時相論理の充足可能性判定器のための
ベンチマーク用論理式生成法

(Preliminary Version)

関澤俊弦[1,2]，高井利憲[1,2]，

田辺良則[1,2]，高橋孝一[2]

1：科学技術振興機構，CREST

2：産業技術総合研究所 システム検証研究センター

時相論理の充足可能性判定器のための ベンチマーク用論理式生成法*

(Preliminary Version)

A method to generate benchmark formulae for temporal logic satisfiability checkers

関澤 俊弦 †,‡　高井 利憲 †,‡　田辺 良則 †,‡　高橋 孝一 ‡

†: 科学技術振興機構，CREST　‡: 産業技術総合研究所

{sekizawa-t, t-takai, tanabe.yoshinori, k.takahashi}@aist.go.jp

概 要

時相論理などの充足可能性判定器の性能評価のためには，アルゴリズムの計算量的な解析だけでなく，具体的な論理式を用いたベンチマークが不可欠である．しかし，ベンチマークに用いる論理式セットの明確な基準はなく，また，新たな論理体系に対する充足可能性判定器の性能評価のためには，新たにベンチマーク用の論理式セットを構成しなければならない．本論文では，2 方向 CTL を例にとり，系統的なベンチマーク用論理式の自動生成法を提案する．また，その準備として，ベンチマーク用論理式セットに求められる条件について検討する．最後に，提案手法に基づいて生成されたベンチマーク用論理式セットを用いた実験を行い，結果を述べる．

1 はじめに

システム検証の分野では，組み込みプログラムなどに代表される刺激応答型システムに対する自動検証法が求められている．停止しないことが特徴である刺激応答型システムの仕様は，そのモデルとなる遷移系の時系列的な振る舞いを規定したものと考えることができる．そこで使用されるのが時相論理である．特に，刺激応答型システムに対する自動検証法の代表的な手法であるモデル検査技法は，与えられた有限モデルが時相論理式を満たすか否かを効率的に決定するものである．また著者らは，直接モデル検査法を適用するだけでは状態爆発を起こしてしまうようなシステムに対する検証手法である抽象化技法に，時相論理を導入した [10]．時相論理を用いた抽象化技法では，モデル検査だけでなく，時相論理式に対する充足可能性判定が必要になり，充足可能性判定手続きが一回の抽象化の過程で多数回呼び出される．さらに，時相論理式の充足可能性判定は，他にも，並列プログラム合成 [5, 9]，セルオートマトンの解析 [6]，XML データ処理 [14] などさまざまな応用がある．また，これらを実現するツールの作成に向けて，高速な充足可能性判定アルゴリズムが開発されている [12, 13, 14]．

従来の充足可能性判定アルゴリズムの実装に対する評価法は，(a) 計算量を求める，(b) いくつかの論理式に対する充足可能性判定時間の測定，のいずれかが用いられることが多い．しかし，まず (a) は，複雑なアルゴリズムに対する厳密な計算量を求めることは困難であり，求められたとし

*本研究は，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．

1. 証明可能なものと証明不可能な論理式があること．
2. 論理式はさまざま構成をもつこと．
3. 将来の高速な証明器にも対応可能なこと．
4. すべての論理式の結果が既知であること．
5. 単純なトリックでは問題を解くことができないこと．
6. ベンチマークテストにあまり時間がかからないこと．
7. 結果を要約できること．

図 1: 定理証明器の性能評価のための指針 (出典 [1])

ても，計算量だけでは複数のアルゴリズムの比較は難しい．また，(b) の場合は，評価に用いられる論理式の基準が示されることが少なく，厳密な性能評価は難しい．ランダムに論理式を生成する方法は，充足可能性判定器の特性分析には向いていない．命題論理の充足可能性判定器に対するベンチマークは古くから行われているが，時相論理などに対してはあまり見られない．

Balsiger らは，様相論理に対する定理証明器の性能評価を論理式のパターンを用いて行なう手法を提唱している [1]．ここで，論理式のパターンとは，任意の長さの論理式が得られるように自然数パラメータが導入された論理式である．これらのパターンを基に生成された論理式を評価実験に用いる．あらかじめ，パターンごとの特性，例えば，パラメータに対する原子論理式の数や，様相演算子のネストの深さなどを明らかにしておくことにより，定理証明器の特性を評価しようとするものである．しかし，挙げられている論理式のパターンそのものを構成するための手法は述べられておらず，新たな論理体系に対する充足可能性判定器の性能評価のためにはどのように論理式のパターンを作成すればよいか明らかではない．本論文の目的は，定理証明器や充足可能性判定器の性能評価用の論理式のパターンを簡単かつ系統的に作成するための手法を与えることである．

本論文ではまず，ベンチマーク用論理式セットに求められる条件について検討する．Balsiger らは，定理証明器に対する性能評価法が満たすべき条件を述べ，その条件をほぼ満たすものとして，自然数パラメータ付きの論理式を提案している [1]．ここでは，その性能評価用の自然数パラメータ付きの論理式が満たすべき性質について検討する．次に，著者らが提案している，時相論理によるグラフ書き換え系の抽象化手法 [10, 11] で用いられている 2 方向 CTL を例として，系統的な自然数パラメータ付き論理式の自動生成法を提案する．まず，基礎となる定理を示した後，自動的に生成する方法をいくつか挙げる．提案する自然数パラメータ付き論理式は，簡単な恒真式を用意することによって自動的に得られるものなので，簡単な恒真式を得るためのヒントもいくつか挙げる．最後に，提案手法に基づいて生成されたベンチマーク用論理式セットをいくつか列挙し，我々が開発している充足可能性判定器 [12] に適用した実験結果を示す．

2 方針

Balsiger らは，様相論理に対する定理証明器のためのベンチマーク用論理式セットが満たすべき性質として，図 1 の 7 項目を挙げている [1]．これらを実現するために，自然数パラメータが導入された論理式生成手法を提唱している．これは，比較的簡単な論理式に規則を適用し，任意に長いベンチマーク用論理式を用意する手法であり，ここでは生成された論理式を自然数パラメータ付き

論理式またはパラメータ付き論理式と呼ぶことにする．パラメータ付き論理式をいくつか用意しておき，それぞれ，一定時間，例えば，100 秒以内で解くことができる論理式のパラメータの数によって性能評価を行う．自然数パラメータ付き論理式と図 1 の各項目との対応の，Balsiger らの主張は以下のとおりである．

