一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
SS2012-47(2013-01)

制約付き項のインスタンスを受理する制約付き木オートマトンの構成法

中野　靖大[†]　西田　直樹[††]　酒井　正彦[††]　坂部　俊樹[††]　草刈圭一朗[††]

†,†† 名古屋大学大学院情報科学研究科
〒 464–8603 名古屋市千種区不老町
E-mail: † ynakano@trs.cm.is.nagoya-u.ac.jp, ††{nishida, sakai, sakabe, kusakari}@is.nagoya-u.ac.jp

あらまし　制約付き項書換え系の書換え帰納法に基づいた定理自動証明の際に，制約付き項書換え系の R 完全性の判定手続きが必要である．また，対象の制約付き項書換え系が十分完全性を持つ場合に，定理自動証明中の推論規則の適用条件を緩和することができる．これらの性質は，制約付き項のインスタンスの集合に関する積集合空問題に帰着できる．本論文では，制約付き項のインスタンスを受理する制約付き木オートマトンの構成法を提案する．さらに，制約付き木オートマトン中でどの入力基底項も到達することのない状態を削除する手法を提案することで，R 完全性や十分完全性の判定の精度向上と効率化をめざす．

キーワード　制約付き項書換え系，積集合空問題，十分完全性

Construction of Constrained Tree Automata Recognizing Ground Instances of Constrained Terms

Yasuhiro NAKANO[†], Naoki NISHIDA[††], Masahiko SAKAI[††],
Toshiki SAKABE[††], and Keiichirou KUSAKARI[††]

† Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464–8603
E-mail: † ynakano@trs.cm.is.nagoya-u.ac.jp, ††{nishida, sakai, sakabe, kusakari}@is.nagoya-u.ac.jp

Abstract　A theorem proving method for constrained term rewriting systems, which is based on rewriting induction, needs a decision procedure for R-completeness of constrained term rewriting systems. In addition, sufficient completeness of constrained term rewriting systems enables us to relax the side conditions of some inference rules in the proving method. These two properties of constrained term rewriting systems can be reduced to intersection emptiness problems related to sets of ground instances for constrained terms. In this paper, we propose a method to construct constrained tree automata recognizing ground instances of constrained terms. We also propose a method to remove states from a constrained tree automaton, to which no ground term transitions, thus improving the accuracy for the judgements of R-completeness and sufficient completeness.

Key words　constrained term rewriting system, intersection emptiness problem, sufficient completeness

1. はじめに

制約付き項書換え系 [5], [6] とは，書換え規則に付随した制約が真のときにその書換え規則を適用して項の書換えを行う計算モデルである．制約付き項書換え系の書換え帰納法に基づいた定理自動証明の研究が行われており，その証明中で項の R 完全性の判定手続きが必要である．項の位置が R 完全であるとは，その位置にある部分項のあらゆるインスタンスが先頭の位置で書き換えられる性質である．また，十分完全性とはすべての関数が任意の入力に対して定義されている性質であり，制約付き項書換え系が十分完全性を持つ場合には定理自動証明中の推論規則の適用条件を緩和することができる [5]．これらの性質は制約付き項のインスタンスの集合に関する積集合空問題に帰着することができ，制約付き木オートマトンを利用して解くことができる．制約付き項とは制約を付随された項であり，そのインスタンスは付随した制約を充足する代入により具体化された項である．ここで，制約付き木オートマトン [4] とは，項を入力とするオートマトンである木オートマトン [3] を拡張したもの

であり，遷移規則に付随した項中の解釈できる部分項に関する制約が真のときに遷移を行う．

本論文では，制約付き項のインスタンスを受理する制約付き木オートマトンを構成する手法を提案する（3 節）．ここで構成する制約付き木オートマトンは，制約完全性 [4]・完全性・決定性の 3 つの性質を満たすように構成する．制約完全性は構成するオートマトンの正当性に必要である．完全性・決定性を持つようにオートマトンを直接構成することによって，最終的に構成されるオートマトンのサイズをなるべく小さくする．制約付き木オートマトンの積集合空問題は一般には決定不能であり，積集合が空であるための自明な十分条件は積集合を受理するように構成された制約付き木オートマトンの最終状態が存在しないことである．しかし，積集合を受理するオートマトンを構成する際に，あらゆる状態の組が生成されてしまい，さらに，制約付き木オートマトンでは一般に状態が基底項から到達できないかどうかは決定できない．そこで，前述の十分条件を利用するにはオートマトンの状態は少ない方がよく，なるべく状態の少ないオートマトンを作ることが判定可能なクラスの拡大につながると考えられる．そこで，構成された制約付き木オートマトン中でどの入力基底項も到達できない状態を削除する手法を提案する（5 節）．

2. 準　　備

本節では，本論文で必要な諸定義を与える [1], [2].

$\mathcal{F}$ を関数記号の集合，$\mathcal{V}$ を変数の加算無限集合とする．X ($\subseteq\mathcal{V}$) を変数の集合とする．このとき，$\mathcal{F}$，X から生成されるすべての項の集合を $\mathcal{T}(\mathcal{F},X)$ とする．関数記号 f の引数個数を $arity(f)$ で表す．また，変数を含まない項を基底項と呼び，$\mathcal{F}$ からなるすべての基底項の集合を $\mathcal{T}(\mathcal{F})$ とする．項 t 中に現れるすべての変数の集合を $\mathcal{V}ar(t)$ で表す．項 s と t が構造的に同一であるときは $s\equiv t$ と記述する．項 t における位置の集合を $\mathcal{P}os(t)$ とする．ホール $\Box\notin\mathcal{F}$ を特別な定数記号とする．文脈とは，$\Box$ を一つだけ含む項である．ホール自身も文脈であり，このような文脈を空文脈という．文脈 $C[\,]$ において位置 p に出現するホール $\Box$ を項 t で置き換えることによって得られる項を $C[t]_p$ と記す．なお，p を省略してもよい．$\mathcal{F},X$ 中のすべての文脈の集合を $\mathcal{T}_\Box(\mathcal{F},X)$ とする．項 t,u に対して $t\equiv C[u]_p$ となるような文脈 $C[\,]$ が存在するとき，u を t の部分項と呼ぶ．また，p における部分項 u を $t|_p$ と記す．

代入 σ の定義域と値域をそれぞれ $\mathcal{D}om(\sigma)$ $(=\{x\mid x\not\equiv\sigma(x)\})$ と $\mathcal{R}an(\sigma)$ $(=\{\sigma(x)\mid x\in\mathcal{D}om(\sigma)\})$ で表す．$\mathcal{D}om(\sigma)=\{x_1,\ldots,x_n\}$ かつ $\sigma(x_1)\equiv t_1,\ldots,\sigma(x_n)\equiv t_n$ のとき，σ を $\{x_1\mapsto t_1,\ldots,x_n\mapsto t_n\}$ と記す．項 t に対して，$\sigma(t)$ を t のインスタンスと呼び，$t\sigma$ と略記する．$t\sigma_g$ が基底項となるような σ_g を t に対する基底代入という．単純に基底代入と呼ぶときには，それ以降に現れる項にその代入を行うと基底項になる代入を指す．

