状態を表す変数とするとき, $\forall q \in Q.\ n_q = \sum_{\delta \ni d : target(d)=q} y_d + s_q$.

証明. $q_1 \xrightarrow[t]{} q_2$ として, transition の長さに関する帰納法で示す. $|t| = 0$ のとき, $q_1 \neq q_2$ であれば明らかに成り立つ. $q_1 = q_2 = p$ であるとき, $s_q = \begin{cases} 1 & (q = p) \\ 0 & (q \neq p) \end{cases}$ であり $n_p = 0 + 1 = 1$, $\forall q \in Q \setminus \{q\}$ について $n_q = 0 + 0 = 0$ で満たす. $|t| = n$ で満たすとして $|t| = n + 1$ で満たすことを示す. $q_1 \xrightarrow[t]{} q_3 \xrightarrow[d_{i_n}]{} q_2$ とする. 帰納法の仮定から t において $n_q = T_t(q) + s_q$ を満たす. $\forall q \in Q \setminus \{q_2\}$ に対しては満たす. q_2 について

$$n_{q_2} = (T_t(q_2) + 1) + s_q \tag{5.9}$$
$$= (T_{td_{i_n}}(q_2) + s_q) \tag{5.10}$$

より満たす. □

t 中で出現した状態の集合を $Q(t)$ とする.

補題 5.2.3. $q_0 \in Q_0$ に対して, $q_0 \xrightarrow[t]{} q_2$ かつ $p \in Q \setminus Q(t)$ ならば, n_p, z_p を transition 中の出現回数, p が q の transition 中で q_0 からの距離とすると ϕ_p^A.

証明. t の長さに関する帰納法で示す.
$|t| = 0$ のとき, $\forall q \in Q \setminus \{q_0\}$ に対して, $n_q = 0$ であり, このとき $z_q = -1$ である. $|t| = n$ のとき成り立つとして, $q_0 \xrightarrow[t]{} q_1 \xrightarrow[d_{i_n}]{} q_2$ とする. 帰納法の仮定より, $\forall q \in Q \setminus (Q(t) \cup \{q_2\})$ について $(n_q = 0) \wedge (z_q = -1)$. q_2 に関して, $n_{q_2} = 1$, $z_q = z_{q_1} + 1$ であり満たさない. □

補題 5.2.4. $q_0 \in Q_0$ に対して, $q_0 \xrightarrow[t]{} q_2$ かつ $p \in Q(t)$ ならば n_p, z_p, s_p を transition 中の出現回数, p が q の transition 中で q_0 からの距離, 開始状態を表す変数とすると ϕ_p^B.

証明. 見やすさのため, $(z_q = 0) \wedge (s_q = 1) = \phi_q^1$, $\bigvee_{\delta \ni d : target(d)=q} (z_q = z_{source(d)} + 1) \wedge (y_d > 0) \wedge (z_{source(d)} \geq 0) = \phi_q^2$ とする. t の長さに関する帰納法で示す.
$|t| = 0$ のとき, q_0 について, $s_{q_0} = 1$, $z_{q_0} = 0$, $n_q = 1 > 0$ で満たす. $|t| = n$ で満たすとして, $q_0 \xrightarrow[t]{} q_1 \xrightarrow[d_{i_n}]{} q_2$ とする.
帰納法の仮定より, $\forall q \in Q(t)$ について $(n_q > 0) \wedge (\phi_q^1 \vee \phi_q^2)$ を満たす. $q_2 \in Q(t)$ であれば z_{q_2}, s_{q_2} がとれているので満たす. $q_2 \notin Q(t)$ であるとする. $q_1 \in Q(t)$ より, $\phi_{q_1}^B$ を満たすような $z_{q_1} > 0$ が取れる. また, $y_{d_{i_n}} > 0$ であるから $z_{q_2} = z_{q_1} + 1$, $s_{q_2} = 0$ と置くと $n_{q_2} = 1 > 0$ であり $\phi_{q_2}^2$ を満たす. □

補題 5.2.2, 5.2.3, 5.2.4 より次が示せる.

定理 5.2.2. $q_0 \in Q_0$ に対して, $q_0 \xrightarrow[t]{} q_f$ であるとき, n_q, z_q, s_q を transition 中の出現回数, q の transition 中で q_0 からの距離, 開始状態を表す変数とすると ϕ_2 を満たす.

iii について示す.

補題 5.2.5. $q_1 \longrightarrow q_2$ ならば x_γ, y_d をそれぞれ出力の Parikh Image, transition の Parikh Image とすると, $\forall \gamma \in \Gamma.\ x_\gamma = \sum_{d \in \delta} v_d(\gamma) \cdot y_d$

証明. $q_1 \xrightarrow[t]{} q_2$ として, transition の長さに関する帰納法で示す. $|t| = 0$ のとき, 出力 $w' = \varepsilon$ で, $\Psi(w') = \mathbb{0}, \Psi(t) = \mathbb{0}$ であるから, $0 = \sum_{d\in\delta} v_d(\gamma) \cdot 0$ より成り立つ. $|t| = n$ で成り立つとして, $q_1 \xrightarrow[t]{} q_3 \xrightarrow[d_{i_n}]{} q_2$ のとき成り立つことを示す. 帰納法の仮定より, x_γ は出力の Parikh Image で, $x_\gamma = \sum_{d\in\delta} v_d(\gamma) \cdot y_d + v_{d_{i_n}}(\gamma)$ であり満たす. □

補題 5.2.5 より次がわかる.