1. 恒真な論理式と恒真でない論理式を生成するパラメータ付き論理式を用意する．
2. あらかじめ，原子論理式の出現数など，パラメータ付き論理式の特性を調べておき，異なる特性のパラメータ付き論理式を用意する．
3. 自然数パラメータの値を増加させることにより，任意に長い論理式が得られる．
4. 恒真性，または，非恒真性を保存するパラメータ付き論理式を用意する．
5. トリックに対しては不明である．
6. 一定時間内に解ける論理式のパラメータを評価基準とするため，長い時間待たされることはない．
7. 性能評価に用いたパラメータ付き論理式に対し，自然数パラメータで結果を要約することができる．

この主張は，定理証明器の評価に対するものであるが，証明可能な論理式の否定は充足不可能であり，証明不可能な論理式の否定は充足可能なので，証明可能を充足不可能に，証明不可能を充足可能に読み替えれば，充足可能性判定器の評価にも同様のことがいえる．

Balsiger らは，様相論理の定理証明器の評価を行うため，用意する自然数パラメータ付き論理式について，パラメータに対し，(1) 原子論理式の種類が一定のものと増えるもの，(2) 様相演算子のネストの深さが一定のものと増えるもの，などの組み合わせを考慮し，いくつか用意している．しかし，パラメータに対する論理式の長さに関しては，任意に長くできること以外には言及していない．2 方向 CTL の充足可能性判定の計算量の下限は EXPTIME なので [4]，例えば論理式の長さが線形に増えるだけであっても，判定時間は指数的に増大すると予想される．したがって，充足可能性判定器を高い精度で性能評価するためには，論理式の長さが線形に増えるものが望ましいと予想される．しかし，図 1 の項目 3 を満たすためには，論理式の長さが指数的に増えるものも必要になる．従って，自然数パラメータ付き論理式の特性として，(3) 論理式の長さが線形に増えるものと指数的に増えるもの，が必要になると考える．本論文では，上記 (1) と (3) の特性に加え，(2') 時相演算子のネストの深さが一定のものと増えるもの，(4) 様相記号の数が一定のものと増えるもの，(5) 逆向きの様相記号を含むものと含まないもの，を考慮したベンチマーク用論理式セットの提案を目指す．

3 準備

3.1 構文

AP を原子論理式の集合，Mod を様相記号の集合，Mod の各要素 a に対し $\overline{a} \in \mathrm{Mod}$ が存在し $\overline{\overline{a}} = a$ とする．2 方向 CTL 論理式は，次のように定義される．

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathtt{E}_A \mathtt{X}\varphi \mid \mathtt{E}_A[\varphi\ \mathtt{U}\ \varphi] \mid \mathtt{E}_A[\varphi\ \mathtt{R}\ \varphi]$$

ここで，p は原子論理式，A は様相記号の空でない有限部分集合である．以下では，2 方向 CTL 論理式を単に論理式と呼ぶ．論理式 φ に出現する原子論理式 $p_1, \ldots, p_n$ を，それぞれ論理式 $\psi_1, \ldots, \psi_n$ で置き換えた論理式を $\varphi[\psi_1/p_1, \ldots, \psi_n/p_n]$ と表す．$\varphi[\psi_1/p_1, \ldots, \psi_n/p_n]$ に対し，同じ置き換えを適用した論理式を $\varphi[\psi_1/p_1, \ldots, \psi_n/p_n]^2$ とかく．同様に，m 回繰り返し適用した論理式を $\varphi[\psi_1/p_1, \ldots, \psi_n/p_n]^m$ とかく．

3.2 意味

AP を原子論理式の集合，Mod を様相記号の集合とする．次の 3 つの条件を満たす $\mathcal{M} = (M, \{R_a \mid a \in \mathrm{Mod}\}, \lambda)$ をクリプキ構造という．1) $R_a \subseteq M \times M$，2) $R_{\overline{a}} = \{(s,t) \mid (t,s) \in R_a\}$，3) $\lambda\colon \mathrm{AP} \to 2^M$．$M$ の要素を状態と呼ぶ．$x \in \mathrm{AP}, X \subseteq M$ とするとき，関数 $\lambda\colon \mathrm{AP} \to 2^M$ に対して，x の値を X に変更した関数を $\lambda\{x \mapsto X\}$ とかく．また，クリプキ構造 $\mathcal{M}$ の λ を，上記のように変更したものを，$\mathcal{M}\{x \mapsto X\}$ とかく．本論文では，クリプキ構造が全域的であることを仮定しない．$A \subseteq \mathrm{Mod}$ に対し，次を満たす σ を A-パスと呼ぶ．(1) $\sigma\colon \alpha \to M, 1 \leq \alpha \leq \omega$．この α を $\mathrm{len}(\sigma)$ と表し，$\sigma(i)$ を σ_i ともかく．(2) $i+1 < \alpha$ なる i に対して $a \in A$ が存在して $(\sigma_i, \sigma_{i+1}) \in R_a$．(3) $\alpha < \omega$ ならば，$(\sigma_{\alpha-1}, s) \in R_a$ なる $s \in M$ と $a \in A$ は存在しない．クリプキ構造 $\mathcal{M} = (M, \{R_a \mid a \in \mathrm{Mod}\}, \lambda)$ とその状態 $s \in M$，論理式 φ に対し，$\mathcal{M}, s \models \varphi$ を次のように，φ の形で場合分けして定義する．

