AIST-PS-2005-014

一次元セルオートマトンの有限近似解析

(Preliminary Version)

高橋孝一* 田辺良則** 関澤俊弦**

* 産業技術総合研究所 システム検証研究センター

** 科学技術振興機構 CREST / 産業技術総合研究所

一次元セルオートマトンの有限近似解析 (Preliminary Version)

Finite Approximation Analysis of One Dimensional Cellular Automata

高橋 孝一 † 田辺 良則 ‡ 関澤 俊弦 ‡

Koichi TAKAHASHI Yoshinori TANABE Toshifusa SEKIZAWA

† 産業技術総合研究所 システム検証研究センター

‡ 科学技術振興機構 CREST / 産業技術総合研究所 システム検証研究センター

概 要

一次元セルオートマトンの有限近似解析法を提案する．一次元に並んだセル全体を両方向無限パスとみなすことによって，セルの性質を 2 方向 LTL を使って表現することができる．有限個の 2 方向 LTL の式を選び，その真偽値の列を立方体と呼ぶ．立方体の集合によって状態を有限近似する．有限近似から次世代の状態の有限近似を計算する方法を与える．これによって一次元セルオートマトンの有限近似解析を行うことができる．

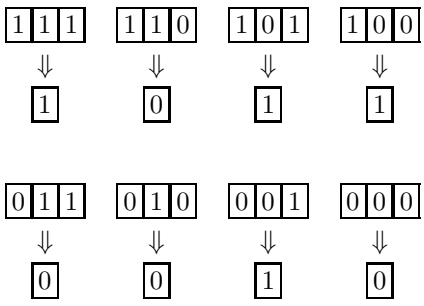
1 はじめに

一次元セルオートマトン [1, 2] を解析する方法を提案する．基本的なアイディアは著者らの時相論理による抽象化を使った解析方法 [3] である．時相論理を 2 方向 LTL とし，一次元セルオートマトンに限定することによって，解析手法を見通しよく整理し，簡略化に成功した．以前の方法では，到達可能性式 (Reachability formulas) というものを用意する必要があったが本稿の方法では必要なくなっている．

一次元セルオートマトンは，次のようなものである．

- セルオートマトンはセルの両方向無限一次元配列から成る．
- 各セルは 0 または 1 という状態を持つ．
- 各セルは単位時間毎に状態を変化させる．
- 全てのセルは一斉に状態を変化させる．これを一世代進むと呼ぶ．
- 変化のルールは現在の自分および周辺のセルの状態によって定まる (過去の状態や将来の状態には依存しない)
- 変化のルールは全てのセルにおいて同じである．

例 1 自分と両隣の状態のみを観察して次の状態を決める次のルールを考える．これはルール $178(= 10110010_2)$ と呼ばれる．この番号は自分と両隣の状態を 2 進数と見て大きい順にルールを並べ，変化後の値を並べて得られる．



セルオートマトンは次のように単位時間ごとに変化していく．

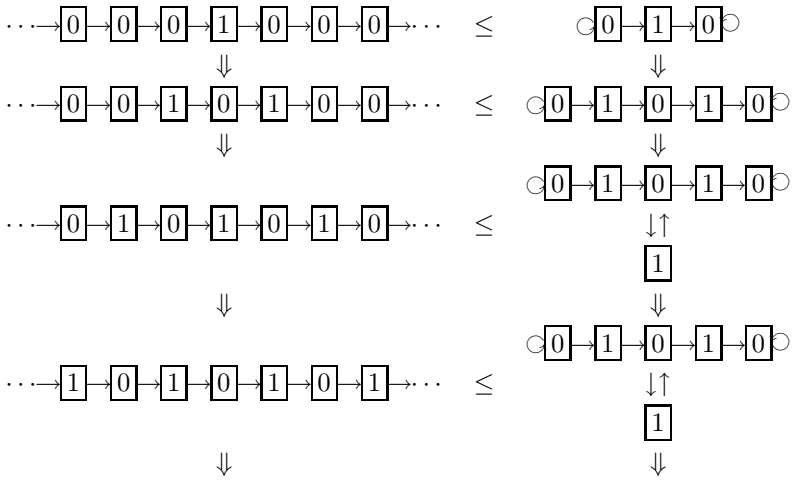
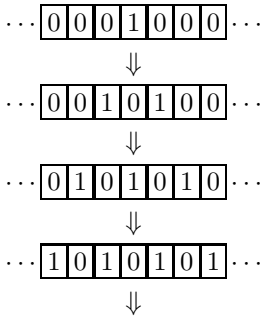


図 1: ルール 178 のセルオートマトンの有限 Kripke 構造の近似列



原始論理式の集合として $\{1\}$ をもつ Kripke 構造を考える．セルオートマトンは右隣のセルへ遷移関係がある一直線の Kripke 構造と自然にみなせる．