定理 5.2.3. $q_1 \longrightarrow q_2$ ならば, x_γ, y_d をそれぞれ出力の Parikh Image, transition の Parikh Image とすると ϕ_3 を満たす.

ivについて示す.

定理 5.2.4. $q_0 \longrightarrow q_f$ かつ $q_0 \in Q_0, q_f \in F$ ならば s_q, r_q を開始状態, 終了状態を表す変数とするとき, ϕ_4 を満たす.

証明. $s_q = \begin{cases} 1 & (q = q_0) \\ 0 & (q \neq q_0) \end{cases}$, $r_q = \begin{cases} 1 & (q = q_f) \\ 0 & (q \neq q_f) \end{cases}$ であるから満たす. □

定理 5.2.1, 5.2.2, 5.2.3, 5.2.4 から以下を示すことができる.

定理 5.2.5. $q_0 \in Q_0$, $q_f \in F$ に対して, $q_0 \longrightarrow q_f$ であるとき, $x_\gamma, y_d, z_q, s_q, r_q, n_q$ をそれぞれ出力の Parikh Image, transition の Parikh Image, q の transition 中で q_0 からの距離, 開始状態・終了状態を表す変数, transition 中での q の出現回数とする. このとき, これらの変数は $\phi = \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4$ を満たす.

次に, Presburger formula を真にする変数に対応する T の tansition が存在することを示す.

補題 5.2.6. $\phi_2 \wedge (s_{q_0} = 1)$ かつ $z_q = n \geq 0$ であるとき, $t \in \delta^*$ が存在して $q_0 \xrightarrow[t]{} q$ かつ $(\Psi(t))_d \leq y_d$ を満たす.

証明. n に関する帰納法で示す. $n = 0$ のとき, $\varepsilon \in \delta^*$ について $q_0 \xrightarrow[\varepsilon]{} q_0$ であり, $(\varepsilon)_d = 0 \leq y_d$ より満たす. $n \leq k$ で満たすとして, $n = k + 1$ で満たすことを示す. ϕ_2 を満たすから, ある $d_{i_k} \in \delta$ と $p \in Q$ が存在して $z_p = z_q - 1 = k \geq 0$ かつ $y_{d_{i_k}} > 0$ を満たす. 帰納法の仮定より, $q_0 \xrightarrow[t]{} p$ かつ $(\Psi(t))_d \leq y_d$ なる $t \in \delta^*$ が存在する. ここで, $(\Psi(t))_{d_{i_k}} \neq 0$ とすると $q_0 \xrightarrow[t']{} q$, $\Psi(t') \leq \Psi(t)$ なる $t' \in \delta^*$ がとれるので満たす. $(\Psi(t))_{d_{i_k}} = 0$ とすると $q_0 \xrightarrow[t]{} p \xrightarrow[d_{i_k}]{} q$ で $y_{d_{i_k}} > 0$ より満たす. □

補題 5.2.7. ϕ_2 を満たすとき, $P = \{q \in Q |\ z_q \geq 0\}$, $\delta' = \{d \in \delta |\ y_d > 0\}$ とすると (P, δ') は (Q, δ) の q_0 から到達可能な部分グラフになる.

証明. 補題 5.2.6 より $z_q \geq 0$ なる $q \in Q$ は q_0 から $y_d > 0$ なる $d \in \delta$ のみを用いて到達可能. 逆に, $z_q = -1$ であるとき ϕ_2 を満たすから ϕ_q^A が成り立たなければならない. このとき, $n_q = 0$ であるから $\sum_{\delta \ni d : target(d) = q} y_d + s_q = 0$ であり $y_d \in \mathbb{N}, s_q \in \{0, 1\}$ より $target(d) = q$ なる d について $y_d = 0$. よって, $z_q = -1$ であるとき $y_d > 0$ なる $d \in \delta$ のみでは q_0 から到達できない. □

補題 5.2.8. $\phi_1 \wedge \phi_2 \wedge \phi_4$ を満たすとき, ある $q_0 \in Q_0, q_f \in F, t \in \delta^*$ が存在して, $q_0 \xrightarrow[t]{} q_f$ かつ $\forall d \in \delta.\ (\Psi_\delta(t))_d = y_d$.

証明. ϕ_2 を満たすから補題 5.2.7 より, (P, δ') をとることができる. $q, q' \in P$ 間に $q \underset{d}{\longrightarrow} q'$, $d \in \delta'$ が存在するとき $q, q' \in P$ 間に y_d 本の辺があるような有向多重グラフ G を考える. ϕ_4 を満たすからある $q_0, q_f \in Q$ が存在して, $s_q = \begin{cases} 1 & (q = q_0) \\ 0 & (q \neq q_0) \end{cases}$, $r_q = \begin{cases} 1 & (q = q_f) \\ 0 & (q \neq q_f) \end{cases}$ である. ϕ_1 を満たすから

$$\sum_{\delta \ni d : target(d) = q} y_d - \sum_{\delta \ni d : source(d) = q} y_d = \begin{cases} -1 & (q = q_0) \\ 1 & (q = q_f) \\ 0 & (q \neq q_0, q_f) \end{cases}$$

である. 故に, G で q_0 から q_f までのオイラー路を取ることができてこのオイラー路が $q_0 \underset{t}{\longrightarrow} q_f$ に対応し, $\forall d \in \delta.\ (\Psi_\delta(t))_d = y_d$ を満たす. □

定理 5.2.6. $\phi(x_{\gamma_1}, \ldots, x_{\gamma_n})$ を満たすとき, $q_0 \underset{t}{\longrightarrow} q_f$ かつ $\forall d \in \delta.\ (\Psi_\delta(t))_d = y_d$ なる $q_0 \in Q_0, q_f \in F, t \in \delta^*$ が存在し t の出力を w とするとき $(\Psi(w))_\gamma = x_\gamma$ を満たす.