- $p \in \mathrm{AP}$ の場合，$s \in \lambda(p)$ のとき $\mathcal{M}, s \models p$.
- $\neg\varphi'$ の場合，$\mathcal{M}, s \not\models \varphi'$ のとき，$\mathcal{M}, s \models \neg\varphi'$.
- $\varphi_1 \vee \varphi_2$ の場合，$\mathcal{M}, s \models \varphi_1$ または $\mathcal{M}, s \models \varphi_2$ のとき $\mathcal{M}, s \models \varphi_1 \vee \varphi_2$.
- $\mathtt{E}_A\mathtt{X}\varphi'$ の場合，$s' \in M$ と $a \in A$ で，$(s, s') \in R_a$ かつ $\mathcal{M}, s' \models \varphi'$ となるものが存在するとき，$\mathcal{M}, s \models \mathtt{E}_A\mathtt{X}\varphi'$.
- $\mathtt{E}_A[\varphi_1\ \mathtt{U}\ \varphi_2]$ の場合，$\sigma_0 = s$ なる A-パス σ と $i < \mathrm{len}(\sigma)$ が存在し，(1) $\sigma_i \models \varphi_2$ かつ (2) $0 \leq j < i$ ならば $\sigma_j \models \varphi_1$ が成り立つとき，$\mathcal{M}, s \models \mathtt{E}_A[\varphi_1\ \mathtt{U}\ \varphi_2]$.
- $\mathtt{E}_A[\varphi_1\ \mathtt{R}\ \varphi_2]$ の場合，$\sigma_0 = s$ なる A-パス σ が存在し，(1) すべての $i < \mathrm{len}(\sigma)$ に対し，$\sigma_i \models \varphi_2$ か，または (2) $i < \mathrm{len}(\sigma)$ が存在し，$\sigma_i \models \varphi_1$ であり，かつ，全ての $j \leq i$ に対し，$\sigma_j \models \varphi_2$ が成り立つとき，$\mathcal{M}, s \models \mathtt{E}_A[\varphi_1\ \mathtt{R}\ \varphi_2]$.

記号 $\wedge, \rightarrow, \leftrightarrow, \mathtt{AX}, \mathtt{AU}, \mathtt{AR}, \mathtt{AG}, \mathtt{AF}, \mathtt{EG}, \mathtt{EF}, \top, \bot$ などは，通常通り他の論理式の別名として用いる．クリプキ構造 $\mathcal{M} = (M, R, \lambda)$，論理式 φ に対し，φ を満たす状態すべてを $[\![\varphi]\!]_{\mathcal{M}}$ とかく．すなわち，$[\![\varphi]\!]_{\mathcal{M}} = \{s \in M \mid \mathcal{M}, s \models \varphi\}$ である．

なお，以下では，論理式 φ に対して，$\mathtt{EX}^n\varphi$ $(n \geq 0)$ を，φ の前に $\mathtt{EX}$ が n 個つくことを意味する略記として使用する．ただし，$n = 0$ のとき $\mathtt{EX}^0\varphi \leftrightarrow \varphi$ とする．

4 論理式の自動生成

本節では，パラメータ付き論理式の構成法を与えることを目標とする．パラメータ付き論理式 χ の与え方の方針として，(1) 種となる論理式 $\chi(0)$ と (2) 任意のパラメータ n に対する，$\chi(n)$ の構成法を与えることとする．上記 (2) の手続きを以下では複雑化と呼ぶ．$\chi(0)$ が恒真なら $\chi(n)$ も

恒真になる複雑化を，恒真性を保存する複雑化といい，$\chi(0)$ が充足可能なら $\chi(n)$ も充足可能になる複雑化を，充足可能性を保存する複雑化という．充足可能性判定器の評価のためには，充足可能なものと不可能なものを用意しなければならないが，充足不可能なものは，恒真な論理式の否定をとればよいので，ここでは，恒真性を保存する複雑化について議論する．

4.1 恒真性を保存する自動生成

恒真なパラメータ付き論理式を得る単純な方法として，恒真な論理式に現れる原子論理式を複雑な論理式に置き換える手法が考えられる．つまり，恒真な論理式 α_0 に対し，α_0 に含まれる x を論理式 φ で置き換えるとするとすると，パラメータ付き論理式 χ_0 は，

$$\chi_0(n) = \alpha_0[\varphi/x]^n$$

と書けるものである．しかし，このような複雑化では，充足可能性判定器がもとの恒真な論理式の形を検出してしまえば，パラメータを増やしても，一定時間で解くことができてしまう可能性がある．これは，充足可能性判定器の単純なトリックにより，論理式をいくら大きくしても，簡単に解かれてしまう例である．したがって，この複雑化では，充足可能性判定器の正しい評価が行えない可能性がある．以下では，上記のように単純でない複雑化を与えるための系統的な方法を提案する．初めに定理を示し，提案する複雑化が恒真性を保存することを保証する．次に，それらの定理に基づいた具体的な複雑化を与える．

いくつか準備する．論理式が正形式であるとは，否定記号 $\neg$ が原子論理式の直前にのみ現れることをいう． 任意の論理式は，正形式に変換できる．原子論理式 x が論理式 φ で正にしか現れないとは，φ を変換した正形式に含まれるすべての x の前に否定記号がないときをいう．φ を論理式，x を原子論理式，$\mathcal{M} = (M, R, \lambda)$ をクリプキ構造，$X, X' \subseteq M$ を状態の集合とする． x が φ で正に現れて，かつ $X \subseteq X'$ ならば，

$$[\![\varphi]\!]_{\mathcal{M}\{x \mapsto X\}} \subseteq [\![\varphi]\!]_{\mathcal{M}\{x \mapsto X'\}} \tag{1}$$

が成り立つ．φ に関する帰納法で証明できる．

次の定理は，以下で複雑化を考える上で基礎となるものである．この定理から系をいくつか示し，それらの系を基にして実際の複雑化を生成する．

定理 1. x を原子論理式，α_1, β_1 をそれぞれ，x が正にしか現れない論理式，α_0, β_0 を論理式とする． $\alpha_1 \to \beta_1$ と $\alpha_0 \to \beta_0$ が恒真ならば，$\alpha_1[\alpha_0/x] \to \beta_1[\beta_0/x]$ は恒真である．

証明 $\mathcal{M}$ を任意のクリプキ構造とすると，$[\![\alpha_1[\alpha_0/x]]\!]_{\mathcal{M}} \subseteq [\![\beta_1[\beta_0/x]]\!]_{\mathcal{M}}$ が成り立てば十分である．以下のように示される．

$$\begin{aligned}
[\![\alpha_1[\alpha_0/x]]\!]_{\mathcal{M}} &= [\![\alpha_1]\!]_{\mathcal{M}\{x \mapsto X_1\}} && (2)\\
&\subseteq [\![\beta_1]\!]_{\mathcal{M}\{x \mapsto X_1\}} && (3)\\
&\subseteq [\![\beta_1]\!]_{\mathcal{M}\{x \mapsto X_2\}} && (4)\\
&= [\![\beta_1[\beta_0/x]]\!]_{\mathcal{M}}
\end{aligned}$$