Kripke 構造 $K_1 = (N_1, E_1, \lambda_1)$ から $K_2 = (N_2, E_2, \lambda_2)$ へ次のような準同型写像が存在するとき，$K_1 \leq K_2$ が成り立つと書くこととする．写像 $h : N_1 \to N_2$ が準同型写像であるとは，$(n, n') \in E_1$ なら $(h(n), h(n')) \in E_2$ であり，$\lambda_1(n) = \lambda_2(h(n))$ が成り立つことである．

この関係は推移律を満たす．$K_1 \leq K_2$ の時，K_2 は K_1 の「近似 (approximation)」になっているという．一次元セルオートマトンを有限の大きさの Kripke 構造によって解析することが目標である．

例 2 例 1 は図 1 のような有限 Kirpke 構造の近似列が存在する．右の列では，同じ Kripke 構造が繰り返され続けるようになり，将来決して 1 1 と並ぶことがないことがわかる．

セルオートマトンの近似になっている Kripke 構造が与えられたとき，次世代でも近似になっているような Kripke 構造を計算することができれば，有限解析が可能になる．

次節において，近似 Kripke 構造から次世代の近似 Kripke 構造の計算アルゴリズムを提案し，第 3 節でアルゴリズムに従った実装とそれを使った解析例を紹介する．

2 2LTL 式を使った抽象化

一次元セルオートマトンの近似列を計算するのに 2LTL 式を使った抽象化を用いる．

2.1 2LTL

2LTL 式とは次の BNF によって表される式である．

$$
\begin{aligned}
f \quad ::= \quad & 0 \mid 1 \mid \neg f \mid f \wedge f \mid f \vee f \\
& \mid \mathrm{X} f \mid \overline{\mathrm{X}} f \mid \mathrm{F} f \mid \overline{\mathrm{F}} f \mid \mathrm{G} f \mid \overline{\mathrm{G}} f \\
& \mid f \mathrm{U} f \mid f \overline{\mathrm{U}} f \mid f \mathrm{R} f \mid f \overline{\mathrm{R}} f
\end{aligned}
$$

$$
\begin{array}{lll}
\pi \vDash 0 & \text{iff} & \pi(0) = 0 \\
\pi \vDash 1 & \text{iff} & \pi(0) = 1 \\
\pi \vDash \neg f & \text{iff} & \pi \nvDash f \\
\pi \vDash f_1 \wedge f_2 & \text{iff} & \pi \vDash f_1 \text{ かつ } \pi \vDash f_2 \\
\pi \vDash f_1 \vee f_2 & \text{iff} & \pi \vDash f_1 \text{ または } \pi \vDash f_2 \\
\pi \vDash \mathrm{X} f & \text{iff} & (\pi + 1) \vDash f \\
\pi \vDash \overline{\mathrm{X}} f & \text{iff} & (\pi - 1) \vDash f \\
\pi \vDash \mathrm{F} f & \text{iff} & \exists k \geq 0.\ (\pi + k) \vDash f \\
\pi \vDash \overline{\mathrm{F}} f & \text{iff} & \exists k \geq 0.\ (\pi - k) \vDash f \\
\pi \vDash \mathrm{G} f & \text{iff} & \forall k \geq 0.\ (\pi + k) \vDash f \\
\pi \vDash \overline{\mathrm{G}} f & \text{iff} & \forall k \geq 0.\ (\pi - k) \vDash f \\
\pi \vDash f_1 \mathrm{U} f_2 & \text{iff} & \exists k \geq 0.\ k > \forall j \geq 0. \\
 & & (\pi + k) \vDash f_2 \text{ かつ } (\pi + j) \vDash f_1 \\
\pi \vDash f_1 \overline{\mathrm{U}} f_2 & \text{iff} & \exists k \geq 0.\ k > \forall j \geq 0. \\
 & & (\pi - k) \vDash f_2 \text{ かつ } (\pi - j) \vDash f_1 \\
\pi \vDash f_1 \mathrm{R} f_2 & \text{iff} & \forall k \geq 0.\ k > \exists j \geq 0. \\
 & & (\pi + k) \vDash f_2 \text{ または } (\pi + j) \vDash f_1 \\
\pi \vDash f_1 \overline{\mathrm{R}} f_2 & \text{iff} & \forall k \geq 0.\ k > \exists j \geq 0. \\
 & & (\pi - k) \vDash f_2 \text{ または } (\pi - j) \vDash f_1
\end{array}
$$