証明. 補題 5.2.8 より $q_0 \underset{t}{\longrightarrow} q_f$ かつ $\forall d \in \delta.\ (\Psi_\delta(t))_d = y_d$ なる $q_0 \in Q_0, q_f \in F, t \in \delta^*$ をとることができる. t の出力 w の Parikh Image は $(\Psi(w))_\gamma = \sum_{d \in \delta} v_d(\gamma) \cdot y_d$ と書けるので ϕ_2 を満たすから, $(\Psi(w))_\gamma = x_\gamma$. □

第 6 章

文字列制約の SST を用いた充足可能性判定

直線制約 $\varphi = \varphi_{sl} \wedge \varphi_{reg}\varphi_{int}$ は各制約を等価な決定性 SST に変換することで解くことができる．以下にその概略を示す．定理 6.0.1, 6.0.2, 6.0.3, 6.0.4 の証明は [8] による．

φ_{sl} は各 $x_m = e_m$ に対し，$x_0\#\cdots\#x_{m-1}\#$ を入力として，$x_0\#\cdots\#x_{m-1}\#x_m\#$ を出力とするような SSTS_m を構成し，それらの $S_m, S_{m+1}, \ldots, S_{n-1}$ を合成することにより解くことができる．得られた決定性 SSTS は入力 $w_0\#\cdots\#w_{m-1}\#$ に対し，φ_{sl} を満たすような変数割り当ての一つ $w_0\#\cdots\#w_{n-1}\#$ を返す．S_m は入力 $x_0, \ldots, x_{m-1}$ に対して x_i を記録する状態 q_i をもち，$\#$ により q_i と q_{i+1} 間を遷移し，q_m において e_m に対応する出力関数をもつ出力 $x_0\#\cdots\#x_{n-1}\#$ の決定性 SST である

φ_{reg} は各変数 x_i が R_i に含まれるか調べる SSTS_n に変換することで解くことができる．具体的には入力 $x_0\#\cdots\#x_{n-1}\#$ に対して，各 R_i を受理するオートマトンで各変数 x_i を記録する状態 q_i を置き換えた出力 $x_0\#\cdots\#x_{n-1}\#$ の決定性 SST である．

定理 6.0.1. S を制約 $\varphi_{sl} \wedge \varphi_{reg}$ から構成された SST とする．
$[\![S]\!](w_0\#\ldots\#w_{m-1}\#) = w_0\#\ldots\#w_{n-1}\#$ ならば

$$[x_0 \to w_0, \ldots, x_{n-1} \to w_{n-1}] \models \varphi_{sl} \wedge \varphi_{reg}$$

.

定理 6.0.2. S を制約 $\varphi_{sl} \wedge \varphi_{reg}$ から構成された SST とする．
$[x_0 \to w_0, \ldots, x_{n-1} \to w_{n-1}] \models \varphi_{sl} \wedge \varphi_{reg}$ ならば

$$[\![S]\!](w_0\#\ldots\#w_{m-1}\#) = w_0\#\ldots\#w_{n-1}\#$$

φ_{int} は各 x_i の長さを出力するような SST に変換することで解くことができる．(Zhu によるソルバでは入力 $x_0\#\ldots\#x_m\#$ に対して $a_0^{|x_0|}\ldots a_m^{|x_m|}$(ただし、$a_i$ は相異なる文字) を出力とするような決定性 SST を構成し，Parikh image を出力するトランスデューサ $T_{\mathbb{N}}$ の半線形集合を構成し各変数の長さを得ていた.) これらの決定性 SST を合成することにより $\varphi_{sl} \wedge \varphi_{reg} \wedge \varphi_{int}$ を解く．

定理 6.0.3. S を制約 $\varphi_{sl} \wedge \varphi_{reg}$ から構成された SST, $T_{\mathbb{N}}$ を Parikh image を出力するトランスデューサとする．
$[\![T_{\mathbb{N}}]\!](w_0\#\ldots\#w_{m-1}\#) = (c_0, \ldots, c_{n-1})$ ならば，ある文字列 $w_m, \ldots, w_{n-1}$ が存在して

$$[x_0 \to w_0, \ldots, x_{n-1} \to w_{n-1}] \models \varphi_{sl} \wedge \varphi_{reg} \wedge (\bigwedge_{i=0}^{n-1} |w_i| = c_i)$$