ここで，(2) では，$X_1 = [\![\alpha_0]\!]_{\mathcal{M}}$ としている．(3) は，$\alpha_1 \to \beta_1$ が恒真から，および (1) からいえる．(4) は，$X_2 = [\![\beta_0]\!]_{\mathcal{M}}$ とすると，$\alpha_0 \to \beta_0$ が恒真であることからいえる． □

χ_0 のような単純な置き換えではないことに注意してほしい．定理 1 を複数回適用することにより，以下の系 1，2 を得る．

系 1. x を原子論理式，α_i, β_i $(1 \leq i \leq n)$ をそれぞれ，x が正にしか現れない論理式，α_0, β_0 を論理式とする．$\alpha_i \to \beta_i$ $(1 \leq i \leq n)$ および $\alpha_0 \to \beta_0$ が恒真ならば，$\varphi_i \to \psi_i$ $(1 \leq i \leq n)$ も恒真である．ここで，$\varphi_0 = \alpha_0$，$\psi_0 = \beta_0$，$\varphi_i = \alpha_i[\varphi_{i-1}/x]$，$\psi_i = \beta_i[\psi_{i-1}/x]$ $(1 \leq i \leq n)$ である． □

系 2. x_i $(1 \leq i \leq m)$ を原子論理式，α，β をそれぞれ，$x_1, \ldots, x_m$ が正にしか現れない論理式とする．また，α_i, β_i $(0 \leq i \leq m)$ を論理式とする． $\alpha \to \beta$ および $\alpha_i \to \beta_i$ $(0 \leq i \leq m)$ が恒真ならば $\alpha[\alpha_1/x_1, \ldots, \alpha_m/x_m] \to \beta[\beta_1/x_1, \ldots, \beta_m/x_m]$ も恒真である． □

次に，上の系からベンチマーク用の論理式を生成するための具体的な方法を示す．上の系を適用するためには，$\alpha \to \beta$ の形をした恒真な論理式が多数必要である．以下では，そのような論理式の簡単な与え方をいくつか紹介することにより，ベンチマーク用論理式の生成法の指針を与える．また，具体的なベンチマーク用論理式を得るためには，複雑化の種となる恒真な論理式 $\alpha_0 \to \beta_0$ を与える必要があるが，それについても簡単な指針を 4.2 節で述べる．

例 1. 系 1 の論理式 α_i, β_i $(1 \leq i \leq n)$ が同じ形しているような複雑化を考える．例えば，系 1 における α_i, β_i をそれぞれ，$\alpha_i = \alpha, \beta_i = \beta$ $(1 \leq i \leq n)$，$\alpha = \mathtt{EX}x$，$\beta = \mathtt{EX}x$ とする． $\mathtt{EX}x \to \mathtt{EX}x$ は恒真なので，系 1 が適用でき，次のようなパラメータ付き論理式 χ_1 が得られる．

$$\chi_1(n) = \mathtt{EX}^n\alpha_0 \to \mathtt{EX}^n\beta_0$$

このパラメータ付き論理式は，CTL の形式的体系 [4] の推論規則

$$\frac{\gamma \to \delta}{\mathtt{EX}\gamma \to \mathtt{EX}\delta}$$

の適用に対応する．したがってこの推論規則を知っている充足可能性判定器であれば，簡単に解かれてしまう可能性がある． そこで，次のような派生形も考える．

$$\chi_2(n) = \bigvee_{1 \leq i \leq n} \left(\mathtt{EX}^i(\alpha_0) \to \mathtt{EX}^n(\beta_0)\right)$$

この複雑化はちょうど，[1] における k_d4_p の複雑化に対応している．我々の提案手法では，$\alpha \to \beta$ の形が恒真であればよいので，例えば，$\alpha_i = \alpha_a, \beta_i = \beta_a$ (i が偶数のとき)，$\alpha_i = \alpha_b, \beta_i = \beta_b$ (i が奇数のとき)，$\alpha_a = x \wedge \mathtt{E}_a\mathtt{X}x$，$\beta_a = \mathtt{E}_a\mathtt{X}\mathtt{E}_{\overline{a}}\mathtt{X}(x)$，$\alpha_b = x \wedge \mathtt{E}_b\mathtt{X}x$，$\beta_b = \mathtt{E}_b\mathtt{X}\mathtt{E}_{\overline{b}}\mathtt{X}(x)$ と適用できる．$x \wedge \mathtt{E}_a\mathtt{X}x \to \mathtt{E}_a\mathtt{X}\mathtt{E}_{\overline{a}}\mathtt{X}(x)$ および $x \wedge \mathtt{E}_b\mathtt{X}x \to \mathtt{E}_b\mathtt{X}\mathtt{E}_{\overline{b}}\mathtt{X}(x)$ は恒真であり，次のようなパラメータ付き論理式が得られる．

$$\chi_3(n) = \bigvee_{1 \leq i \leq n} (\varphi_i \to \psi_n)$$

ここで，$\varphi_0 = \alpha_0$，$\psi_0 = \beta_0$，$\varphi_i = \varphi_{i-1} \wedge \mathtt{E}_a\mathtt{X}\varphi_{i-1}$ (i が偶数のとき)，$\varphi_i = \varphi_{i-1} \wedge \mathtt{E}_b\mathtt{X}\varphi_{i-1}$ (i が偶数のとき)，$\psi_i = \mathtt{E}_a\mathtt{X}\mathtt{E}_{\overline{a}}\mathtt{X}(\psi_{i-1})$ (i が奇数のとき)，$\psi_i = \mathtt{E}_b\mathtt{X}\mathtt{E}_{\overline{b}}\mathtt{X}(\psi_{i-1})$ (i が偶数のとき) である． □