図 2: 2LTL **式の意味**

オーバーラインの付いた記号は遷移を逆方向に考えたものである．両方向に無限に伸びる $0, 1$ **の状態列** $\pi : Z \to \{0, 1\}$ **について，意味が図** 2 **のように与えられる．無限列** $(\pi + k)$ **は** $(\pi + k)(i) = \pi(i + k)$ **として定義する．**

2.2 抽象セル

2LTL **式** f **とセルオートマトンを考える．一つのセルに注目すると，そこを中心に両方向に無限に伸びる** $0, 1$ **の状態列** π **とみなせる．そこで，** $\pi \vDash f$ **の時，そのセルにおいて** f **が成り立つと呼ぶ．**2LTL **式の長さ** n **の有限列** $\langle f_1, \ldots, f_n \rangle$ **を考える．各々の** 2LTL **式について，そのセルにおいて成り立つ場合に** T **を，そうでない場合には** F **を割り当てることによって，長さ** n **の** $\{\mathrm{T}, \mathrm{F}\}$ **の有限列をセルから計算することができる．これをそのセルの「抽象セル」と呼ぶ．抽象セルは** 2^n **通りしかありえない．**

例 3 2LTL **式の列として** $\langle \overline{\mathrm{X}}1, 1, \mathrm{X}1, \mathrm{F}1 \rangle$ **を考える．例** 1 **の初期状態の中央の五つセルの抽象セルは次のようになる．**

$\boxed{0}$	$\boxed{0}$	$\boxed{1}$	$\boxed{0}$	$\boxed{0}$
$\langle \mathrm{FFFT} \rangle$	$\langle \mathrm{FFTT} \rangle$	$\langle \mathrm{FTFT} \rangle$	$\langle \mathrm{TFFF} \rangle$	$\langle \mathrm{FFFF} \rangle$

これらより左側のセルの抽象セルは全て $\langle \mathrm{FFFT} \rangle$ **であり，右側のセルの抽象セルは全て** $\langle \mathrm{FFFF} \rangle$ **である．**

あるセルにおいて 2LTL **式が成り立たないということは，その否定の式が成り立つということである．つまり，各セルでは** $\{f_1, \neg f_1\}, \ldots, \{f_n, \neg f_n\}$ **から一つずつ選んだ** 2LTL **式の連言が成り立つことになる．それを「立方体** (cube)**」と呼ぶ．つまり抽象セルは立方体である．**

例 4 例 3 **で** $\langle \overline{\mathrm{X}}1, 1, \mathrm{X}1, \mathrm{F}1 \rangle$ **の各々の否定は** $\langle \overline{\mathrm{X}}0, 0, \mathrm{X}0, \mathrm{G}0 \rangle$ **となるので，先ほどの抽象セルはそれぞれ次の立方体に対応する．**

$$
\begin{array}{ll}
\langle \mathrm{FFFT} \rangle & \overline{\mathrm{X}}0 \wedge 0 \wedge \mathrm{X}0 \wedge \mathrm{F}1 \\
\langle \mathrm{FFTT} \rangle & \overline{\mathrm{X}}0 \wedge 0 \wedge \mathrm{X}1 \wedge \mathrm{F}1 \\
\langle \mathrm{FTFT} \rangle & \overline{\mathrm{X}}0 \wedge 1 \wedge \mathrm{X}0 \wedge \mathrm{F}1 \\
\langle \mathrm{TFFF} \rangle & \overline{\mathrm{X}}1 \wedge 0 \wedge \mathrm{X}0 \wedge \mathrm{G}0 \\
\langle \mathrm{FFFF} \rangle & \overline{\mathrm{X}}0 \wedge 0 \wedge \mathrm{X}0 \wedge \mathrm{G}0
\end{array}
$$

2.3 被覆集合

立方体の集合 A **がセルオートマトンの全てのセルの抽象セルを含んでいるとき，「被覆」しているという．セルオートマトンから直接被覆集合を計算することはセルの数が無限なので不可能である．そこで，ある有限近似** Kripke **構造が与えられているとする．** Kripke **構造から被覆集合を計算する方法を，存在判定法と呼ぶ．**

存在判定法 5 Kripke **構造** M **が与えられた時，** $\{a \in$ **立方体の全集合** $\mid \exists v \in M.\ M, v \vDash \mathrm{E}(a)\}$ **を返す．** $\mathrm{E}(a)$ **は** 2CTL*式なので，2CTL*のモデル検査が必要である．

定理 6 Kripke **構造がセルオートマトンの近似になっているならば，存在判定法** 5 **によって得られた立方体の集合はセルオートマトンを被覆する．当然，存在判定法** 5 **より弱い** (**より大きな立方体の集合を返す**) **判定法によって得られた立方体の集合も被覆する．**