定理 6.0.4. S を制約 $\varphi_{sl} \wedge \varphi_{reg}$ から構成された SST, $T_{\mathbb{N}}$ を Parikh image を出力するトランスデューサとする.
$[x_0 \to w_0, x_1 \to w_1, \ldots, x_{n-1} \to w_{n-1}] \models \varphi_{sl} \wedge \varphi_{reg} \wedge (\bigwedge_{i=0}^{n-1} |w_i| = c_i)$ ならば,

$$[\![T_{\mathbb{N}}]\!](w_0\#\ldots\#w_{m-1}\#) = (c_0, \ldots, c_{n-1})$$

以下では, 制約 $\varphi_{sl} \wedge \varphi_{reg} \wedge \varphi_{int} \wedge (x_i \neq x_j), (j < i)$ が解けることを示す. SST の等価性判定 [9] を利用してこれを解く. ここで決定性 SSTS, S' が等価であるとは, S と S' の定義域が等しく, 任意の入力 w について $[\![S]\!](w) = [\![S']\!](w)$ が成り立つことを表す. SST の等価性判定を利用することにより, $(x_i \neq x_j)$ を満たすことは以下の二条件に分割することができる.

- $|x_i| \neq |x_j|$.
- $|x_i| = |x_j|$ かつ $\exists p \in \{0, 1, \ldots \min\{|x_i|, |x_j|\}\}$. $x_i(p) \neq x_j(p)$. ただし, $x(p)$ は文字列 x の p 文字目を表す.

ここで, $|x_i| \neq |x_j|$ に関しては, φ_{int} に含まれるので, 定理 6.0.3, 6.0.4 で構成したような $S, T_{\mathbb{N}}$ を用いて解くことができる. 故に, 制約 $\varphi_{sl} \wedge \varphi_{reg} \wedge \varphi_{int} \wedge (|x_i| = |x_j|) \wedge (\exists p \in \{0, 1, \ldots |x_i|\}.\ x_i(p) \neq x_j(p))$ を解く.

$S = \langle \Sigma, \Gamma, Q, X, \delta, \eta, q_0, F \rangle$ の出力関数を x_i に変形した SST から x_i の $\sigma \in \Sigma$ で終わるような部分文字列を出力する非決定性 SST $S^i_\sigma = \langle \Sigma, \Gamma, Q', X', \delta', \eta', q'_0, F' \rangle$ が構成できる. S^i_σ から部分文字列長を出力するようなトランスデューサ T^i_σ が構成できる. (section 4, 3 より.) $P_{a,b}(x, y) = (\exists p \in \{0, 1, \ldots \min\{|x_i|, |x_j|\}\}.\ x(p) = a \wedge y(p) = b)$ とする. ただし, $a, b \in \Sigma$, $a \neq b$.

$T^i_a = \langle \Sigma, \mathbb{N}, Q, \delta^a, q_0, F^a \rangle$, $T^j_b = \langle \Sigma, \mathbb{N}, P, \delta^b, p_0, F^b \rangle$ の直積構成により, 入力 w に対する S^i_a, S^j_b の出力文字列長差を出力するようなトランスデューサが構成できる. これを $T_{a,b} = \langle \Sigma, \mathbb{N}, Q \times P, \delta^{a,b}, (q_0, p_0), F^a \times F^b \rangle$ とする. ただし, $\delta^{a,b}$ は $q \xrightarrow[T^i_a]{\sigma/n_1} q' \in \delta^a$ かつ $p \xrightarrow[T^j_b]{\sigma/n_2} p' \in \delta^b$ のとき $(q, p) \xrightarrow{\sigma/n_1 - n_2} (q', p') \in \delta^{a,b}$ とする. 以下の補題は入力 w の長さに関する帰納法で示すことができる. 証明は省略する.

補題 6.0.1. $q \xrightarrow[T^i_a]{w/n_1} q'$ かつ $p \xrightarrow[T^j_b]{w/n_2} p'$ ならば, $(q, p) \xrightarrow[T_{a,b}]{w/n_1 - n_2} (q', p')$ が成り立つ.

補題 6.0.2. $(q, p) \xrightarrow[T_{a,b}]{w/n} (q', p')$ であるとき, ある $n_1, n_2 \in \mathbb{N}$ が存在し, $n = n_1 - n_2$ かつ $q \xrightarrow[T^i_a]{w/n_1} q'$ かつ $p \xrightarrow[T^j_b]{w/n_2} p'$ が成り立つ.

S^i_a と S^j_b から構成した出力文字列長差をカウントするトランスデューサを $T^{i,j}_{a,b}$ とする.

定理 6.0.5. ある文字列 w で $0 \in [\![T^{i,j}_{a,b}]\!](w)$ ならば, ある $w^i \in [\![S^i]\!](w), w^j \in [\![S^j]\!](w)$ が存在して $P_{a,b}(w^i, w^j)$ を満たす.

証明. 補題 6.0.2 より, $0 \in [\![T^{i,j}_{a,b}]\!](w)$ であればある $n \in \mathbb{N}$ が存在して, $q \xrightarrow[T^i_a]{w/n} q'$, $p \xrightarrow[T^j_b]{w/n_1} p'$ を満たす.

定理 3.0.4 より, $[\![S^i_a]\!](w) = w^i_a, [\![S^j_b]\!](w) = w^j_b, |w^i_a| = |w^j_a|$ なる w^i_a, w^j_b が存在する. ここで, w^i_a, w^j_b は $[\![S^i_a]\!](w), [\![S^j_b]\!](w)$ の出力であるから, それぞれ a, b で終わる. 故に, w^i_a, w^j_b は $w^i = [\![S^i]\!](w), w^j = [\![S^j]\!](w)$ の部分文字列であるので, $w^i(n) = a, w^j(n) = b$ を満たすから, $P_{a,b}(w^i, w^j)$ が成り立つ. □

この定理は逆も成り立つ.