例 2. 系 1 の論理式 α_i, β_i $(1 \leq i \leq n)$ を，原子論理式だけ違う，形の同じ論理式で与える複雑化を考える．例えば，系 1 における α_i，β_i をそれぞれ，$\alpha_i = \mathtt{E}[x\ \mathtt{U}\ q_i], \beta_i = \mathtt{E}[x\ \mathtt{U}\ q_i]$ $(1 \leq i \leq n)$ とする．ここで，q_i $(1 \leq i \leq n)$ は原子論理式であり，$\alpha_i \to \beta_i$ $(1 \leq i \leq n)$ は恒真である．系 1 が適用でき，次のようなパラメータ付き論理式 χ_4 が得られる．

$$\chi_4(n) = \varphi_n \to \psi_n$$

ここで，$\varphi_0 = \alpha_0$，$\psi_0 = \beta_0$，$\varphi_n = \mathtt{E}[\varphi_{n-1}\ \mathtt{U}\ q_n]$，$\psi_n = \mathtt{E}[\psi_{n-1}\ \mathtt{U}\ q_n]$ である．χ_4 は，原子論理式の数が，パラメータに応じて増えることが特徴である． 前の例の中であげたパラメータ付き論理式は，原子論理式の数は一定である． 充足可能性判定器を評価するためには，原子論理式の数が増えるものと一定のものを両方用意しておくのが望ましい．また，4.2 節でも述べるとおり，$\mathtt{A}[c_0\ \mathtt{U}\ p_0] \wedge \mathtt{AG}(p_0 \rightarrow \mathtt{AG}(\neg c_0 \wedge x)) \rightarrow \mathtt{AFAG}(\neg c_0 \wedge x)$ も恒真式なので，上の α_i, β_i の代わりに，$\alpha_i = \mathtt{A}[c_i\ \mathtt{U}\ p_i] \wedge \mathtt{AG}(p_i \rightarrow \mathtt{AG}(\neg c_i \wedge x))$，$\beta_i = \mathtt{AFAG}(\neg c_i \wedge x)$ $(1 \leq i \leq n)$ としてもよい．これを，χ_{13} とする． □

例 3. 時相演算子の深さが，一定の場合のパラメータ付き論理式を作成したい．系 1 の論理式 α_i と β_i として，$\alpha_i = x \wedge \mathtt{A}[p_i\ \mathtt{U}\ q]$，$\beta_i = x \wedge (q \vee (p_i \wedge \mathtt{AXA}[p_i\ \mathtt{U}\ q]))$ をとる $(1 \leq i \leq n)$．これは，$\mathtt{A}[p\ \mathtt{U}\ q] \rightarrow q \vee (p \wedge \mathtt{AXA}[p\ \mathtt{U}\ q])$ が恒真であることを利用した複雑化である．得られるパラメータ付き論理式は，

$$\chi_5(n) = \bigwedge_{1 \leq i \leq n} \mathtt{A}[p_i\ \mathtt{U}\ q] \rightarrow \bigwedge_{1 \leq i \leq n} (q \vee (p_i \wedge \mathtt{AXA}[p_i\ \mathtt{U}\ q]))$$

である． □

例 4. 系 1 の論理式 α_i, β_i $(1 \leq i \leq n)$ を，様相記号だけ違う，形の同じ論理式で与える複雑化を考える．例えば，系 1 における α_i，β_i をそれぞれ，$\alpha_i = \mathtt{A}_{a_i, a_{i+1}}\mathtt{G}x, \beta_i = \mathtt{A}_{a_i}\mathtt{G}x$ $(1 \leq i \leq n)$ とする．$\alpha_i \rightarrow \beta_i$ は恒真であり，あとは上の例 2 と同様の議論ができる．得られたパラメータ付き論理式を χ_{14} とする．様相記号の数が増えるものである． □

例 5. 系 2 の論理式 α_i，β_i をすべて同じ形をしたものを与える複雑化を考える．例えば，系 2 における α, β として，それぞれ，上の例 2 における φ_n，ψ_n を採用する．α_i，β_i をそれぞれ，$\alpha_i = \alpha', \beta_i = \beta'$ $(1 \leq i \leq n)$，$\alpha' = x \wedge \mathtt{E}_a\mathtt{Xtrue}$，$\beta' = \mathtt{E}_a\mathtt{XE}_{\overline{a}}\mathtt{X}x$ とする．$x \wedge \mathtt{E}_a\mathtt{Xtrue} \rightarrow \mathtt{E}_a\mathtt{XE}_{\overline{a}}\mathtt{X}x$ は恒真なので，系 2 が適用でき，次のようなパラメータ付き論理式 χ_6 が得られる．

$$\chi_6(n) = \varphi'_n \rightarrow \psi'_n$$

ここで，$\varphi'_0 = \alpha_0$，$\psi'_0 = \beta_0$，$\varphi'_n = \mathtt{E}[\varphi'_{n-1}\ \mathtt{U}\ (p \wedge \mathtt{E}_a\mathtt{Xtrue})]$，$\psi'_n = \mathtt{E}[\psi'_{n-1}\ \mathtt{U}\ \mathtt{E}_a\mathtt{XE}_{\overline{a}}\mathtt{X}p]$ である． □

以上，種となる恒真な論理式から，恒真性を保存するパラメータ付き論理式を得るための手法を与えたが，次に，その種を得るための指針をいくつか与える．

4.2 簡単な恒真式を得る方法

例 1 で用いた恒真な論理式は，$\mathtt{EX}x \rightarrow \mathtt{EX}x$ である． これは，このような自明な論理式でも場合によっては本論文で提案する手法に用いることができることを示している．しかし，一般には，$\alpha \rightarrow \beta$ という形をしている恒真式の中で，α と β が違うものが望ましい．

自明な定理 時相論理を用いたシステム検証に関する標準的なテキスト [2, 7] などには，CTL の定理がいくつか記載されている．例えば，$\mathtt{E}[\varphi\ \mathtt{U}\ \psi] \leftrightarrow \psi \vee (\varphi \wedge \mathtt{EXE}[\varphi\ \mathtt{U}\ \psi])$, $\mathtt{A}[p\ \mathtt{U}\ q] \leftrightarrow \neg\mathtt{E}[\neg q\ \mathtt{U}\ (\neg p \vee \neg q) \wedge \neg\mathtt{EG}(\neg q)]$ などである．また，CTL の形式的体系などの公理も参考になる [4, 8]．例えば，$\mathtt{EX}(\varphi \vee \psi) \leftrightarrow \mathtt{EX}\varphi \vee \mathtt{EX}\psi$ や，$\mathtt{AX}\varphi \wedge \mathtt{AX}\psi \rightarrow \mathtt{AX}(\varphi \wedge \psi)$ などである．他には，2 方向 CTL でいえば，$\mathtt{A}_{a,b}\mathtt{G}x \rightarrow \mathtt{A}_a\mathtt{G}x$，$x \wedge \mathtt{E}_a\mathtt{Xtrue} \rightarrow \mathtt{E}_a\mathtt{XE}_{\overline{a}}\mathtt{X}x$ なども，自明な恒真式である．