$\mathcal{G}$ を $\mathcal{F}\cap\mathcal{G}=\emptyset$ となる関数記号の集合、$\mathcal{P}$ を述語記号の集合とする．$\mathcal{P}$ 中の述語記号 p の引数個数を $arity(p)$ で表す．$\mathcal{G}$ と $\mathcal{P}$ で定義される論理式 ϕ を BNF 記法を用いて次のように定める：
$\phi ::= p(t_1,\ldots,t_n)\mid\neg\phi\mid\phi\vee\phi\mid\phi\wedge\phi\mid\top\mid\bot$．ただし，$p\in\mathcal{P}$，$arity(p)=n$，$t_1,\ldots,t_n\in\mathcal{T}(\mathcal{G},X)$ とする．$\top$，$\bot$ はそれぞれ真と偽に相当する．なお，本論文では限量子は用いない．論理式 ϕ 中に現れるすべての変数の集合を $f_\mathcal{V}(\phi)$ で表す．また，変数を含まない論理式を閉論理式と呼ぶ．$\mathcal{G},\mathcal{P},X$ からなる論理式の集合を $L(\mathcal{G},\mathcal{P},X)$ と記述する．$\mathcal{G}$ と $\mathcal{P}$ に解釈を与える構造を $\mathcal{M}$ とする．$\mathcal{M}$ は，領域と呼ばれる空でない集合 A，$\mathcal{G}$ と $\mathcal{P}$ の解釈から成り立つ．任意の n 引数関数記号 $g\in\mathcal{G}$ は，$A^n\to A$ の関数で解釈され，その関数を $g^\mathcal{M}$ と書く．任意の n 引数述語関数 $p\in\mathcal{P}$ は，A^n の組の集合 ($\subseteq A^n$) で解釈され，その組を $p^\mathcal{M}$ と書く（すなわち $p^\mathcal{M}\subseteq A^n$）．基底項 $g(t_1,\ldots,t_n)\in\mathcal{T}(\mathcal{G})$ の解釈は $(g(t_1,\ldots,t_n))^\mathcal{M}=g^\mathcal{M}((t_1)^\mathcal{M},\ldots,(t_n)^\mathcal{M})$ とし，閉論理式 ϕ の $\mathcal{M}$ での解釈は通常の定義に従う．ϕ が $\mathcal{M}$ で真と解釈されるとき，ϕ は $\mathcal{M}$ のもとで真であるといい，$\mathcal{M}\models\phi$ と書く．一方，偽と解釈されるときは，ϕ は $\mathcal{M}$ のもとで偽であるといい，$\mathcal{M}\not\models\phi$ と書く．$\mathcal{M}$ で解釈される論理式を（$\mathcal{M}$ で解釈される）制約と呼ぶ．

次に，制約付き木オートマトンに関する用語や概念を文献 [4] に基づき定義する．制約付き木オートマトンの遷移規則に付随される論理式では，通常の変数の代わりに対象項の位置を表す自然数列を用いる．自然数列 $a\in\mathcal{N}^+$ を $\langle\ \rangle$ で囲んだ記号 $\langle a\rangle$ を変数とみなし，それらの集合を $\langle\mathcal{N}^+\rangle$ ($\subseteq\mathcal{V}$) とする．制約 $\phi\in L(\mathcal{G},\mathcal{P},\langle\mathcal{N}^+\rangle)$ 中に出現するすべての自然数列の集合を $CPos(\phi)$ とする．付随した論理式が $\top$ である遷移規則 $l\xrightarrow{\top}r$ は単に $l\to r$ と書いてもよい．

$(\mathcal{F},\mathcal{G},\mathcal{P},\mathcal{M})$ 上の制約付き木オートマトン（制約付き TA）を，七つ組み $(Q,\mathcal{F},\mathcal{G},\mathcal{P},\mathcal{M},Q_f,\Delta)$ で定義する．ただし，Q を $Q\cap(\mathcal{F}\cup\mathcal{G})=\emptyset$ を満たす状態の有限集合とし，Q_f を $Q_f\subseteq Q$ を満たす受理状態の集合とする．さらに，Δ を遷移規則 $f(q_1(x_1),\ldots,q_n(x_n))\xrightarrow{\phi}q(f(x_1,\ldots,x_n))$ の有限集合とする．ただし，$f\in\mathcal{F}\cup\mathcal{G}$，$arity(f)=n$，$q_1,\ldots,q_n,q\in Q$，$\phi\in L(G,\mathcal{P},\langle\mathcal{N}^+\rangle)$，$x_1,\ldots,x_n$ はすべて異なる変数である．制約付き木オートマトン $(Q,\mathcal{F},\mathcal{G},\mathcal{P},\mathcal{M},Q_f,\Delta)$ を (Q,Q_f,Δ) の三つ組で略記する．

次に，制約付き TA の遷移関係の定義に必要である，項に対する論理式の解釈を定義する．基底項 $t\in\mathcal{T}(\mathcal{F}\cup\mathcal{G})$ を環境とした項 $u\in\mathcal{T}(\mathcal{G},\langle\mathcal{N}^+\rangle)$ から得られる項 $I_t(u)$ を以下のように再帰的に定義する．

- $f\in\mathcal{G}$ かつ $arity(f)=n$ かつ $I_t(t_1),\ldots,I_t(t_n)$ が定義されているならば，$I_t(f(t_1,\ldots,t_n))=f(I_t(t_1),\ldots,I_t(t_n))$.
- $\langle\pi\rangle\in\langle\mathcal{N}^+\rangle$ かつ $\pi\in\mathcal{P}os(t)$ かつ $t|_\pi\in\mathcal{T}(\mathcal{G})$ ならば，$I_t(\langle\pi\rangle)=t|_\pi$.

項 $t\in\mathcal{T}(\mathcal{F}\cup\mathcal{G})$ を環境とした論理式 $\phi\in L(\mathcal{G},\mathcal{P},\langle\mathcal{N}^+\rangle)$ の $\mathcal{M}$ における成立 $\mathcal{M},t\models\phi$ を以下のように再帰的に定義する．

- $p\in\mathcal{P}$ かつ $arity(p)=n$ かつ $t_1,\ldots,t_n\in\mathcal{T}(\mathcal{G},\langle\mathcal{N}^+\rangle)$ かつ $I_t(t_1),\ldots,I_t(t_n)$ が定義されており $\mathcal{M}\models p(I_t(t_1),\ldots,I_t(t_n))$ ならば，$\mathcal{M},t\models p(t_1,\ldots,t_n)$.
- $\mathcal{M},t\models\top$.
- $\mathcal{M},t\not\models\phi'$ ならば，$\mathcal{M},t\models\neg\phi'$.

- $\mathcal{M}, t \models \phi_1$ または $\mathcal{M}, t \models \phi_2$ ならば，$\mathcal{M}, t \models \phi_1 \vee \phi_2$.
- $\mathcal{M}, t \models \phi_1$ かつ $\mathcal{M}, t \models \phi_2$ ならば，$\mathcal{M}, t \models \phi_1 \wedge \phi_2$.

次に，制約付き TA の遷移関係を定義する．$(\mathcal{F}, \mathcal{G}, \mathcal{P}, \mathcal{M})$ 上の制約付き TA $\mathcal{A} = (Q, Q_f, \Delta)$ の遷移関係 $\rightarrow_{\mathcal{A}}$ は以下のすべてを満たす最小の関係である．

- $f \in \mathcal{F} \cup \mathcal{G}$ かつ $arity(f) = n$ かつ $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F} \cup \mathcal{G})$ かつ $f(q_1(x_1), \ldots, q_n(x_n)) \xrightarrow{\phi} q(f(x_1, \ldots, x_n)) \in \Delta$ かつ $\mathcal{M}, f(t_1, \ldots, t_n) \models \phi$ ならば，$f(q_1(t_1), \ldots, q_n(t_n)) \rightarrow_{\mathcal{A}} q(f(t_1, \ldots, t_n))$.
- $f \in \mathcal{F} \cup \mathcal{G}$ かつ $arity(f) = n$ かつ $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup Q)$ かつ $t_i \rightarrow_{\mathcal{A}} u$ かつ $t_1, \ldots, t_n$ には Q の入れ子が存在しないならば，$f(t_1, \ldots, t_n) \rightarrow_{\mathcal{A}} f(t_1, \ldots, t_{i-1}, u, t_{i+1}, \ldots, t_n)$.