定理 6.0.6. ある文字列 $w^i \in [\![S^i]\!](w), w^j \in [\![S^j]\!](w)$ に対して, $P_{a,b}(w^i, w^j)$ ならば $0 \in [\![T^{i,j}_{a,b}]\!](w)$

証明. $P_{a,b}(w^i, w^j)$ であるから, ある $n \in \mathbb{N}$ が存在して, $w^i(n) = a \wedge w^j(n) = b$. ここで, $[\![T^i_a]\!](w), [\![T^j_b]\!](w)$ は w^i, w^j の部分文字列で a, b で終わるものの長さをすべて含む. 故に, $T^i_a(w) \ni n, T^j_b(w) \ni n$. よって, 補題 6.0.1 より, $0 \in T^{i,j}_{a,b}(w)$. □

定理 6.0.5, 6.0.6 から $\{w \mid S^i(w) \neq S^j(w)\}$ が半線形集合であることもわかる.

$T^{i,j}_{a,b} = \langle \Sigma, \mathbb{N}, Q, \delta^{a,b}, q_0, F^{a,b} \rangle$ と $\varphi_{sl} \wedge \varphi_{reg}$ から構成された $T_{\mathbb{N}} = \langle \Sigma, \mathbb{N}^\Gamma, P, \delta^{\mathbb{N}}, p_0, F^{\mathbb{N}} \rangle$ を直積構成する. 得られたトランスデューサ $T = \langle \Sigma, \mathbb{N}^\Gamma \times \mathbb{N}, \delta, (p_0, q_0), F^{\mathbb{N}} \times F^{a,b} \rangle$ が $\varphi_{sl} \wedge \varphi_{reg} \wedge P_{a,b}(w_i, w_j)$ を判定できることを示す. ただし, δ は $p \xrightarrow{w/v} p' \in \delta^{\mathbb{N}}$ かつ $q \xrightarrow{w/n} q' \in \delta^{a,b}$ であるとき, $(p, q) \xrightarrow{w/(v,n)} (p', q') \in \delta$ であるとする.

補題 6.0.3. $(p, q) \xrightarrow[T]{w/(v,n)} (p', q')$ ならば, $p \xrightarrow[T_{mathbbN}]{w/v} p'$ かつ $q \xrightarrow[T^{a,b}]{w/n} q'$.

補題 6.0.4. $p \xrightarrow[T_{\mathbb{N}}]{w/v} p'$ かつ $q \xrightarrow[T^{a,b}]{w/n} q'$ ならば, $(p, q) \xrightarrow[T]{w/(v,n)} (p', q')$ が成り立つ.

いずれも w の長さに関する帰納法で簡単に示すことができるので, 証明は省略する.

制約 $\varphi_{sl} \wedge \varphi_{reg}$ から構成された SST を S, $T_{\mathbb{N}}, T^{i,j}_{a,b}$ を S から構成されたトランスデューサとし, これらの直積により得られたトランスデューサを T とする.

定理 6.0.7. $[\![T]\!](w_0\#\ldots\#w_{m-1}\#) \ni ((c_0, \ldots, c_{n-1}), 0)$ のとき, ある文字列 $w_m, \ldots, w_{n-1}$ が存在して,

$$[x_0 \to w_0, x_1 \to w_1, \ldots, x_{n-1} \to w_{n-1}] \models \varphi_{sl} \wedge \varphi_{reg} \wedge (\bigwedge(|w_i| = c_i)) \wedge P_{a,b}(w_i, w_j)$$

証明. $[\![T]\!](w_0\#\ldots\#w_{m-1}\#) \ni ((c_0, \ldots, c_{n-1}), 0)$ より補題 6.0.3 から, $[\![T_{\mathbb{N}}]\!](w_0\#\ldots\#w_{m-1}\#) = (c_0, \ldots, c_{n-1})$ かつ $[\![T^{i,j}_{a,b}]\!](w_0\#\ldots\#w_{m-1}\#) \ni 0$. 故に, 定理 6.0.1, 6.0.3, 6.0.5 より, ある文字列 $w_m, \ldots, w_{n-1}$ が存在して, $[x_0 \to w_0, x_1 \to w_1, \ldots, x_{n-1} \to w_{n-1}] \models \varphi_{sl} \wedge \varphi_{reg}$ かつ $\bigwedge(|w_i| = c_i)$ かつ $P_{a,b}(w_i, w_j)$ が成り立つ. □

定理 6.0.8. $[x_0 \to w_0, x_1 \to w_1, \ldots, x_{n-1} \to w_{n-1}] \models \varphi_{sl} \wedge \varphi_{reg} \wedge (\bigwedge(|w_i| = c_i)) \wedge P_{a,b}(w_i, w_j)$ ならば,

$$[\![T]\!](w_0\#\ldots\#w_{m-1}\#) \ni ((c_0, \ldots, c_{n-1}), 0)$$

証明. 定理 6.0.2 より, $[\![S]\!](w_0\#\ldots\#w_{m-1}\#) = w_0\#\ldots\#w_{n-1}\#$ である. $P_{a,b}(w_i, w_j)$ であるとき, 定理 6.0.6 より $0 \in T^{i,j}_{a,b}(w)$. 定理 6.0.4 より, $T_{\mathbb{N}}(w_0\#\ldots\#w_{m-1}\#) = (c_0, \ldots, c_{n-1})$ であり, 補題 6.0.4 から, $[\![T]\!](w_0\#\ldots\#w_{m-1}\#) \ni ((c_0, \ldots, c_{n-1}), 0)$ である. □