一般に $M, v \vDash \mathrm{E}(f_1 \wedge f_2)$ **なら，** $M, v \vDash \mathrm{E}(f_1) \wedge \mathrm{E}(f_2)$ **である．従って以下の判定法は判定法** 5 **より弱い．**

存在判定法 7 与えられた立方体 a **を構成する各** 2LTL **式に過去限量子** $(\overline{\mathrm{X}}, \overline{\mathrm{F}}, \overline{\mathrm{G}}, \overline{\mathrm{U}}, \overline{\mathrm{R}})$ **と未来限量子** $(\mathrm{X}, \mathrm{F}, \mathrm{G}, \mathrm{U}, \mathrm{R})$ **が混在していない場合，立方体から過去限量子が現れている式を削除した立方体** (LTL **式**) **を順方向立方体と呼び，** $\overrightarrow{a}$ **と書く．逆に未来限量子が現れている式を削除し，更にオーバーラインのみを消去した立方体** (LTL **式**) **を逆方向立方体と呼び** $\overleftarrow{a}$ **と書く．例えば，** $a = \overline{\mathrm{X}}0 \wedge 0 \wedge \mathrm{X}1$ **なら，** $\overrightarrow{a} = 0 \wedge \mathrm{X}1, \overleftarrow{a} = \mathrm{X}0 \wedge 0$ **となる．このような場合，ある** v **が存在し，** $M, v \vDash \mathrm{E}(\overrightarrow{a})$ **かつ** $\overline{M}, v \vDash \mathrm{E}(\overleftarrow{a})$ ($\overline{M}$ **は遷移の向きを逆にした** Kripke **構造**) **の時，その立方体が存在すると判定する．この形の式については，**LTL **モデル検査器によって検査可能なので，**LTL **モデル検査器もしくは** CTL*-**モデル検査器が存在すれば実装可能である．**

存在判定法 8 判定法 5 **は** $2^n|M|$ **回の** 2CTL*-**モデル検査が必要である．そこで，各ノードにおいて** $\{\mathrm{E}(f_1), \mathrm{E}(\neg f_1), \ldots, \mathrm{E}(f_n), \mathrm{E}(\neg f_n)\}$ **のそれぞれが成り立つかどうかを調べ，成り立つもののみからなる立方体を返す方法を考える．判定法** 5 **より弱い．**$2n|M|$ **回の** 2CTL*-**モデル検査でよい．過去限量子と未来限量子が混在していない場合には** LTL **モデル検査もしくは** CTL*-**モデル検査になる．**

例 9 図 1 **の右上の** Kripke **構造の各ノードで成り立つ式は左から順に** $\{\mathrm{E}(\overline{\mathrm{X}}0 \wedge 0 \wedge \mathrm{X}0 \wedge \mathrm{F}1),\ \mathrm{E}(\overline{\mathrm{X}}0 \wedge 0 \wedge \mathrm{X}1 \wedge \mathrm{F}1),\ \mathrm{E}(\overline{\mathrm{X}}0 \wedge 0 \wedge \mathrm{X}0 \wedge \mathrm{G}0)\}$**,** $\{\mathrm{E}(\overline{\mathrm{X}}0 \wedge 1 \wedge \mathrm{X}0 \wedge \mathrm{F}1)\}$**,** $\{\mathrm{E}(\overline{\mathrm{X}}1 \wedge 0 \wedge \mathrm{X}0 \wedge \mathrm{G}0),\ \mathrm{E}(\overline{\mathrm{X}}0 \wedge 0 \wedge \mathrm{X}0 \wedge \mathrm{G}0)\}$ **となる．従って，判定法** 5 **で得られる集合は** $\{\langle \mathrm{FFFT} \rangle, \langle \mathrm{FFTT} \rangle, \langle \mathrm{FTFT} \rangle, \langle \mathrm{TFFF} \rangle, \langle \mathrm{FFFF} \rangle\}$ **となる．また，各ノードでは次の式が成り立つ．**

○0	→	1	→	0○
$\mathrm{E}(\overline{\mathrm{X}}0)$		$\mathrm{E}(\overline{\mathrm{X}}0)$		$\mathrm{E}(\overline{\mathrm{X}}1), \mathrm{E}(\overline{\mathrm{X}}0)$
$\mathrm{E}(0)$		$\mathrm{E}(1)$		$\mathrm{E}(0)$
$\mathrm{E}(\mathrm{X}1), \mathrm{E}(\mathrm{X}0)$		$\mathrm{E}(\mathrm{X}0)$		$\mathrm{E}(\mathrm{X}0)$
$\mathrm{E}(\mathrm{F}1), \mathrm{E}(\mathrm{G}0)$		$\mathrm{E}(\mathrm{F}1)$		$\mathrm{E}(\mathrm{G}0)$