$(\mathcal{F}, \emptyset, \emptyset, \mathcal{M})$ 上の制約付き TA は木オートマトン (TA) [3] となる．$\rightarrow^*_{\mathcal{A}}$ は $\rightarrow_{\mathcal{A}}$ の反射推移閉包を表し，$\rightarrow^n_{\mathcal{A}}$ は $\rightarrow_{\mathcal{A}}$ の n ステップの遷移を表す．以降では，状態記号 q の引数を省略する．例えば，$q(x)$ や $q(t)$ は q と記述し，遷移規則 $f(q_1(x_1), \ldots, q_n(x_n)) \xrightarrow{\phi} q(f(x_1, \ldots, x_n))$ は単に $f(q_1, \ldots, q_n) \xrightarrow{\phi} q$ と記す．基底項 $t \in \mathcal{T}(\mathcal{F} \cup \mathcal{G})$ に対して $t \rightarrow^*_{\mathcal{A}} q$ かつ $q \in Q_f$ のとき，$\mathcal{A}$ は t を受理するという．$\mathcal{A}$ が受理する項の集合を $L(\mathcal{A})$ で表す．任意の基底項 $t \in \mathcal{T}(\mathcal{F} \cup \mathcal{G})$ について $t \rightarrow^*_{\mathcal{A}} q$ となる状態 $q \in Q$ が存在するとき，$\mathcal{A}$ は完全であるという．任意の基底項 $t \in \mathcal{T}(\mathcal{F} \cup \mathcal{G})$ と状態 $q, q' \in Q$ について $t \rightarrow^*_{\mathcal{A}} q$ かつ $t \rightarrow^*_{\mathcal{A}} q'$ を満たすならば $q = q'$ であるとき，$\mathcal{A}$ は決定的であるという．任意の基底項 $t \in \mathcal{T}(\mathcal{F} \cup \mathcal{G})$ と任意の遷移規則 $f(q_1, \ldots, q_n) \xrightarrow{\phi} q \in \Delta$ について，$t \rightarrow^*_{\mathcal{A}} f(q_1, \ldots, q_n) \xrightarrow{\phi} q$ を満たすときに任意の位置 $p \in CPos(\phi)$ について $p \in \mathcal{P}os(t)$ かつ $t|_p \in \mathcal{T}(\mathcal{G})$ であるならば，$\mathcal{A}$ は制約完全 (constraint-complete) であるという．

3. 制約付き項のインスタンスを受理する制約付き木オートマトン構成手法

本節では，制約付き項とそのインスタンスを定義し，制約付き項の有限集合が与えられたときにそのインスタンスの集合を受理する制約付き TA を構成する方法を提案する．また，この手法で構成される制約付き TA の性質について述べる．まずは，制約付き項を定義する．

定義 1. $\mathcal{T}(\mathcal{F} \cup \mathcal{G}, \mathcal{V})$ 中の線形項 t と，$L(\mathcal{G}, \mathcal{P}, \mathcal{V}ar(t))$ 中の論理式 ϕ の対 (t, ϕ) を制約付き項と呼び，$t \llbracket \phi \rrbracket$ と表す．

$t \llbracket \phi \rrbracket$ のインスタンスの集合 $G(t \llbracket \phi \rrbracket)$ を次のように定義する．

$$G(t \llbracket \phi \rrbracket) = \left\{ t\theta \;\middle|\; \begin{array}{l} \mathcal{R}an(\theta|_{f\mathcal{V}(\phi)}) \subseteq \mathcal{T}(\mathcal{G}), \\ t\theta \in \mathcal{T}(\mathcal{F} \cup \mathcal{G}),\ \mathcal{M} \models \phi\theta \end{array} \right\}$$

また，制約付き項の有限集合 T のインスタンスの集合 $G(T)$ を次のように定義する：$G(T) = \bigcup_{t \llbracket \phi \rrbracket \in T} G(t \llbracket \phi \rrbracket)$.

$G(T)$ を受理する制約付き TA の構成法を示す前に，制約を考慮しない場合，つまり，項の集合が与えられたときに，そのインスタンスを受理し，完全性・決定性を満たす TA の構成法 [3] を紹介する．以下に全体のアルゴリズムを示す．

線形項の有限集合 T に対して，以下のように TA $\mathcal{A} = (Q, Q_f, \Delta)$ を構成する．

(1) T の項の変数を □ に置き換え，その集合を $T_\square$ とする．

(2) $T_\square$ の項の部分項を $T_\square$ に追加する．

(3) $T_\square$ の任意の 2 つの項のパターンを組み合わせて，より具体的にした項のパターンを $T_\square$ に追加する（例えば，$\mathsf{f}(\square, \mathsf{s}(\square))$ と $\mathsf{f}(\mathsf{s}(\square), \square)$ から $\mathsf{f}(\mathsf{s}(\square), \mathsf{s}(\square))$ を追加）．

(4) これまでで構成した項のパターンをラベルに持つ状態を構成する：$Q = \{q_t \mid t \in T_\square\}$.

(5) 状態中で，T の項が照合できるパターンを持つ状態を最終状態にする．

(6) パターンを出来るだけ具体的に保持するように遷移規則を構成する．

本論文で提案する相違点とは 1 つは，項のパターンに加えて制約付きパターンを考えることである．その中で □ と ■ の 2 種類の記号を使用することで，解釈可能な項と解釈不可能な項の区別をする．これによって制約完全性を持つように構成することを可能にする．また，同じ項のパターンをラベルに持った状態であっても，その状態に至る直前の遷移時に満たす制約によって受理すべきものと受理すべきでないものが分かれる．これは状態のラベルであるパターンから判別できない．よって，最終状態とそうでない状態を区別して表し，遷移時の制約によって遷移先の状態を決定する．

以下で，上述のアイディアに沿って，形式的な定義を与える．初めに，制約付き項の項中の変数を特別な定数記号 □, ■ に，制約中の変数をその変数が項中で現れる位置に置き換える．以降では，□ を解釈可能な項のパターンを表す特別な記号，■ を解釈不可能な項のパターンを表す特別な記号とする．

定義 2. $\square, \blacksquare$ ($\notin \mathcal{F} \cup \mathcal{G}$) を定数記号とする．制約付き項 $t \llbracket \phi \rrbracket$ について，$Rep_{\square,\blacksquare}(t \llbracket \phi \rrbracket)$ を次の集合と定義する．

$$\left\{ t\theta \llbracket \phi\theta' \rrbracket \;\middle|\; \begin{array}{l} \theta \in \{\{x \mapsto \square \mid x \in \mathcal{V}ar(t) \cap f\mathcal{V}(\phi)\} \cup \\ \{x \mapsto v \mid x \in \mathcal{V}ar(t) \setminus f\mathcal{V}(\phi)\} \mid v \in \{\square, \blacksquare\}\}, \\ \theta' = \{x \mapsto p \mid x \in \mathcal{V}ar(t) \cap f\mathcal{V}(\phi),\ t|_p \equiv x\} \end{array} \right\}$$

また制約付き項の有限集合 T について，$Rep_{\square,\blacksquare}(T)$ を次のように定義する：$Rep_{\square,\blacksquare}(T) = \{Rep_{\square,\blacksquare}(t \llbracket \phi \rrbracket) \mid t \llbracket \phi \rrbracket \in T\}$.