$\varphi_{sl} \wedge \varphi_{reg} \wedge (\bigwedge_{h=0}^{k-1} P^{i_h,j_h}_{a_h,b_h}(w^{i_h}, w^{j_h}))$ を解くようなトランスデューサ T が構成できることを示す. これは $P^{i_h,j_h}_{a_h,b_h}$ に対して, 上記のようにトランスデューサを構成し, それらの直積をとることによってできる.

$k = 2$ のときに構成法を示す. $k = 1$ のとき, 上述のように $\varphi_{sl} \wedge \varphi_{reg} \wedge (x_i \neq x_j)$ から各 $x_i \in X$ の長さと x_i, x_j の部分文字列長差を出力するトランスデューサ T が構成できる. 同様に, $\varphi_{sl} \wedge \varphi_{reg}$ から構成された SST S から $(x'_i \neq x'_j)$ に対して, 部分文字列長差を出力するトランスデューサ $T^{i',j'}_{a',b'}$ が構成できる. T と $T^{i',j'}_{a',b'}$ を直積構成することで, 各 $x_i \in X$ の長さと x_i, x_j の部分文字列長差, x'_i, x'_j の部分文字列長差を出力するトランスデューサ $T' = \langle \Sigma, \mathbb{N} \times \mathbb{N}^2, Q', \delta', q'_0, F' \rangle$ が構成できる. ここで, Q', q'_0, F' は T と $T^{i',j'}_{a',b'}$ の直積をとったものである. δ' について, $(p, q) \xrightarrow[T]{\sigma/(v,n)} (p', q')$ かつ $r \xrightarrow[T^{i',j'}_{a',b'}]{\sigma/m} r'$ であるときに, $(p, q, r) \xrightarrow[T']{\sigma/(v,u)} (p', q', r')$ とす

る. ここで, $u=(n,m)\in\mathbb{N}$ である. $\varphi_{sl}\wedge\varphi_{reg}\wedge(\bigwedge_{h=0}^{k-1}P_{a_h,b_h}^{i_h,j_h}(w^{i_h},w^{j_h}))$ から構成されたトランスデューサ T' は入力 $(\Sigma\#)^m$, 出力 $\mathbb{N}^n\times\mathbb{N}^k$ である. $\varphi_{sl}\wedge\varphi_{reg}\wedge(\bigwedge_{h=0}^{k-1}P_{a_h,b_h}^{i_h,j_h}(w^{i_h},w^{j_h}))$ により構成されたトランスデューサ T, $\varphi_{sl}\wedge\varphi_{reg}$ と $P_{a_k,b_k}^{i_k,j_k}(w^{i_k},w^{j_k})$ から構成されたトランスデューサ T_k とする. T,T_k の直積構成によるトランスデューサを T' とするとき, 以下が成り立つ.

補題 6.0.5. $q\xrightarrow[T]{w/(v,u)}q'$ かつ $p\xrightarrow[T_k]{w/n}p'$ ならば,

$$q@p\xrightarrow[T']{w/(v,u@n)}q'@p'$$

ただし, $v\in\mathbb{N}^k, n\in\mathbb{N}$ に対し, $v@n\in\mathbb{N}^{k+1}$ は v に n を新しい最後の要素として加えた $k+1$ 次元のベクトルとする.

証明. $P_{a,b}^{i,j}(w^i,w^j)$ の個数による帰納法で示す. $k=0$ のとき, 定理 6.0.4 より成り立つ. $k=n$ まで成り立っているとき, $k=n+1$ で成り立つことを示す. w の長さの帰納法により示す. $w=0$ のとき, 明らかに成り立つ. $w=m$ で成り立つとき, $w=m+1$ でも成り立つことを示す.
$q\xrightarrow[T]{w/(v,u)}q''\xrightarrow[T]{\sigma/(v',u')}q'$ かつ $p\xrightarrow[T_k]{w/n}p''\xrightarrow[T_k]{\sigma/n'}p'$ であるとする. 帰納法の仮定より, $q@p\xrightarrow[T']{w/(v,u@n)}q''@p''$ である. また, T' の構成法より, $q''@p''\xrightarrow[T']{\sigma/(v',u'@n')}q''@p''$ である. 故に, $q@p\xrightarrow[T']{w/(v,u@n)}q''@p''\xrightarrow[T']{\sigma/(v',u'@n')}q''@p''$ であるから, $w=m+1$ でも成り立つ. 故に, $k=m+1$ でも成り立つ. □

同様にして逆も示すことができる.

補題 6.0.6. $q@p\xrightarrow[T']{w/(v,u@n)}q'@p'$ ならば, $q\xrightarrow[T]{w/(v,u)}q'$ かつ $p\xrightarrow[T_k]{w/n}p'$

以下では, 見やすさのため, $\varphi_{sl}\wedge\varphi_{reg}=P_0$, $(\bigwedge_{h=0}^{k-1}P_{a_h,b_h}^{i_h,j_h}(w^{i_h},w^{j_h}))=P(k)$ とかく.