従って，判定法 8 **で得られる集合には** $\langle \mathrm{FFTF} \rangle$ **が余分に含まれることになる．**

存在判定法 10 2LTL **式** a **について，** $M, v \vDash \mathrm{E}(a)$ **なら，** a **の全ての時間限量子にパス限量子** E **を付加した** 2CTL **式** a' **についても** $M, v \vDash a'$ **となる．そこで，判定法** 5,7,8 **に現れる** $\mathrm{E}(a)$ **の代わりに** a' **を用いた判定法は，より弱い．これらは** 2CTL **モデル検査器** (**もしくは** CTL **モデル検査器**) **があればよい．**

存在判定法 11 いずれの判定法も，手続きに制限時間を設け，判定できなかった立方体は存在していると返す判定法は元の判定法より弱い．

存在判定法 12 全ての立方体の集合を返す方法は最弱の判定法である．

2.4 Kripke 構造化

二つの抽象セル，つまり二つの立方体 a_1, a_2 **を与え，これらが「隣接」しているかどうかを判定する方法を隣接判定法と呼ぶ．隣接している場合に** a_1 **から** a_2 **へ遷移が存在することにすると，立方体の集合は自然に** Kripke **構造になる．**(**ただし，考えている** 2LTL **式の有限列には** 1 **か** 0 **が含まれているとする．**)

隣接判定法 13 $a_1 \wedge \mathrm{X}(a_2)$ **が充足可能になる** (**つまり，ある** π **が存在して** $\pi \vDash a_1 \wedge \mathrm{X}(a_2)$ **となる**) **時のみ隣接していると判定する．**2LTL **の充足判定が必要である．**

定理 14 セルオートマトンを立方体の集合 A **が被覆しているならば，** A **を隣接判定法** 13 **によって** Kripke **構造化した時，それはセルオートマトンを近似する．**

隣接判定法 13 より弱い (より多くの立方体のペアを隣接すると判定する) 判定法によって得られた Kripke 構造も近似する．

証明．A はセルオートマトン C を被覆しているので，セルを対応する抽象セルに写す自然な写像 $h : C \to A$ が存在する．これが Kripke 構造間の準同型になっていることを示す．まず二つのセル c_1, c_2 が遷移関係にあるとする．対応する抽象セルを a_1, a_2 とすると，c_1 で立方体 a_1 が成り立ち，c_2 で立方体 a_2 が成り立つ．よって，c_1 において $a_1 \wedge \mathrm{X}(a_2)$ が成り立ち，$a_1 \wedge \mathrm{X}(a_2)$ は充足可能となる．つまり $(h(c_1), h(c_2)) \in E$ である．次にセルの状態が 1 だった場合，対応する抽象セルに 1 が現れる．0 の場合も同様なので $\lambda(c) = \lambda(h(c))$．後半は近似は推移律が成り立つことから導ける．証明終わり．

隣接判定法 15 立方体 a を構成する各 2LTL 式に過去限量子と未来限量子が混在している式がない場合，$\overrightarrow{a_1} \wedge \mathrm{X}(\overrightarrow{a_2})$ が充足可能かつ $\mathrm{X}\overleftarrow{a_1} \wedge \overleftarrow{a_2}$ も充足可能なとき，隣接していると判定する判定法は判定法 13 より弱い．この判定法は LTL 式の充足判定のみでよい．

隣接判定法 16 $a_1 \wedge \mathrm{X}(a_2)$ が 2LTL 式として充足可能なら，$a_1 \wedge \mathrm{X}(a_2)$ の任意の場所に任意のパス限量子を加えた 2CTL 式も充足可能である．よって，それらの全ての連言も 2CTL 式として充足可能である．この連言が充足可能な時に隣接していると判定する．これは判定法 13 より弱い．この判定法は 2CTL 充足判定器があればよい．

隣接判定法 17 与えられた立方体 a を構成する各 2LTL 式に過去限量子と未来限量子が混在している式がない場合，隣接判定法 16 を隣接判定法 15 と同様に二つの充足可能問題に分離した判定法はより弱い．この判定法は CTL 充足判定器があればよい．