$\mathcal{F}, \mathcal{G}, \square, \blacksquare$ からなる基底項 t を項のパターン，項のパターンと $L(\mathcal{G}, \mathcal{P}, \langle \mathcal{N}^+ \rangle)$ 中の制約 ϕ（ただし $CPos(\phi) \subseteq \mathcal{P}os(t)$ かつ $\forall \pi \in CPos(\phi).\ t|_\pi \equiv \square$）の対 (t, ϕ) を制約付きパターンと呼び，$t \llbracket \phi \rrbracket$ と記す．

制約付きパターンは，項のパターン中の □ に解釈可能な基底項，■ に解釈不可能な基底項が入り，対となる制約を満たすような基底項の集合を表す．

例 3. $\mathcal{F} = \{\mathsf{g}(,)\}$ $\mathcal{G} = \{\mathsf{0}, \mathsf{s}()\}$ $\mathcal{P} = \{\leq, \geq\}$ とする．また構造 $\mathcal{M}$ によって関数記号 $\mathsf{0}, \mathsf{s}()$ はそれぞれ整数の 0 と後者関数に解釈され，述語記号 $\leq, \geq$ は記号通りの整数上の順序関係に解釈されるとする．このとき $T = \{\mathsf{g}(x, y) \llbracket x \leq \mathsf{0} \rrbracket, \mathsf{g}(\mathsf{s}(x), y) \llbracket x \geq \mathsf{0} \rrbracket\}$ とすると，$Rep_{\square,\blacksquare}(T)$ は次の集合になる．

$$\left\{ \begin{array}{ll} \mathsf{g}(\square, \square) \llbracket \langle 1 \rangle \leq \mathsf{0} \rrbracket, & \mathsf{g}(\square, \blacksquare) \llbracket \langle 1 \rangle \leq \mathsf{0} \rrbracket, \\ \mathsf{g}(\mathsf{s}(\square), \blacksquare) \llbracket \langle 1 \cdot 1 \rangle \geq \mathsf{0} \rrbracket, & \mathsf{g}(\mathsf{s}(\square), \blacksquare) \llbracket \langle 1 \cdot 1 \rangle \geq \mathsf{0} \rrbracket \end{array} \right\}$$

次に，制約付きパターン中の項のパターンの部分項にあたる制約付きパターンを得るための関数を定義する．

定義 4. 制約付きパターンの有限集合 U について，$\mathit{Sub}(U)$ を次のように定義する．

$$\mathit{Sub}(U) = U \cup \{u|_p \llbracket \top \rrbracket \mid u \llbracket \phi \rrbracket \in U,\ p \in \mathcal{P}os(u) \setminus \{\varepsilon\}\}$$

例 5. 例 3 と同様の T に対して，$\mathit{Sub}(\mathit{Rep}_{\square,\blacksquare}(T)) = \mathit{Rep}_{\square,\blacksquare}(T) \cup \{\square \llbracket \top \rrbracket,\ \blacksquare \llbracket \top \rrbracket,\ \mathsf{s}(\square) \llbracket \top \rrbracket\}$ となる．

次に，制約付きパターンの順序関係を定義する．これは，パターンの具体さを比較する順序である．

定義 6. $\mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\})$ 中の項のパターンについて，順序関係 $\sqsubseteq$ を次のように再帰的に定義する．

- $\square \sqsubseteq t$ （ただし,$t \in \mathcal{T}(\mathcal{G} \cup \{\square\})$）
- $\blacksquare \sqsubseteq t$ （ただし,$t \in \mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\}) \setminus \mathcal{T}(\mathcal{G} \cup \{\square\})$）
- $f(t_1, \ldots, t_n) \sqsubseteq f(t'_1, \ldots, t'_n)$ （ただし,$f \in \mathcal{F} \cup \mathcal{G}$, $t_1, \ldots, t_n, t'_1, \ldots, t'_n \in \mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\})$, $\forall i.\ t_i \sqsubseteq t'_i$）

また，$t \sqsubseteq t'$ かつ $t' \sqsubseteq t$ のとき $t \simeq t'$ と表し，$t \sqsubseteq t'$ かつ $t \neq t'$ のとき $t \sqsubset t'$ と表す．

$L(\mathcal{G}, \mathcal{P}, X)$ 中の論理式 ϕ, ϕ' について，$\mathit{fv}(\phi) \subseteq \mathit{fv}(\phi')$ かつ $\phi' \Rightarrow \phi$ が恒真のとき，$\phi \sqsubseteq \phi'$ であるとする．項のパターンの順序関係と同様にして $\phi \simeq \phi'$，$\phi \sqsubset \phi'$ も定義する．

制約付きパターン $t \llbracket \phi \rrbracket$，$t' \llbracket \phi' \rrbracket$ について，$t \sqsubseteq t'$ かつ $\phi \sqsubseteq \phi'$ のとき，$t \llbracket \phi \rrbracket \sqsubseteq t' \llbracket \phi' \rrbracket$ であるとする．項のパターンの順序関係と同様にして $t \llbracket \phi \rrbracket \simeq t' \llbracket \phi' \rrbracket$，$t \llbracket \phi \rrbracket \sqsubset t' \llbracket \phi' \rrbracket$ も定義する．

次に，2 つの項のパターンを組み合わせてより具体的な項のパターンを構成するための演算を定義する．

定義 7. $\mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\})$ 中の項のパターンについて，二項演算子 $\sqcap$ を次のように再帰的に定義する．

- $\square \sqcap u = u$ （ただし,$u \in \mathcal{T}(\mathcal{G} \cup \{\square\})$）
- $t \sqcap \square = t$ （ただし,$t \in \mathcal{T}(\mathcal{G} \cup \{\square\})$）
- $\blacksquare \sqcap u = u$ （ただし,$u \in \mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\}) \setminus \mathcal{T}(\mathcal{G} \cup \{\square\})$）
- $t \sqcap \blacksquare = t$ （ただし,$t \in \mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\}) \setminus \mathcal{T}(\mathcal{G} \cup \{\square\})$）
- $f(t_1, \ldots, t_n) \sqcap f(u_1, \ldots, u_n) = f(t_1 \sqcap u_1, \ldots, t_n \sqcap u_n)$

（ただし,$f \in \mathcal{F} \cup \mathcal{G}$, $t_1, \ldots, t_n, u_1, \ldots, u_n \in \mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\})$）

次に，制約付きパターンの有限集合について，任意の基底項をただ 1 つの制約付きパターンに置き換えられるように制約付きパターンを追加するための関数を定義する．

定義 8. 制約付きパターンの有限集合 U について，$\mathit{Inst}(U)$ は次の条件を満たす $\simeq$ を法とした最小の集合と定義する．

- $\square \llbracket \top \rrbracket, \blacksquare \llbracket \top \rrbracket \in \mathit{Inst}(U)$.
- $t \llbracket \phi \rrbracket \in U$ かつ ϕ が充足可能ならば，$t \llbracket \phi \rrbracket \in \mathit{Inst}(U)$.
- $t \llbracket \phi \rrbracket \in U$ かつ $\neg\phi$ が充足可能ならば，$t \llbracket \neg\phi \rrbracket \in \mathit{Inst}(U)$.
- $t_1 \llbracket \phi_1 \rrbracket, t_2 \llbracket \phi_2 \rrbracket \in \mathit{Inst}(U)$ かつ $\phi_1 \wedge \phi_2$ が充足可能ならば，$t_1 \sqcap t_2 \llbracket \phi_1 \wedge \phi_2 \rrbracket \in \mathit{Inst}(U)$.