パターンの組み合わせ Dwyer らは，システム検証の分野でよく現れる，システムの時系列的な性質や仕様のパターンを分析し，どのように時相論理式で表現するかについて考察してる [3]. 例えば，イベント P が起こる前には，イベント S は起こらない，という性質は CTL では $\mathtt{A}[\neg S\ \mathtt{U}\ (P \vee \mathtt{AG}\neg P)]$ と記述できる． これらを用いることにより，恒真式を構成する． 例えば，上記論理式に，$\mathtt{absence}(P,S)$ と名前を付けると，イベント S が存在しなくて，将来必ずイベント P が起こるのであれば，当然であるが，P の前には S は起こらないので，$\mathtt{AF}P \wedge \mathtt{AG}\neg S \rightarrow \mathtt{absence}(P,S)$ は恒真式である．

同じように，論理式のモデルを考えると，恒真な式を考えやすい．例えば，ある時点から，イベント c_1 が常に成り立ち，c_0 が成り立たなくなることは，$\mathtt{AFAG}(\neg c_0 \wedge c_1)$ と表せるが，$\mathtt{A}[c_0\ \mathtt{U}\ p_0] \wedge \mathtt{AG}(p_0 \rightarrow \mathtt{AG}(\neg c_0 \wedge c_1))$ のときも，上の命題は成り立つ．つまり，イベント p_0 まで c_0 が成り立ち，かつ，イベント p_0 が成り立つところからは，c_0 が成り立たなくなり，c_1 が成り立つようになるときである．以上の議論から，恒真式 $\mathtt{A}[c_0\ \mathtt{U}\ p_0] \wedge \mathtt{AG}(p_0 \rightarrow \mathtt{AG}(\neg c_0 \wedge c_1)) \rightarrow \mathtt{AFAG}(\neg c_0 \wedge c_1)$ が得られる．

4.3 充足可能な自然数パラメータ付き論理式

充足可能な論理式を生成する方針として，自然数パラメータ n によりパラメータ化されたクリプキ構造の性質 $P(n)$ を考える．例えば,「原子論理式 p が成り立つ状態に n ステップで到達する」などが挙げられる．性質 $P(n)$ としては，n が大きくなるにしたがって，$P(n)$ を満たすクリプキ構造が複雑になるようなものを採用する．このような $P(n)$ を 2 方向 CTL で表現することにより，充足可能な自然数パラメータ付き論理式を得ることができる．上に挙げた例は，$\chi_7(n) = \mathtt{EX}^n(p)$ と表現できる．以下に，いくつか具体的な例を示す.

例 6. 上で述べた，EX による複雑化，つまり χ_7 の派生型を考える．まず，逆方向の様相記号を含まない論理式 α_0 に対して，$\chi_8(n) = \bigwedge_{0\le i\le n} \mathtt{EX}^i(\alpha_0)$ とする．$\chi_8(n)$ を満たすクリプキ構造は，α_0 と原子論理式 p の差はあるものの，時相演算子に関して χ_7 と同じ構造である．しかし，論理式はより複雑になっている．また，適当な α_0 を選べば，$\chi_9(n) = \bigwedge_{1\le i\le n} \left(\mathtt{EX}^{i-1}(\neg\alpha_0) \wedge \mathtt{EX}^i(\alpha_0)\right)$ でもよい. □

例 7. 2 方向 CTL では有限モデル性が成立しない．例えば，論理式 $z \wedge \mathtt{A}_a\mathtt{XA}_a\mathtt{G}(\neg z) \wedge \mathtt{A}_a\mathtt{G}(\mathtt{E}_a\mathtt{X}p \wedge \mathtt{A}_{\overline{a}}\mathtt{F}z)$ が充足するクリプキ構造の状態集合は，必ず無限になる．無限モデルであることは以下のようにわかる． 有限のクリプキ構造 $\mathcal{M}$ の状態 s で上記論理式が成り立つと仮定する．すると，$s \models \mathtt{A}_a\mathtt{GE}_a\mathtt{X}p$ より，s を始点とする無限の A-パス σ が存在する．状態空間は有限なので，$\sigma_i = \sigma_j\ (i < j)$ となるパスが存在し，σ_i から逆向きの様相記号 $\overline{a}$ をたどって σ_j にいたるループを考えると，$\sigma_i \models \mathtt{E}_{\overline{a}}\mathtt{G}\neg z$ となり，矛盾する．自然数パラメータ付き論理式は，論理式を満たすクリプキ構造がより複雑になるようにとる．まず，数直線，すなわち，左右に無限に続くクリプキ構造を表す論理式は，次のようになる.

$$\begin{aligned}\chi_{10}(0) \quad = \quad & z_0 \wedge \mathtt{A}_{a_0}\mathtt{XA}_{a_0}\mathtt{G}(\neg z_0) \wedge \mathtt{A}_{\overline{a_0}}\mathtt{XA}_{\overline{a_0}}\mathtt{G}(\neg z_0) \wedge \\ & \mathtt{A}_{a_0}\mathtt{G}(\mathtt{E}_{a_0}\mathtt{X}p_0 \wedge \mathtt{A}_{\overline{a_0}}\mathtt{F}z_0) \wedge \\ & \mathtt{A}_{\overline{a_0}}\mathtt{G}(\mathtt{E}_{\overline{a_0}}\mathtt{X}n_0 \wedge \mathtt{A}_{a_0}\mathtt{F}z_0)\end{aligned}$$