例 18 例 9 での五つの抽象セルの集合を隣接判定法 13 によって Kripke 構造化する．わかりやすくするため，2LTL 式 1 のところは T, F の代わりに $1, 0$ を書く．例えば，$\langle \mathrm{F0FT} \rangle$ と $\langle \mathrm{F1FT} \rangle$ が隣接可能かどうかは $\overline{\mathrm{X}}0 \wedge 0 \wedge \mathrm{X}0 \wedge \mathrm{F}1 \wedge \mathrm{X}(\overline{\mathrm{X}}0 \wedge 1 \wedge \mathrm{X}0 \wedge \mathrm{F}1)$ が充足可能かどうかである．$\overline{\mathrm{X}}0 \wedge 0 \wedge \mathrm{X}0 \wedge \mathrm{F}1 \wedge \mathrm{X}(\overline{\mathrm{X}}0 \wedge 1 \wedge \mathrm{X}0 \wedge \mathrm{F}1)$ が充足可能なら $\overline{\mathrm{X}}0 \wedge 0 \wedge \mathrm{X}0 \wedge \mathrm{F}1 \wedge \mathrm{X}\overline{\mathrm{X}}0 \wedge \mathrm{X}1 \wedge \mathrm{X}\mathrm{X}0 \wedge \mathrm{X}\mathrm{F}1$ が充足可能である．しかし，これには X0 と X1 が同時に現れているので充足不能である．よって，元の式も充足不能であるので，遷移関係は存在しない．同様にして抽象セル間の遷移関係を計算すると，次のような Kripke 構造になる．

$$\circlearrowleft\langle \mathrm{F0FT} \rangle \to \langle \mathrm{F0TT} \rangle \to \langle \mathrm{F1FT} \rangle \to \langle \mathrm{T0FF} \rangle \to \langle \mathrm{F0FF} \rangle \circlearrowright$$

存在判定法 7 を使った場合の余分の六つ目の抽象セル $\langle \mathrm{FFTF} \rangle$ は，それ自身が充足不能なので Kripke 構造の中で孤立点になる．

セルオートマトンの全てのセルは $\mathrm{GX}(1 \vee 0) \wedge \overline{\mathrm{GX}}(1 \vee 0)$ が成り立つので，Kripke 構造において $\mathrm{E}(\mathrm{GX}(1 \vee 0) \wedge \overline{\mathrm{GX}}(1 \vee 0))$ が成り立たないノードは削除しても被覆性は失われない．つまり，孤立点などの行き止まりの抽象セルは削除してよい．

隣接判定法 19 いずれの判定法も，制限時間を設けたヴァージョン，つまり，ある時間内に決定できなかったら，隣接していると判定する判定法は，元の判定法より弱い．

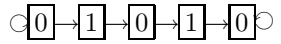
隣接判定法 20 全ての抽象セルのペアは隣接していると判定する判定法は，最弱である．

2.5 次世代の Kripke 構造

考えている 2LTL 式の有限列が一次元セルオートマトンのルールを記述するのに十分，つまり，セルの次世代の状態が対応する抽象セルから一意に決定できる場合，抽象セルの Kripke 構造から次世代の Kripke 構造が計算できる．セルと対応する抽象セルは次世代において必ず同じ値になるので，現世代の準同型写像がそのまま次世代でも準同型写像になる．

定理 21 抽象セルの Kripke 構造がセルオートマトンの近似になっているならば，次世代の Kripke 構造は次世代のセルオートマトンの近似になっている．

例 22 例 1 の場合，$1, \mathrm{X}1, \overline{\mathrm{X}}1$ の三つによってルールが完全に記述できる．従って，例 18 の抽象セルの Kripke 構造の次世代の Kripke 構造は次のようになる．



このように図 1 の初期 Kripke 構造の次世代の Kripke 構造を得ることができる．その次の世代も同様にして得ることができる．

2.6 解析

Kripke 構造を与えると，次の手続きによって，次世代の Kripke 構造を得ることができる．

1. **いずれかの存在判定法を使い，抽象セルの集合を計算し，**
2. **いずれかの隣接判定法を使い，Kripke 構造化し，**
3. **ルールに従って，次世代の Kripke 構造を計算する．**

ここで用いる存在判定法と隣接判定法が定理の条件を満たしている時，初期の Kripke 構造が近似になっていれば，次世代の Kripke 構造も近似になっている．これを繰り返せば，Kripke 構造の世代列が得られる．このようにして得られる Kripke 構造は有限通りしかないので，ある世代はそれ以前のどこかの世代に一致する．これによって，セルオートマトンの有限解析ができる．例えば，近似 Kripke 構造に現れないパターンは具体的なセルオートマトンでも決して現れないことがわかる．

これまでの例では自分および両隣の状態に依存して次の状態が定まるルールだったが，更にもう一つ隣の状態に依存するようなルールの場合，$\mathrm{XX}1, \overline{\mathrm{XX}}1$ を加えておけばルールを完全に記述でき，解析が可能となる．

解析の精度は 2LTL 式の列の選び方，判定法の選び方に依存する．より弱い判定法は解析の精密さを失っていく．2LTL 式は多いほど一般には精度があがるが，時間を要し，結果が複雑になりがちである．