例 9. 例 3 の $(\mathcal{F}, \mathcal{G}, \mathcal{P}, \mathcal{M})$ と T に対して，Inst で $\mathit{Sub}(\mathit{Rep}_{\square,\blacksquare}(T))$ に次の集合が追加される．

$$\left\{\begin{array}{ll}
\mathsf{g}(\mathsf{s}(\square), \square) \llbracket \langle 1 \cdot 1 \rangle \geqq 0 \rrbracket, & \mathsf{g}(\mathsf{s}(\square), \square) \llbracket \neg\langle 1 \cdot 1 \rangle \geqq 0 \rrbracket, \\
\mathsf{g}(\mathsf{s}(\square), \blacksquare) \llbracket \langle 1 \cdot 1 \rangle \geqq 0 \rrbracket, & \mathsf{g}(\mathsf{s}(\square), \blacksquare) \llbracket \neg\langle 1 \cdot 1 \rangle \geqq 0 \rrbracket, \\
\multicolumn{2}{l}{\mathsf{g}(\mathsf{s}(\square), \square) \llbracket \langle 1 \rangle \leqq 0 \wedge \langle 1 \cdot 1 \rangle \geqq 0 \rrbracket,} \\
\multicolumn{2}{l}{\mathsf{g}(\mathsf{s}(\square), \square) \llbracket \langle 1 \rangle \leqq 0 \wedge \neg\langle 1 \cdot 1 \rangle \geqq 0 \rrbracket,} \\
\multicolumn{2}{l}{\mathsf{g}(\mathsf{s}(\square), \square) \llbracket \neg\langle 1 \rangle \leqq 0 \wedge \langle 1 \cdot 1 \rangle \geqq 0 \rrbracket,} \\
\multicolumn{2}{l}{\mathsf{g}(\mathsf{s}(\square), \square) \llbracket \neg\langle 1 \rangle \leqq 0 \wedge \neg\langle 1 \cdot 1 \rangle \geqq 0 \rrbracket,} \\
\multicolumn{2}{l}{\mathsf{g}(\mathsf{s}(\square), \blacksquare) \llbracket \langle 1 \rangle \leqq 0 \wedge \langle 1 \cdot 1 \rangle \geqq 0 \rrbracket,} \\
\multicolumn{2}{l}{\mathsf{g}(\mathsf{s}(\square), \blacksquare) \llbracket \langle 1 \rangle \leqq 0 \wedge \neg\langle 1 \cdot 1 \rangle \geqq 0 \rrbracket,} \\
\multicolumn{2}{l}{\mathsf{g}(\mathsf{s}(\square), \blacksquare) \llbracket \neg\langle 1 \rangle \leqq 0 \wedge \langle 1 \cdot 1 \rangle \geqq 0 \rrbracket,} \\
\multicolumn{2}{l}{\mathsf{g}(\mathsf{s}(\square), \blacksquare) \llbracket \neg\langle 1 \rangle \leqq 0 \wedge \neg\langle 1 \cdot 1 \rangle \geqq 0 \rrbracket}
\end{array}\right\}$$

次に，制約付き項のインスタンスを受理する制約付き TA の構成法を定義する．

定義 10. 制約付き項の有限集合 T について，制約付き TA $\mathcal{A} = (Q, Q_f, \Delta)$ を次のように定義する．

$$Q_f = \left\{ \widetilde{q}_t \;\middle|\; \begin{array}{l} t \llbracket \phi \rrbracket \in \mathit{Inst}(\mathit{Sub}(\mathit{Rep}_{\square,\blacksquare}(T))), \\ \exists t' \llbracket \phi' \rrbracket \in \mathit{Rep}_{\square,\blacksquare}(T).\ t' \llbracket \phi' \rrbracket \sqsubseteq t \llbracket \phi \rrbracket \end{array} \right\}$$

$$Q = Q_f \cup \left\{ q_t \;\middle|\; \begin{array}{l} t \llbracket \phi \rrbracket \in \mathit{Inst}(\mathit{Sub}(\mathit{Rep}_{\square,\blacksquare}(T))), \\ \nexists t' \llbracket \phi' \rrbracket \in \mathit{Rep}_{\square,\blacksquare}(T).\ t' \llbracket \phi' \rrbracket \sqsubseteq t \llbracket \phi \rrbracket \end{array} \right\}$$

$$\Delta = \left\{ f(\dot{q}_{t_1}, \ldots, \dot{q}_{t_n}) \xrightarrow{\phi} \dot{q}_t \;\middle|\; \begin{array}{l} f \in \mathcal{F} \cup \mathcal{G},\ \dot{q}_t, \dot{q}_{t_1}, \ldots, \dot{q}_{t_n} \in Q, \\ t \llbracket \phi \rrbracket \in \mathit{Inst}(\mathit{Sub}(\mathit{Rep}_{\square,\blacksquare}(T))), \\ t \sqsubseteq f(t_1, \ldots, t_n), \\ \nexists t' \llbracket \phi' \rrbracket \in \mathit{Inst}(\mathit{Sub}(\mathit{Rep}_{\square,\blacksquare}(T))). \\ t' \sqsubseteq f(t_1, \ldots, t_n) \wedge t \llbracket \phi \rrbracket \sqsubset t' \llbracket \phi' \rrbracket \end{array} \right\}$$

ただし，状態を $\dot{q}_t$ と書いた場合，q_t もしくは $\widetilde{q}_t$ のどちらかの形であることを表す．

定理 11. 定義 10 によって構成される制約付き TA $\mathcal{A}$ は制約完全・完全・決定的であり，$L(\mathcal{A}) = G(T)$ である．

略証. $\mathit{Rep}_{\square,\blacksquare}$，$\mathit{Sub}$ の定義，また Inst の定義に関する帰納法によって制約完全であることを証明でき，$\mathcal{T}(\mathcal{F} \cup \mathcal{G})$ 中の基底項 t について，t の構造に関する帰納法によって完全であることを証明できる．また，$\mathcal{T}(\mathcal{F} \cup \mathcal{G})$ 中の基底項 t と Q 中の状態 q_1，q_2 について，$t \to^m_{\mathcal{A}} q_1$ かつ $t \to^n_{\mathcal{A}} q_2$ ならば $q_1 = q_2$ となることを $m + n$ に関する帰納法で証明でき，この補題より決定的であることを証明できる．$\mathcal{A}$ が制約完全であることと完全であることを利用して $L(\mathcal{A}) = G(T)$ となることを証明できる． □

例 12. $\mathcal{F} = \{\mathsf{f}()\}$ とし，例 3 の $\mathcal{G}, \mathcal{P}, \mathcal{M}$ を考え，$T = \{\mathsf{f}(x) \llbracket x \leqq 0 \rrbracket, \mathsf{f}(\mathsf{s}(x)) \llbracket x \geqq 0 \rrbracket\}$ とする．定義 10 より制約完全性・完全性・決定性を満たす次の制約付き TA $\mathcal{A} = (Q, Q_f, \Delta)$ が構成される：$Q = \{q_{\square}, q_{\blacksquare}, q_{\mathsf{s}(\square)}, q_{\mathsf{f}(\square)}, \widetilde{q}_{\mathsf{f}(\square)}, q_{\mathsf{f}(\mathsf{s}(\square))}, \widetilde{q}_{\mathsf{f}(\mathsf{s}(\square))}\}$，$Q_f = \{\widetilde{q}_{\mathsf{f}(\square)}, \widetilde{q}_{\mathsf{f}(\mathsf{s}(\square))}\}$，

