目次

1 論理と有限オートマトン

坂口 和彦

1.1 プレスバーガー算術

プレスバーガー算術 (*Presburger arithmetic*) とは、自然数とその加法に関する一階述語論理の理論である。ここでは、定数記号として 0, 1、項数 (arity) が 2 の関数記号として $+$、項数が 2 の述語記号として $\leq$ を持つものとしてプレスバーガー算術を定義する [*1]。

変数　$x, y, z, \ldots$

項　$t_1, t_2, \ldots ::= x \mid 0 \mid 1 \mid t_1 + t_2$

命題　$\varphi, \psi, \ldots ::= t_1 \leq t_2$

$$\mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \Rightarrow \psi \mid \exists x.\, \varphi \mid \forall x.\, \varphi$$

自然数に関する問題で良く出現するいくつかの式や命題とプレスバーガー算術の項や命題の対応関係を表 1.1 に示す。このように自然数に関するいくつかの重要な概念がプレスバーガー算術上で表現できる一方で、定数倍以外の積を使わないと表現できないものはプレスバーガー算術の上では記述できないことが知られている。例えば、「p は素数である」に相当する論理式はプレスバーガー算術の上で記述できない。これは自然数の積を使う場合、

$$2 \leq p \wedge \forall x.\, (\exists y.\, \boldsymbol{xy} = p) \Rightarrow (x = 1 \vee x = p)$$

のようにして書ける。

プレスバーガー算術の文の真偽は**決定可能** (*decidable*) である。即ち、プレスバーガー算術の文を入力として取り、その真偽を判定するアルゴリズムが存在する [*2]。プレスバーガー算術の決定手続きの応用の範囲は、自動証明からプログラムの最適化まで様々である。以下にその例をいくつか挙げる。

[*1] $\leq$ の代わりに等号 $=$ を使って定義する場合もあるが、どちらも論理体系としての表現力は同等である。

[*2] このようなアルゴリズムを**決定手続き** (*decision procedure*) と呼ぶ。

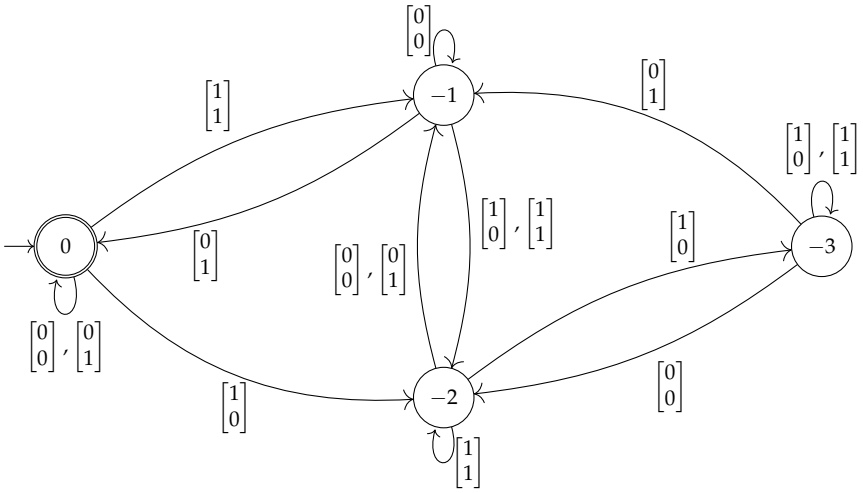

図 1.1: 原始命題 $3x_0 - x_1 \leq 0$ を変換して得られる DFA

補題 1.10 任意の原始命題 $\varphi := a_0x_0 + \cdots + a_{n-1}x_{n-1} \leq b$ と $w \in (2^n)^*$、定義 1.8 を φ に適用して得られる DFA A_φ について、以下が成り立つ。

$$w \in L(A_\varphi) \Leftrightarrow \sum_{i=0}^{n-1} a_i \mathsf{a{\leftarrow}w}(w)_i \leq b$$

Proof. A_φ の定義より、$w \in L(A_\varphi)$ は $0 \leq \hat{\delta}(b, w)$ と同値である。$0 \leq \hat{\delta}(b, w) \Leftrightarrow \sum_{i=0}^{n-1} a_i \mathsf{a{\leftarrow}w}(w)_i \leq b$ を w に関する帰納法で示す。

- $w = \varepsilon$ の場合:

$$\begin{aligned} 0 \leq \hat{\delta}(b, \varepsilon) &\Leftrightarrow 0 \leq b && (\hat{\delta} \text{ の定義}) \\ &\Leftrightarrow \sum_{i=0}^{n-1} a_i \mathbf{0}_i \leq b \\ &\Leftrightarrow \sum_{i=0}^{n-1} a_i \mathsf{a{\leftarrow}w}(\varepsilon)_i \leq b && (\mathsf{a{\leftarrow}w} \text{ の定義}) \end{aligned}$$

- $w = \zeta w'$ の場合:

$$0 \leq \hat{\delta}(b, \zeta w')$$

1.4.3　Coq での定義

原始命題に対応する DFA を、Coq 上で構成する。以下の変数 cs と n は、原始命題 $\varphi := a_0x_0 + \cdots + a_{n-1}x_{n-1} \leq b$ の a_i と b に対応する。

```
Section dfa_of_atomic_formula.
Variable (fvs : nat) (cs : int ^ fvs) (n : int).
```

まず、q_l と q_u 相当の state_lb, state_ub を定義する。

```
Definition state_lb : int := Num.min n (- \sum_(i : 'I_fvs | 0 <= cs i) cs i)%R.
Definition state_ub : int := Num.max n (- \sum_(i : 'I_fvs | cs i <= 0) cs i)%R.
```