3 実装および解析例

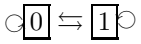
存在判定法 7 および隣接判定法 15 を用いた解析ツールを作成した．判定法に必要となる LTL モデル検査と LTL 式の充足可能性判定には，共に Maude[4] を用いた．自分および両隣を見る 256 個のルールについて解析可能なツールが，著者のホームページ `http://staff.aist.go.jp/k.takahashi/` から利用可能である．スナップショットを図 3 に示す．ツールでは，順方向立方体は forward formula として，逆方向立方体は backward formula としてユーザが与える．$1, \mathrm{X}1, \overline{\mathrm{X}}1$ は標準として 2LTL 式の列に入っている．解析結果は Kripke 構造の世代列として得られる．最後に loop と書かれた世代の次世代は最初に loop と書かれた世代になる．解析結果は pdf ファイルとしてダウンロードできる．矢印が見づらい場合は，ダウンロードして拡大表示するとよい．

これを用いた解析例をいくつか紹介する．

例 23 ルール 178．F1 を追加することによって図 1 の右の列が得られる．最後の Kripke 構造はループとなっており，この初期状態からは $\boxed{1}\boxed{1}$ というパターンは将来決して出てこないことがわかる．

F1 は初期状態の $\boxed{1}$ の左と右を区別するために必要となる．実際 F1 がないと，結果が変わり，ある世代で $\boxed{1}\boxed{1}$ というパターンが現れてしまう．なお，$\boxed{1}\boxed{1}\boxed{1}$ というパターンが決して現れないことは，これでも言える．

例 24 ルール 0．任意の初期状態を被覆する Kripke 構造



から解析した場合，次の世代から

になる．従ってセルオートマトンは，どんな初期状態から始めても次の世代からは決して $\boxed{1}$ が現れないことがわかる．

ルール 8 の場合も同様だが，$\boxed{1}$ が一世代だけ生き残る可能性があることがわかる．

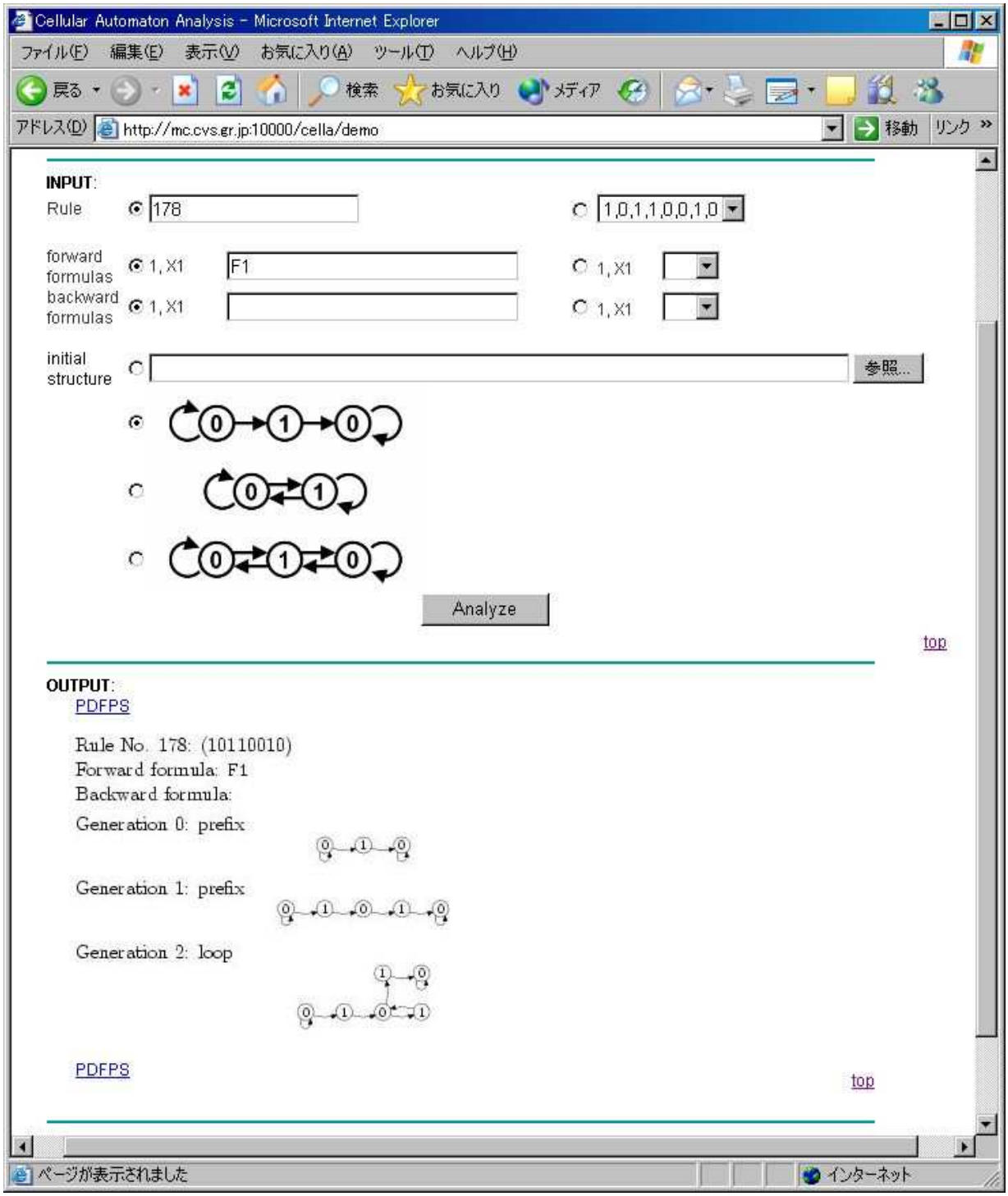
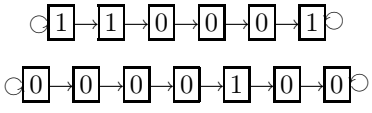