$$\Delta = \left\{\begin{array}{l}
0 \to q_{\square}, \quad \mathsf{s}(q_{\square}) \to q_{\mathsf{s}(\square)}, \quad \mathsf{s}(q_{\blacksquare}) \to q_{\blacksquare}, \\
\mathsf{s}(q_{\mathsf{s}(\square)}) \to q_{\mathsf{s}(\square)}, \quad \mathsf{s}(q_{\mathsf{f}(\square)}) \to q_{\blacksquare}, \quad \mathsf{s}(\widetilde{q}_{\mathsf{f}(\square)}) \to q_{\blacksquare}, \\
\mathsf{s}(q_{\mathsf{f}(\mathsf{s}(\square))}) \to q_{\blacksquare}, \quad \mathsf{s}(\widetilde{q}_{\mathsf{f}(\mathsf{s}(\square))}) \to q_{\blacksquare}, \quad \mathsf{f}(q_{\blacksquare}) \to q_{\blacksquare}, \\
\mathsf{f}(q_{\square}) \xrightarrow{\langle 1 \rangle \leqq 0} \widetilde{q}_{\mathsf{f}(\square)}, \quad \mathsf{f}(q_{\square}) \xrightarrow{\neg\langle 1 \rangle \leqq 0} q_{\mathsf{f}(\square)}, \\
\mathsf{f}(q_{\mathsf{s}(\square)}) \xrightarrow{\langle 1 \rangle \leqq 0 \wedge \langle 1 \cdot 1 \rangle \geqq 0} \widetilde{q}_{\mathsf{f}(\mathsf{s}(\square))}, \quad \mathsf{f}(q_{\mathsf{f}(\square)}) \to q_{\blacksquare}, \\
\mathsf{f}(q_{\mathsf{s}(\square)}) \xrightarrow{\langle 1 \rangle \leqq 0 \wedge \neg\langle 1 \cdot 1 \rangle \geqq 0} \widetilde{q}_{\mathsf{f}(\mathsf{s}(\square))}, \quad \mathsf{f}(\widetilde{q}_{\mathsf{f}(\square)}) \to q_{\blacksquare}, \\
\mathsf{f}(q_{\mathsf{s}(\square)}) \xrightarrow{\neg\langle 1 \rangle \leqq 0 \wedge \langle 1 \cdot 1 \rangle \geqq 0} \widetilde{q}_{\mathsf{f}(\mathsf{s}(\square))}, \quad \mathsf{f}(q_{\mathsf{f}(\mathsf{s}(\square))}) \to q_{\blacksquare}, \\
\mathsf{f}(q_{\mathsf{s}(\square)}) \xrightarrow{\neg\langle 1 \rangle \leqq 0 \wedge \neg\langle 1 \cdot 1 \rangle \geqq 0} q_{\mathsf{f}(\mathsf{s}(\square))}, \mathsf{f}(\widetilde{q}_{\mathsf{f}(\mathsf{s}(\square))}) \to q_{\blacksquare}
\end{array}\right\}$$

4. 十分完全性判定への応用

本節では，前節で構成した制約付き TA の応用として，制約付き項書換え系 [5], [6] の十分完全性の判定法について述べる．

制約付き項書換え系に関する用語や概念を文献 [5], [6] に基づき定義する．$(\mathcal{F},\mathcal{P},\mathcal{G},\mathcal{M})$ 上の制約付き項書換え系は，$l \to r$ $[\![\phi]\!]$ の形の制約付き書換え規則の集合である．ただし，l, r は $\mathcal{T}(\mathcal{F}\cup\mathcal{G},\mathcal{V})$ 中の項，ϕ は $L(\mathcal{G},\mathcal{P},\mathcal{V}ar(l))$ 中の論理式であり，$l \notin \mathcal{V}$, $\mathcal{V}ar(l) \supseteqq \mathcal{V}ar(r)$ を満たす．R を $(\mathcal{F},\mathcal{G},\mathcal{P},\mathcal{M})$ 上の制約付き書換え規則の有限集合とする．R で定められる書換え関係 $\to_R$ を $\to_R = \{(C[l\sigma], C[r\sigma]) \mid l \to r\ [\![\phi]\!] \in R,\ C[\,] \in \mathcal{T}_\square(\mathcal{F}\cup\mathcal{G},\mathcal{V}), f_\mathcal{V}(\phi) \subseteqq \mathcal{D}om(\sigma),\ \mathcal{R}an(\sigma|_{f_\mathcal{V}(\phi)}) \subseteqq \mathcal{T}(\mathcal{G}),\ \phi\sigma$ は真 $\}$ と定義する．制約付き項書換え系 R が十分完全性を持つとは，すべての基底項 $s \in \mathcal{T}(\mathcal{F}\cup\mathcal{G})$ に対して $s \xrightarrow{*}_R t \in \mathcal{T}(\mathcal{G})$ を満たす t が存在することである．

次に，制約付き項書換え系が左線形である場合の十分完全性の必要十分条件を示す．

命題 13. T を制約付き項の有限集合とし，$Red(T)$ を次のように定める：$Red(T) = \{C[t] \mid t \in G(T),\ C[\,] \in \mathcal{T}_\square(\mathcal{F}\cup\mathcal{G})\}$．

R は左線形な制約付き項書換え系とする．制約付き項の集合 T を $T_R = \{t\ [\![\phi]\!] \mid t \to u\ [\![\phi]\!] \in R\}$ とする．このとき $\overline{Red(T_R)} \subseteqq \mathcal{T}(\mathcal{G})$ を満たすことは，R が $\mathcal{G}$ に関して十分完全であることの必要十分条件である．

上の定理より，制約付き項書換え系 R が $\mathcal{G}$ に関して十分完全であることの十分条件は，$\overline{Red(T_R)}$ を受理する制約付き TA の最終状態に付随した項のパターンがすべて $\mathcal{T}(\mathcal{G}\cup\{\square\})$ に含まれることである．

$Red(T)$ を受理する制約付き TA は，$G(T)$ を受理する制約付き TA を利用して構成することができる．

定義 14. 制約付き項の有限集合 T を入力として，定義 10 によって構成される制約付き TA を $\mathcal{A} = (Q, Q_f, \Delta)$ とする．このとき，$(\mathcal{F},\mathcal{G},\mathcal{P},\mathcal{M})$ 上の制約付き TA $\mathcal{A}' = (Q', Q'_f, \Delta')$ を図 1 のように定義する．

$Red(T)$ は $G(T)$ を文脈に閉じるように拡大した集合であるので，定義 14 の制約付き TA によって受理される項を部分項に持つ項を受理するように構成した．定義 14 によって構成された制約付き TA の受理集合 $L(\mathcal{A}')$ は $Red(T)$ と等価であり，構成される制約付き TA は制約完全性，完全性，決定性を満たす．

例 15. 例 12 の $(\mathcal{F},\mathcal{G},\mathcal{P},\mathcal{M})$ について，制約付き項書換え系 $R = \{\mathsf{f}(x) \to 0\ [\![x \leqq 0]\!],\ \mathsf{f}(\mathsf{s}(x)) \to \mathsf{s}(\mathsf{f}(x))\ [\![x \geqq 0]\!]\}$ を考える．定義 14 より，制約付き項の集合 T_R は例 12 の T と一致し，制約完全性・完全性・決定性を満たす制約付き TA $\mathcal{A}' = (Q', Q'_f, \Delta')$ は図 2 のように構成される．

この制約付き TA は決定性・完全性を満たすので，$\overline{Red(T)}$ を受理する制約付き TA は $(Q', Q' \setminus Q'_f, \Delta')$ と構成でき，そのときの最終状態は次のようになる：$Q' \setminus Q'_f = \{q_\square,\ q_\blacksquare,\ q_{\mathsf{s}(\square)},\ q_{\mathsf{f}(\square)},\ q_{\mathsf{f}(\mathsf{s}(\square))}\}$．これより最終状態に付随した項のパターンとして 5 つ得られるが，そのうちの $\blacksquare, \mathsf{f}(\square), \mathsf{f}(\mathsf{s}(\square))$ は $\mathcal{T}(\mathcal{G}\cup\{\square\})$ に含まれない．よって，十分完全性の十分条件は満たさない．

$$Q'_f = \{\widetilde{q}_\square,\ \widetilde{q}_\blacksquare\} \qquad Q' = (Q \cup Q'_f) \setminus Q_f$$