$n \geq 1$ のときは，$\chi_{10}(n) = \alpha_n \wedge \chi_{10}(n-1) \wedge \beta_n[\chi_{10}(n-1)/x] \wedge \gamma_n[\chi_{10}(n-1)/x]$ となる．ここで，

$$\begin{aligned}
\alpha_i &= z_i \wedge \mathtt{A}_{a_i}\mathtt{X}\mathtt{A}_{a_i}\mathtt{G}(\neg z_j) \wedge \mathtt{A}_{\overline{a_i}}\mathtt{X}\mathtt{A}_{\overline{a_i}}\mathtt{G}(\neg z_j) \\
\beta_i &= \mathtt{E}_{a_i}\mathtt{X}(p_i) \wedge \mathtt{A}_{a_i}\mathtt{G}(\mathtt{E}_{a_i}\mathtt{X}(p_i) \wedge \mathtt{A}_{\overline{a_i}}\mathtt{F}(z_i) \wedge x) \\
\gamma_i &= \mathtt{E}_{\overline{a_i}}\mathtt{X}(p_i) \wedge \mathtt{A}_{\overline{a_i}}\mathtt{G}(\mathtt{E}_{\overline{a_i}}\mathtt{X}(p_i) \wedge \mathtt{A}_{a_i}\mathtt{F}(z_i) \wedge x)
\end{aligned}$$

である．論理式の長さは指数的に増える． □

例 8. 様相記号を増やすことにより，満たすべきクリプキ構造を複雑にすることを目指す．次の論理式 $\chi_{11}(0) = \mathtt{A}_{a_1}\mathtt{G}(\neg\alpha_0) \wedge \mathtt{A}_{a_2}\mathtt{G}(\neg\alpha_0) \wedge \mathtt{E}_{a_1,a_2}\mathtt{F}(\alpha_0)$ は，a_1 または a_2 のみでは α_0 に到達できないが，a_1 と a_2 両方を使えば到達できることを表す．パラメータ化は次のようにする．

$$\begin{aligned}
\chi_{11}(n) &= \mathtt{A}_{a_1,\ldots,a_{n-1}}\mathtt{G}(\neg\alpha_0) \wedge \\
&\quad \mathtt{A}_{a_1,\ldots,a_{n-2},a_n}\mathtt{G}(\neg\alpha_0) \wedge \cdots \wedge \\
&\quad \mathtt{A}_{a_2,\ldots,a_n}\mathtt{G}(\neg\alpha_0) \wedge \mathtt{E}_{a_1,\ldots,a_n}\mathtt{F}(\alpha_0)
\end{aligned}$$

この複雑化だけでは単純すぎるので，様相記号が増えるパラメータ付き論理式がほしいときに，他の複雑化と組み合わせて使う． □

例 9. 時相演算子の深さを一定にしたい．次の論理式

$$\begin{aligned}
\chi_{12}(0) &= \mathtt{A}_a\mathtt{F}\alpha_0 \wedge \mathtt{A}_a\mathtt{F}\alpha_1 \wedge \neg\alpha_0 \wedge \neg\alpha_1 \wedge \\
&\quad \mathtt{A}_a\mathtt{G}(\alpha_0 \rightarrow \neg(\alpha_1 \vee \mathtt{E}_a\mathtt{X}\alpha_1 \vee \mathtt{E}_{\overline{a}}\mathtt{X}\alpha_1)) \wedge \\
&\quad \mathtt{A}_a\mathtt{G}(\alpha_1 \rightarrow \neg(\alpha_0 \vee \mathtt{E}_a\mathtt{X}\alpha_0 \vee \mathtt{E}_{\overline{a}}\mathtt{X}\alpha_0))
\end{aligned}$$

は，$\chi_{12}(n) = \bigwedge_{1\leq i\leq n}(\mathtt{A}_a\mathtt{F}\alpha_i \wedge \neg\alpha_i) \wedge \bigwedge_{0\leq i\leq n}\mathtt{A}_a\mathtt{G}(\alpha_i \rightarrow \neg(\gamma_{n,i} \vee \delta_{n,i,a} \vee \delta_{n,i,\overline{a}})$，$\gamma_{i,j} = \alpha_0 \vee \cdots\alpha_{j-1} \vee \alpha_{j+1}\cdots \vee \alpha_i$，$\delta_{i,j,m} = \mathtt{E}_m\mathtt{X}\alpha_0 \vee \cdots \vee \mathtt{E}_m\mathtt{X}\alpha_{j-1} \vee \mathtt{E}_m\mathtt{X}\alpha_{j+1} \vee \cdots \vee \mathtt{E}_m\mathtt{X}\alpha_i$ と一般化する．ここで，例えば，$\gamma_{i,j}$ は，$\alpha_0 \vee \cdots \vee \alpha_i$ から，α_j を除いたものである． □

5 実験

前節で提案した手法に基づいて作成したベンチマーク用の論理式セットを図 2 に示す．これらの論理式セットは，いずれも，本論文で提案された系統的な手法に基づいて作成されたものである．定義の中で使われている χ_1 から χ_{14} は, 本文を参照してほしい．例えば，test1 は，例 1 で紹介した χ_1 であり，パラメータが 0 のときは，$\neg(p \wedge q \rightarrow p)$ という論理式であることを表している．test1 から test10 は，恒真な $\chi_1(n)$ などの否定であるから充足不可能な論理式である．test8 は，例 3 で紹介したパラメータ付き論理式であり，test9，test10 はその拡張である．test11 から test15 は，充足可能な式である．test15 は，χ_9 と χ_{11} の組み合わせであり，χ_{11} の α_0 として，χ_9 を採用したものである．これらのパラメータ付き論理式の特性を表 1 に示す．ここで，sat は充足可能か (y) 不可能か (n)，AP は原子論理式の種類がパラメータに従って増える (i) か否か (c)，depth は時相演算子のネストの数が増える (i) か否か (c)，Mod は様相記号が増えるか (i) 否か (c)，inverse は，逆の様相記号が現れるか (i) 否か (c) を表している．length は，パラメータに対する論理式の長さを表したもので，linear は線形，quadratic は 2 乗，exp は指数的であることを意味する．表 1 は，2 節で述べた特性に関して，図 2 の論理式セットは十分な多様性を持つことを示し