補題 1.12 を証明する。

```
Lemma afdfa_s_proof : (state_lb <= n <= state_ub)%R.
Proof. by rewrite /state_lb /state_ub ler_minl ler_maxr lerr. Qed.
```

次に、補題 1.13 を証明する。補題 lez_divL は、補題 afdfa_trans_proof を示すための道具である。

```
Lemma lez_divL d m n : (0 < d -> m <= n * d -> m %/ d <= n)%Z%R.
Proof.
  by move => H H0;
    rewrite -(ler_pmul2r H) (ler_trans _ H0) // -[X in (X <= _)%R]addr0
            {2}(divz_eq m d) ler_add2l; apply modz_ge0, lt0r_neq0.
Qed.
```

```
Lemma afdfa_trans_proof (q : range state_lb state_ub) (ch : bool ^ fvs) :
  (state_lb <=
   ((int_of_range q - \sum_(i : 'I_fvs | ch i) cs i) %/ 2)%Z <=
   state_ub)%R.
Proof.
  case: q => /= q /andP []; rewrite /state_lb /state_ub // => H H0.
  apply/andP; split;
    [case: minrP H {H0} => H H0; rewrite lez_divRL // |
     case: maxrP H0 {H} => H H0; rewrite lez_divL //];
  rewrite mulz2 ler_add //; [apply (ler_trans H) | | apply: (ler_trans _ H) |];
  rewrite ler_opp2 big_mkcond [X in (_ <= X)%R]big_mkcond /=;
  apply (big_ind2 (R1 := int) Num.le (lerr 0) (@ler_add _)) => i _;
  case: (ch i); case: ifP => // /negbT; rewrite -ltrNge ltr_def => /andP [].
Qed.
```

以上の 2 つの補題を使うことで、目的の DFA を定義できる。

```
Definition dfa_of_af : dfa [finType of bool ^ fvs] :=
  {| dfa_state      := [finType of range state_lb state_ub];
     dfa_s          := Range afdfa_s_proof;
```

題を w に関する帰納法で示す。

$$\mathrm{a}{\leftarrow}\mathrm{w}(w) = (x_0, \mathbf{0})) \Rightarrow q\ R_0^*\ \hat{\delta}(q, w)$$

- $w = \varepsilon$ の場合:
  $\hat{\delta}(q, \varepsilon) = q$ より明らか。
- $w = \zeta w'$ の場合:

$$\begin{aligned}
&\mathrm{a}{\leftarrow}\mathrm{w}(\zeta w') = (x_0, \mathbf{0}) \\
&\Rightarrow \zeta + 2\mathrm{a}{\leftarrow}\mathrm{w}(w') = (x_0, \mathbf{0}) && (\mathrm{a}{\leftarrow}\mathrm{w}\ \text{の定義}) \\
&\Rightarrow \zeta = (x_0 \bmod 2, \mathbf{0}) \wedge \mathrm{a}{\leftarrow}\mathrm{w}(w') = \left(\frac{x_0}{2}, \mathbf{0}\right)
\end{aligned}$$

$\zeta = (x_0 \bmod 2, \mathbf{0})$ と $\mathrm{a}{\leftarrow}\mathrm{w}(w') = \left(\frac{x_0}{2}, \mathbf{0}\right)$ から、それぞれ以下が導ける。

$$\begin{aligned}
&\zeta = (x_0 \bmod 2, \mathbf{0}) \\
&\Rightarrow \delta(q, \zeta) \in \delta'(q, \mathbf{0}) && (\delta'\ \text{の定義}) \\
&\Rightarrow q\ R_0\ \delta(q, \zeta) && (R_0\ \text{の定義})
\end{aligned}$$

$$\begin{aligned}
&\mathrm{a}{\leftarrow}\mathrm{w}(w') = \left(\frac{x_0}{2}, \mathbf{0}\right) \\
&\Rightarrow \delta(q, \zeta)\ R_0^*\ \hat{\delta}(\delta(q, \zeta), w') && (\mathrm{IH}) \\
&\Rightarrow \delta(q, \zeta)\ R_0^*\ \hat{\delta}(q, \zeta w') && (\hat{\delta}\ \text{の定義})
\end{aligned}$$

$q\ R_0\ \delta(q, \zeta)$ と $\delta(q, \zeta)\ R_0^*\ \hat{\delta}(q, \zeta w')$ より、$q\ R_0^*\ \hat{\delta}(q, \zeta w')$ が成り立つ。
□

NFA $A_{\exists x_0.\,\varphi}$ が $\exists x_0.\,\varphi$ に対応することを示すのが、以下の補題である。

補題 1.18 命題 $\varphi(x_0, \ldots, x_n)$ と 2^{n+1} の語を入力として取る DFA A_φ について、$L(\varphi) = L(A_\varphi)$ と仮定する。このとき、$L(\exists x_0.\,\varphi) = L(A_{\exists x_0.\,\varphi})$ 即ち以下が成り立つ。

$$\forall w \in (2^n)^*.\, w \in L(A_{\exists x_0.\,\varphi}) \Leftrightarrow (\exists x_0 \in \mathbb{N}.\, \varphi(x_0, \mathrm{a}{\leftarrow}\mathrm{w}(w)))$$

Proof. NFA の言語の定義より、左辺の $w \in L(A_{\exists x_0.\,\varphi})$ は $\hat{\delta}'(q_0, w) \cap F' \neq \varnothing$ と同値である。