$$\begin{aligned}
\Delta' = & \left\{ f(q_{t_1},\dots,q_{t_n}) \xrightarrow{\phi} q_t \;\middle|\; \begin{array}{l} f \in \mathcal{F}\cup\mathcal{G},\ q_t, q_{t_1},\dots,q_{t_n} \in Q \setminus Q_f, \\ f(q_{t_1},\dots,q_{t_n}) \xrightarrow{\phi} q_t \in \Delta \end{array} \right\} \\
\cup & \left\{ f(q_{t_1},\dots,q_{t_n}) \xrightarrow{\phi} \widetilde{q}_{t'} \;\middle|\; \begin{array}{l} f \in \mathcal{F}\cup\mathcal{G},\ \widetilde{q}_{t'} \in Q'_f, \\ q_{t_1},\dots,q_{t_n} \in Q \setminus Q_f, \\ \exists \widetilde{q}_t \in Q_f.\ t' \sqsubseteq t \wedge \\ \qquad f(q_{t_1},\dots,q_{t_n}) \xrightarrow{\phi} \widetilde{q}_t \in \Delta \end{array} \right\} \\
\cup & \left\{ f(\dot{q}_{t'_1},\dots,\dot{q}_{t'_n}) \xrightarrow{\top} \widetilde{q}_{t'} \;\middle|\; \begin{array}{l} f \in \mathcal{F}\cup\mathcal{G},\ t' \sqsubseteq f(t'_1,\dots,t'_n), \\ \widetilde{q}_{t'} \in Q'_f,\ \dot{q}_{t'_1},\dots,\dot{q}_{t'_n} \in Q', \\ f(\dot{q}_{t_1},\dots,\dot{q}_{t_n}) \xrightarrow{\phi} \dot{q}_t \in \Delta, \\ \exists i.\ \dot{q}_{t'_i} \in Q'_f, \\ \quad \forall j.\ ((\dot{q}_{t'_j} \notin Q_f \wedge \dot{q}_{t_j} = \dot{q}_{t'_j}) \\ \qquad \vee (\dot{q}_{t_j} \in Q_f \wedge \dot{q}_{t'_j} \in Q'_f \\ \qquad\quad \wedge \dot{t}'_j \sqsubseteq t_j)) \end{array} \right\}
\end{aligned}$$

図 1　$Red(T)$ を受理する制約付き TA の構成法

$$Q' = \{q_\square, q_\blacksquare, q_{\mathsf{s}(\square)}, q_{\mathsf{f}(\square)}, q_{\mathsf{f}(\mathsf{s}(\square))}, \widetilde{q}_\square, \widetilde{q}_\blacksquare\} \quad Q'_f = \{\widetilde{q}_\square, \widetilde{q}_\blacksquare\}$$

$$\Delta' = \left\{ \begin{array}{lll}
0 \to q_\square, & \mathsf{s}(q_\square) \to q_{\mathsf{s}(\square)}, & \mathsf{s}(q_\blacksquare) \to q_\blacksquare, \\
\mathsf{s}(q_{\mathsf{s}(\square)}) \to q_{\mathsf{s}(\square)}, & \mathsf{s}(q_{\mathsf{f}(\square)}) \to q_\blacksquare, & \mathsf{s}(q_{\mathsf{f}(\mathsf{s}(\square))}) \to q_\blacksquare, \\
\mathsf{s}(\widetilde{q}_\square) \to \widetilde{q}_\square, & \mathsf{s}(\widetilde{q}_\blacksquare) \to \widetilde{q}_\blacksquare, & \mathsf{f}(q_\blacksquare) \to q_\blacksquare, \\
\mathsf{f}(q_\square) \xrightarrow{\langle 1\rangle \leqq 0} \widetilde{q}_\square, & \mathsf{f}(q_\square) \xrightarrow{\neg\langle 1\rangle \leqq 0} q_{\mathsf{f}(\square)}, & \\
\mathsf{f}(q_{\mathsf{s}(\square)}) \xrightarrow{\langle 1\rangle \leqq 0 \wedge \langle 1\cdot 1\rangle \geqq 0} \widetilde{q}_\blacksquare, & & \mathsf{f}(q_{\mathsf{f}(\square)}) \to q_\blacksquare, \\
\mathsf{f}(q_{\mathsf{s}(\square)}) \xrightarrow{\langle 1\rangle \leqq 0 \wedge \neg\langle 1\cdot 1\rangle \geqq 0} \widetilde{q}_\blacksquare, & & \mathsf{f}(q_{\mathsf{f}(\mathsf{s}(\square))}) \to q_\blacksquare, \\
\mathsf{f}(q_{\mathsf{s}(\square)}) \xrightarrow{\neg\langle 1\rangle \leqq 0 \wedge \langle 1\cdot 1\rangle \geqq 0} \widetilde{q}_\blacksquare, & & \mathsf{f}(\widetilde{q}_\square) \to \widetilde{q}_\blacksquare, \\
\mathsf{f}(q_{\mathsf{s}(\square)}) \xrightarrow{\neg\langle 1\rangle \leqq 0 \wedge \neg\langle 1\cdot 1\rangle \geqq 0} q_{\mathsf{f}(\mathsf{s}(\square))}, & & \mathsf{f}(\widetilde{q}_\blacksquare) \to \widetilde{q}_\blacksquare
\end{array} \right\}$$

図 2　例 15 で構成される制約付き TA の状態と遷移規則

実際には，例 15 の R は十分完全性を持つ．今回判定に失敗した原因は，制約付き TA $\mathcal{A}'$ がどのような基底項の入力においても到達することのない状態を含んでいることが原因である．次節では，この問題を解決するために，そのような状態を削除する手法を提案する．

5. 制約付き木オートマトンの状態数削減手法

本節では，制約付き TA 中にどのような基底項の入力においても到達することのない状態が含まれている場合にそれを出来る限り削減する手法を提案する．制約の無い TA においては，どんな項の入力に対しても到達しない状態を完全に削除する手法が存在する [3]．そのアルゴリズムでは，まず定数から遷移できる状態集合を作り，以下，その集合中の状態を関数の引数にとることで遷移できる状態を随時集合に追加するという作業を繰り返す．そして，最終的に追加されなかった状態を削除する．

制約付き TA においても，制約を無視することで [3] のアルゴリズムを適用できるが，関数記号と状態だけでは制約を満たすかどうかがわからないことから完全ではない．以下では TA の状態数削減手法を拡張することで，制約付き TA の状態数削減手法を提案する．ただしこの手法は，定義 10 や定義 14 に

よって構成された制約付き TA に対しての手法である．まず遷移規則を逆に辿ることで，状態のラベルである項のパターンを具体化する．次に，遷移規則中の制約に出現する位置を項のパターンに置き換え，その制約が充足不能であればその遷移規則を削除する．その結果の制約付き TA に対して制約を無視することで TA の状態数削減手法を適用する．以上の作業を状態数に変化が無くなるまで繰り返す．

以下で，形式的な定義を与える．まず，2 つの項のパターンに共通するパターンを構成する演算子を定義する．

定義 16. $\mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\})$ 中の項のパターンについて，左結合の二項演算子 $\sqcup$ を次のように再帰的に定義する．

- $t \sqcup \square = \square$ （ただし,$t \in \mathcal{T}(\mathcal{G} \cup \{\square\})$）
- $\square \sqcup u = \square$ （ただし,$u \in \mathcal{T}(\mathcal{G} \cup \{\square\})$）
- $t \sqcup \blacksquare = \blacksquare$ （ただし,$t \in \mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\}) \setminus \mathcal{T}(\mathcal{G} \cup \{\square\})$）
- $\blacksquare \sqcup u = \blacksquare$ （ただし,$u \in \mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\}) \setminus \mathcal{T}(\mathcal{G} \cup \{\square\})$）
- $f(t_1, \ldots, t_n) \sqcup g(u_1, \ldots, u_m) = \square$ （ただし,$f(t_1, \ldots, t_n), g(u_1, \ldots, u_m) \in \mathcal{T}(\mathcal{G} \cup \{\square\})$）
- $f(t_1, \ldots, t_n) \sqcup g(u_1, \ldots, u_m) = \blacksquare$ （ただし,$f(t_1, \ldots, t_n)$, $g(u_1, \ldots, u_m) \in \mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\}) \setminus \mathcal{T}(\mathcal{G} \cup \{\square\})$）
- $f(t_1, \ldots, t_n) \sqcup f(u_1, \ldots, u_n) = f(t_1 \sqcup u_1, \ldots, t_n \sqcup u_n)$ （ただし,$f \in \mathcal{F} \cup \mathcal{G}, t_1, \ldots, t_n, u_1, \ldots, u_n \in \mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\})$）