定理 6.0.9. $T'(w_0\#w_1\#\ldots\#w_{m-1}\#)\ni((c_0,c_1,\ldots,c_{n-1}),0)$ とする. このとき, ある文字列 $w_m,\ldots,w_{n-1}$ が存在して, $[x_0\to w_0,x_1\to w_1,\ldots,x_{n-1}\to w_{n-1}]\models\varphi_{sl}\wedge\varphi_{reg}$ かつ $\bigwedge(|w_i|=c_i)$ かつ $(\bigwedge_{h=0}^{k-1}P_{a_h,b_h}^{i_h,j_h}(w^{i_h},w^{j_h}))$ を満たす.

証明. $P_{a,b}^{i,j}$ の個数の帰納法で示す. $k=0$ のとき, 定理 6.0.3 より成り立つ.
$k=n$ のとき, 成り立つとして, $k=n+1$ で成り立つことを示す. $P_0\wedge P(n)\wedge P_{a_k,b_n}^{i_n,j_n}(w^{i_n},w^{j_n})$ なる constraint は $P_0\wedge P(n)$ と $P_0\wedge P_{a_n,b_n}^{i_n,j_n}(w^{i_n},w^{j_n})$ を解くことができればよい. 補題 6.0.6 より, $T'(w_0\#w_1\#\ldots\#w_{m-1}\#)\ni((c_0,c_1,\ldots,c_{n-1}),0)$ であるとき, $T(w_0\#w_1\#\ldots\#w_{m-1}\#)\ni((c_0,c_1,\ldots,c_{n-1}),0)$ かつ $T_n(w_0\#w_1\#\ldots\#w_{m-1}\#)\ni((c_0,c_1,\ldots,c_{n-1}),0)$ が成り立つ. 帰納法の仮定から, $T(w_0\#w_1\#\ldots\#w_{m-1}\#)\ni((c_0,c_1,\ldots,c_{n-1}),0)$ であるとき, ある文字列 $w_m,\ldots,w_{n-1}$ が存在し, $[x_0\to w_0,x_1\to w_1,\ldots,x_{n-1}\to w_{n-1}]\models P_0\wedge(\bigwedge(|w_i|=c_i))\wedge P(n)$ を満たす. 定理 6.0.5 より, $T_n(w_0\#w_1\#\ldots\#w_{m-1}\#)\ni((c_0,c_1,\ldots,c_{n-1}),0)$ であるとき, ある文字列 $w_m,\ldots,w_{n-1}$ が存在し, $[x_0\to w_0,x_1\to w_1,\ldots,x_{n-1}\to w_{n-1}]\models P_0\wedge(\bigwedge(|w_i|=c_i))\wedge P_{a_k,b_n}^{i_n,j_n}(w^{i_n},w^{j_n})$ を満たす. ここで, それぞれの $w_m,\ldots,w_{n-1}$ は, P_0 から構成された SST の出力であるから同様の文字列で成り立つ. 故に, $P_0\wedge(\bigwedge(|w_i|=c_i))\wedge P(n+1)$ を満たす. □

定理 6.0.10. $[x_0\to w_0,x_1\to w_1,\ldots,x_{n-1}\to w_{n-1}]\models\varphi_{sl}\wedge\varphi_{reg}$ かつ $\bigwedge(|w_i|=c_i)$ かつ $(\bigwedge_{h=0}^{k-1}P_{a_h,b_h}^{i_h,j_h}(w^{i_h},w^{j_h}))$ であるとき, $T'(w_0\#w_1\#\ldots\#w_{m-1}\#)\ni((c_0,c_1,\ldots,c_{n-1}),0)$ を満たす.

証明. $P^{i,j}_{a,b}$ の個数の帰納法で示す. $k=0$ のとき, 定理 6.0.4 より成り立つ.
$k=n$ のとき成り立つとして, $k=n+1$ で成り立つことを示す. 帰納法の仮定より, T は $P_0 \wedge \bigwedge(|w_i| = c_i) \wedge P(n)$ を満たす. 定理 6.0.7 より T_n は $P_0 \wedge \bigwedge(|w_i| = c_i) \wedge P^{i_n,j_n}_{a_k,b_n}(w^{i_n}, w^{j_n})$ を満たす. 補題 6.0.5 より, T' は $P_0 \wedge \bigwedge(|w_i| = c_i) \wedge P(n+1)$ を満たす. □

第 7 章

実験

充足可能であるとき sat, 充足不能であるとき unsat とかく.

実験 1. 制約

$$x_0 = x_1$$
$$x_0 \neq x_1$$

結果 unsat　実行時間: 1.228s

実験 2. 制約

$$x_0 \neq x_1$$
$$|x_0| = |x_1|$$
$$|x_0| = 5$$

結果 sat

$$x_0 = abbbb$$
$$x_1 = abbba$$

実行時間: 1.265s

実験 3. 制約

$$x_0 \in \{a\}$$
$$x_1 \in \{a, b\}$$
$$x_2 \in \{a, b, c\}$$
$$x_3 \in \{a, b, c, d\}$$
$$x_4 \in \{a\}$$
$$x_1 \neq x_2$$
$$x_2 \neq x_3$$
$$x_3 \neq x_1$$