図 3: 解析ツールのスナップショット

例 25 ルール 1．F1, G1, XX1 を追加し，図 1 と同じ初期構造から解析すると，次の二つの状態のループになる．



つまり，0 0 0 以外は全部 1 という状態と，1 が一つだけある状態を振動することがわかる．

式を F1, G1, FG0 として解析すると，Kripke 構造に複数の連結成分が現れる．セルオートマトンは連結なので，その世代の近似はいずれかの連結成分であり，それ以外は偽者である．

例 26 ルール 2．任意の初期状態から始めても，次世代からは 1 の両側には必ず二つ以上 0 が並ぶことがわかる．実際には 1 の位置が世代ごとに左にずれていくのだが，この解析手法では絶対的な位置は抽象化され失われているのでわからない．

解析ツールを使って性質を検証するには，一般にその性質の記述に必要な式を入れておく．例えば 1 が三つ連続するという性質は $\overline{\mathrm{X}}1 \wedge 1 \wedge \mathrm{X}1$ と記述できるので，$\overline{\mathrm{X}}1, 1, \mathrm{X}1$ で十分だが，四つ連続するという性質には XX1 が必要になる．抽象化する時，区別したい性質も入れておく必要がある．

セルオートマトンの性質がわかっている時，例えば 1 は有限個しか現れないことがわかっている場合，FG0 はすべてのセルで成り立つ．こういった式を用いることによって偽者を判定できる場合もある．

4 おわりに

一次元セルオートマトンの有限近似解析の手法を提案した．定理が成り立つ範囲で，存在判定法 5 や隣接判定法 13 より強い判定法があるかどうかはわかっていない．

本稿ではセルオートマトンのセルの取りうる状態数が 2 だったので，1 の否定が即 0 を表すので原始論理式が一つでよかった．しかし，3 以上の場合は原始論理式に工夫が必要となる．

謝辞

本研究の一部は，科学技術振興機構戦略的創造研究推進事業 (CREST) 研究領域「情報社会を支える新しい高性能情報処理技術」研究課題「検証における記述量爆発問題の構造変換による解決」として実施された．

参考文献

[1] S. Wolfram, Cellular Automata and Complexity. Addison-Wesley, 1994

[2] J. von Neumann, Theory of Self-Reproducing Automata. Urbana: University of Illinois Press, 1966

[3] M. Hagiya, K. Takahashi, M. Yamamoto, and T. Sato, Analysis of Synchronous and Asynchronous Cellular Automata using Abstraction by Temporal Logic, *FLOPS2004: The Seventh Functional and Logic Programming Symposium*, LNCS, Vol.2998, 2004, pp. 7–21.

[4] Maude. `http://maude.cs.uiuc.edu/`

一次元セルオートマトンの有限近似解析
(算譜科学研究速報)
発行日：2005 年 8 月 25 日
編集・発行：独立行政法人産業技術総合研究所関西センター尼崎事業所
システム検証研究センター
同連絡先：〒661-0974 兵庫県尼崎市若王寺 3-11-46
e-mail：informatics-inquiry@m.aist.go.jp

Finite Approximation Analysis of One Dimensional Cellular Automata
(Programming Science Technical Report)
May 9, 2002
Research Center for Verification and Semantics (CVS)
AIST Kansai, Amagasaki Site
National Institute of Advanced Industrial Science and Technology (AIST)
3-11-46 Nakouji, Amagasaki, Hyogo, 661-0974, Japan
e-mail: informatics- inquiry@m.aist.go.jp