定義 17. $U = \{u_1, \ldots, u_n\}$ を $\mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\})$ 中の項のパターンの集合とする．このとき，$Gen(U)$ を以下のように定義する：$Gen(U) = u_1 \sqcup \cdots \sqcup u_n$．

次に，遷移規則によって状態のラベルを具体化するための関数を定義する．

定義 18. Δ は制約付き TA の遷移規則，q を Q 中の状態とする．このとき，$Source_\Delta(q)$ を次のように定義する．

$$Source_\Delta(q) = \{f(t_1, \ldots, t_n) \mid f(q_{t_1}, \ldots, q_{t_n}) \xrightarrow{\phi} q \in \Delta\}$$

次に，パターンから $\square, \blacksquare$ を含まない項への変換を定義する．

定義 19. t を $\mathcal{T}(\mathcal{F} \cup \mathcal{G} \cup \{\square, \blacksquare\})$ 中の項のパターン，π をポジションとする．このとき，$Rep_\mathcal{V}(t, \pi)$ を次のように再帰的に定義する：$Rep_\mathcal{V}(\square, \pi) = x_\pi$，$Rep_\mathcal{V}(\blacksquare, \pi) = x_\pi$, $Rep_\mathcal{V}(f(t_1, \ldots, t_n), \pi) = f(Rep_\mathcal{V}(t_1, \pi \cdot 1), \ldots, Rep_\mathcal{V}(t_n, \pi \cdot n))$．ただし，$x_\pi \in \mathcal{V}$ である．

(Q, Q_f, Δ) を $(\mathcal{F}, \mathcal{G}, \mathcal{P}, \mathcal{M})$ 上の制約付き TA とする．このとき Q 中で使用されない状態を出来るだけ削除する手法として以下のアルゴリズムを提案する．

入力: 制約付き TA (Q, Q_f, Δ) **出力:** 制約付き TA (Q', Q'_f, Δ')

(1) $Q' := Q$，$Q'_f := Q_f$，$\Delta' := \Delta$ とする．

(2) すべての状態 $q_t \in Q$ について，$Gen(Source_\Delta(q_t))$ が定義されるならば Q' 中の q_t を $q_{t'}$ に置き換える．ただし，$t' \equiv Gen(Source_\Delta(q_t))$ とする．状態の置き換えに合わせて Q'_f と Δ' も更新する．

(3) すべての遷移規則 $f(q_{t_1}, \ldots, q_{t_n}) \xrightarrow{\phi} q_t \in \Delta$ について，ϕ 中のすべての位置 π を $Rep_\mathcal{V}(f(t_1, \ldots, t_n), \varepsilon)|_\pi^\mathcal{M}$ に置き換えた論理式が充足不能ならば，その遷移規則を Δ' から削除する．

(4) 制約を無視し，TA の既約アルゴリズム [3] を適用し，Q', Q'_f, Δ' を更新する．

(5) この時点での Q' が (2) の開始時と変化していないならば，アルゴリズムを終了して (Q', Q'_f, Δ') を出力する．そうでないならば (2) に戻る．

例 20. 例 15 では失敗した十分完全性判定を，状態数削減アルゴリズムを用いて再度行う．例 15 で構成した制約付き TA $\mathcal{A}'$ を入力として上のアルゴリズムを適用した結果の制約付き TA $\mathcal{A}'' = (Q'', Q''_f, \Delta'')$ は次のようになる：$Q'' = \{q_0,\ q_{\mathsf{s}(\square)},\ \widetilde{q}_\blacksquare\}$, $Q''_f = \{\widetilde{q}_\blacksquare\}$,

$$\Delta'' = \left\{ \begin{array}{l} \mathsf{0} \to q_0, \quad \mathsf{s}(q_0) \to q_{\mathsf{s}(\square)}, \quad \mathsf{s}(q_{\mathsf{s}(\square)}) \to q_{\mathsf{s}(\square)}, \\ \mathsf{s}(\widetilde{q}_\blacksquare) \to \widetilde{q}_\blacksquare, \ \ \mathsf{f}(q_0) \xrightarrow{\langle 1 \rangle \leqq 0} \widetilde{q}_\blacksquare, \ \ \mathsf{f}(\widetilde{q}_\blacksquare) \to \widetilde{q}_\blacksquare, \\ \mathsf{f}(q_{\mathsf{s}(\square)}) \xrightarrow{\langle 1 \rangle \leqq 0 \wedge \neg \langle 1 \cdot 1 \rangle \geqq 0} \widetilde{q}_\blacksquare, \\ \mathsf{f}(q_{\mathsf{s}(\square)}) \xrightarrow{\neg \langle 1 \rangle \leqq 0 \wedge \langle 1 \cdot 1 \rangle \geqq 0} \widetilde{q}_\blacksquare \end{array} \right\}$$

よって $Q'' \setminus Q''_f = \{q_0, q_{\mathsf{s}(\square)}\}$ となり，$\mathsf{0}$ と $\mathsf{s}(\square)$ は両方とも $\mathcal{T}(\mathcal{G} \cup \{\square\})$ に含まれることから，制約付き項書換え系 R の十分完全性が証明された．

6. おわりに

本論文では，制約付き項のインスタンスを受理し，制約完全性・完全性・決定性を満たす制約付き TA の構成法を示した．また，制約付き TA の状態数削減アルゴリズムを提案することで，十分完全性の判定に応用できることを例で示した．今回の例では十分完全性の判定のためにオートマトンの受理集合の項のパターンを考えたが，より一般的には，十分完全性や R 完全性の判定は制約付き TA の積集合空問題に帰着できる．ただし，制約付き TA の空判定は決定不能であることからこれらの性質も決定不能となる．本論文では空判定の精度向上のために，なるべく状態数の少ない制約付き TA を構成した．

今後の課題として，本論文の成果を利用した R 完全性の自動判定法を確立し実装すること，また精度向上のための更なる状態数の削減が挙げられる．

文　　献

[1] F. Baader and T. Niplow.: "Term Rewriting and All that", Cambridge University Press, 1998.

[2] M. Huth and M. Ryan.: "Logic in Computer Science: Modelling and Reasoning about Systems", Cambridge University Press, United Kingdom, 2000.

[3] H. Comon, M. Dauchet, R. Gilleron, F. Jacqumard, D. Lugiez, C. Loding, S. Tison and M. Tommasi.: "Tree Automata Techniques and Applications", Available on: http://tata.gforge.inria.fr/, 2008.

[4] N. Nishida, F. Nomura, K. Kurahashi and M. Sakai.: "Constrained Tree Automata and their Closure Properties", In *Proc. of TTATT 2012*, pp. 24–34, 2012.

[5] 坂田，西田，坂部，酒井，草刈：制約付き項書換え系における書換え帰納法，情報処理学会論文誌プログラミング，Vol. 2, No.2, pp. 80–96, 2009.

[6] 古市，西田，酒井，草刈，坂部：制約付き項書換え系の潜在帰納法を利用した手続きプログラミング検証の試み，情報処理学会論文誌プログラミング，Vol. 1, No. 2, pp. 100–121, 2008.