結果 sat

$$x_0 = a, x_1 = a, x_2 = b,$$
$$x_3 = c, x_4 = a$$

実行時間: 21.948s

実験 4. 制約

$$\begin{aligned} y &= reverse(x) \\ z &= y.replaceAll(aa, x) \\ z1 &= reverse(z) \\ x1 &= x.replaceAll(aa, x) \\ x1 &\neq z1 \\ |x| &= 3 \end{aligned}$$

結果 sat

$$\begin{aligned} x = aaa, y = aaa, z = xa, \\ x1 = xa, z1 = ax \end{aligned}$$

実行時間: 2.274s

実験 5. 制約

$$\begin{aligned} y &= reverse(x) \\ z &= y.replaceAll(aa, x) \\ z1 &= reverse(z) \\ x1 &= x.replaceAll(aa, x) \\ x1 &\neq z1 \\ |x| &= 2 \end{aligned}$$

結果 unsat
実行時間: 3.388s

実験 6. 制約

$$\begin{aligned} y_0 &= x_0.replaceAll(< sc >, \varepsilon) \\ y_1 &= x_1.replaceAll(< sc >, \varepsilon) \\ 0 &< |x_0| \\ 0 &< |x_1| \\ |y_0| &= |y_1| \\ |y_0| &= 10 \end{aligned}$$

結果 sat

$$\begin{aligned} x_0 &= sc << sc >< sc < sc >< sc < sc > s \\ x_1 &= s <<<<<<<<< s \\ y_0 &= sc << sc < scs \\ y_1 &= s <<<<<<<<< s \end{aligned}$$

実行時間: 8.663s

Zhu によるソルバでは, SST の Parikh Image を半線形集合として構成していた．この構成法はトランスデューサの状態数の指数オーダーかかることがあり, このような例は 10 分以上かけても解けていなかったが, 本研究では Presburger Formula にすることでトランスデューサの状態数の線形のオーダーで Parikh Image を構成できるため, 短時間で解くことができるようになった．しかし, $< sc >$ を $< scr >$ に置き換えると $40.281s$, $< scri >$ で $162.321s$, $< scrip >$ で $728.253s$ と時間がかかってしまった．これは, $replaceAll$ を表すトランスダクションが置換する文字列の長さと同じ数の状態数をもつ SST によって表しているために, 構成される SST の状態数が大きくなってしまうことが原因としてある．

第 8 章

結論

本論文では, Zhu によるソルバの等号否定 $x_i \neq x_j$ を含む文字列制約に対しての拡張を行った. 具体的な手法としては基礎直線制約と正規制約から構成した関数的 SST の等価性判定により, $x_i \neq x_j$ を満たす入力が存在するかを判定し, SST S, S' の等価性判定において $\{w \mid S(w) \neq S'(w)\}$ が半線形集合であることを証明した. また, トランスデューサの Parikh Image を先行研究においては半線形集合として与えていたため, 半線形集合が大きくなってしまう問題があったが, Presburger Formula として与えることにより論理式の大きさが $O(|Q|)$ となるようにした. さらに, 実際に Scala を用いて実装を行った. しかし入力によっては SST が大きくなってしまい, 結果として SMT ソルバに与える論理式のサイズが大きくなるために, そのような入力については時間がかかってしまうことが課題の一つとしてある. また, 等号否定の数が増えると, 一つの等号否定につき $O(\log_2(|\Sigma|))$ の組みを調べなければならないため, 時間がかかってしまうことも課題としてある.

第 9 章

謝辞

最後に, 本論文を書くにあたり指導教員である南出靖彦教授に多大なるご助力を受け賜わりました. 厚く感謝を申し上げます.

参考文献

[1] Rajeev Alur. Expressiveness of streaming string transducers. *IARCS International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010)*, 2010.

[2] Taolue Chen, Matthew Hague, Anthony W Lin, Philipp Rümmer, and Zhilin Wu. Decision procedures for path feasibility of string-manipulating programs with complex operations. *Proceedings of the ACM on Programming Languages*, Vol. 3, No. POPL, pp. 1–30, 2019.

[3] Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp. 337–340. Springer, 2008.

[4] Lukáš Holík, Petr Janků, Anthony W Lin, Philipp Rümmer, and Tomáš Vojnar. String constraints with concatenation and transducers solved efficiently. *Proceedings of the ACM on Programming Languages*, Vol. 2, No. POPL, p. 4, 2017.

[5] Anthony W Lin and Pablo Barceló. String solving with word equations and transducers: towards a logic for analysing mutation XSS. In *43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming*, Vol. 51, pp. 123–136. ACM, 2016.

[6] Christophe Morvan. On rational graphs. In *International Conference on Foundations of Software Science and Computation Structures*, pp. 252–266. Springer, 2000.

[7] Kumar Neeraj Verma, Helmut Seidl, and Thomas Schwentick. On the complexity of equational Horn clauses. In *International Conference on Automated Deduction*, pp. 337–352. Springer, 2005.

[8] Qizhen Zhu, Hitoshi Akama, and Yasuhiko Minamide. Solving string constraints with streaming string transducers. *Information Processing*, Vol. 27, pp. 810–821, 2019.

[9] 加賀江優幸, 南出靖彦. Streaming string transducer の等価性判定と正規表現による文字列置換への応用. 情報処理学会論文誌プログラミング (PRO), Vol. 8, No. 3, pp. 1–10, 2015.