$$
\begin{array}{lcl}
\mathrm{test1}(n) & = & \neg\chi_1(n)\ (n \geq 1), \quad \neg(p \wedge q \rightarrow p)\ (n = 0) \\
\mathrm{test2}(n) & = & \neg\chi_3(n)\ (n \geq 1), \quad \neg(\mathtt{A}_{a,b}\mathtt{G}p \rightarrow \mathtt{A}_a\mathtt{G}p)\ (n = 0) \\
\mathrm{test3}(0) & = & \neg(\mathtt{AF}P \wedge \mathtt{AG}(\neg S) \rightarrow \mathtt{absence}(P, S)) \\
\mathrm{test3}(n) & = & \neg\chi_5(n), \\
\mathrm{test4}(n) & = & \neg\chi_{13}(n)(n \geq 0) \\
\mathrm{test5}(n) & = & \neg\chi_{14}(n)(n \geq 1),\ \neg(\mathtt{E}[p\ \mathtt{U}\ q] \rightarrow \mathtt{EF}q)\ (n = 0) \\
\mathrm{test6}(0) & = & \neg(\mathtt{AX}p \wedge \mathtt{AX}q \rightarrow \mathtt{AX}(p \wedge q)) \\
\mathrm{test6}(n) & = & \neg\chi_6(n)\ (n \geq 1), \\
\mathrm{test7}(0) & = & \neg(\mathtt{E}[p\ \mathtt{U}\ q] \rightarrow \mathtt{EF}q) \\
\mathrm{test7}(n) & = & \neg(\varphi_n \rightarrow \psi_n)\ (n \geq 1), \\
 & & \varphi_i = \mathtt{EF}a_i\varphi_{i-1}, \psi_i = \mathtt{EF}a_i\psi_{i-1} \\
\mathrm{test8}(n) & = & \neg((\mathtt{EX}(u \vee v) \wedge \bigwedge_{1\leq i\leq n} \mathtt{A}[p_i\ \mathtt{U}\ q]) \rightarrow \\
 & & (\mathtt{EX}u \wedge \mathtt{EX}v \wedge \bigwedge_{1\leq i\leq n} q \vee \mathtt{AXA}[p_i\ \mathtt{U}\ q])) \\
\mathrm{test9}(n) & = & \neg((\mathtt{E}_{a_0}\mathtt{X}(u \vee v) \wedge \bigwedge_{1\leq i\leq n} \mathtt{A}_{a_i}[p_i\ \mathtt{U}\ q]) \rightarrow \\
 & & (\mathtt{E}_{a_0}\mathtt{X}u \wedge \mathtt{E}_{a_0}\mathtt{X}v \wedge \bigwedge_{1\leq i\leq n} q \vee \mathtt{AXA}_{a_i}[p_i\ \mathtt{U}\ q])) \\
\mathrm{test10}(n) & = & \neg((\mathtt{E}_{a_0}\mathtt{X}(u \vee v) \wedge \bigwedge_{1\leq i\leq n} p \wedge \mathtt{E}_{a_i}\mathtt{X}p \rightarrow \\
 & & (\mathtt{E}_{a_0}\mathtt{X}u \wedge \mathtt{E}_{a_0}\mathtt{X}v \wedge \bigwedge_{1\leq i\leq n} \mathtt{E}_{a_i}\mathtt{XE}_{\overline{a_i}}\mathtt{X}p) \\
\mathrm{test11}(n) & = & \chi_9(n)\ (n \geq 1), \quad p \wedge q\ (n = 0) \\
\mathrm{test12}(n) & = & \chi_{10}(n)\ (n \geq 0) \\
\mathrm{test13}(0) & = & \mathtt{A}_a\mathtt{G}(\mathtt{E}_a\mathtt{X}(p \wedge q) \wedge \mathtt{E}_a\mathtt{X}(\neg q \wedge p)) \wedge \mathtt{A}_{\overline{a}}\mathtt{F}\neg p \\
\mathrm{test13}(n) & = & \chi_{11}(n)\ (n \geq 1) \\
\mathrm{test14}(n) & = & \chi_{12}(n)\ (n \geq 0) \\
\mathrm{test15}(n) & = & \chi_{11}(n)[\chi_9(n)/\alpha_0]\ (n \geq 1),\ \chi_9(0)\ (n = 0)
\end{array}
$$

図 2: 論理式セット

表 1: 論理式セットの特性

name	sat	AP	depth	Mod	inverse	legnth
test1	n	c	i	i	n	linear
test2	n	c	i	c	y	exp
test3	n	i	i	c	n	linear
test4	n	i	i	c	n	linear
test5	n	c	i	i	n	linear
test6	n	c	i	c	y	quadratic
test7	n	c	i	i	n	quadratic
test8	n	i	c	c	n	linear
test9	n	i	c	i	n	linear
test10	n	i	c	i	y	linear
test11	y	c	i	c	n	quadratic
test12	y	i	i	i	y	exp
test13	y	c	c	i	n	linear
test14	y	i	c	c	y	linear
test15	y	c	i	i	n	quadratic

表 2: ベンチマーク結果

test1	16	test6	20	test11	>20
test2	13	test7	13	test12	3
test3	7	test8	11	test13	>20
test4	3	test9	7	test14	6
test5	14	test10	13	test15	5

表 3: 実験環境

OS	Red Hat Enterprise Linux ES3
CPU	Intel Xeon 3.0GHz
Memory	4GB
JVM	Java2 1.5.0_03

ている．充足不可能な論理式はすべて簡単な論理式から自動的に生成されたものであり，本手法の有効性が確かめられた．

図 2 の論理式セットをもちいて，著者らが開発している充足可能性判定器 [12] のベンチマークを行った．測定は，100 秒間に充足可能性を判定できた論理式の累積数を対象とした．測定結果を表 2 に示す．また，測定環境は表 3 のとおりである．Balsiger らは，各パラメータに対して論理式の判定時間を測定し，100 秒以内に判定できたパラメータの最大値を測定値としている．表 2 に示されているのは累積数であるが，2 方向 CTL の充足可能性判定の計算量の下限は EXPTIME なので [4]，自然数パラメータが増えるにしたがって判定時間は指数的に増加するため，Balsiger らの測定とほぼ同等の測定基準による結果である．

6 まとめ

本論文では，2 方向 CTL 論理の充足可能性判定器に対する性能評価に使うための，ベンチマーク用論理式の自動生成法について提案した．2 方向 CTL を例にとり説明してきたが，$\alpha \rightarrow \beta$ という形の単純な恒真式から複雑な恒真式を得るという方針は，他の多くの論理体系にも応用できると予想する．

今回，充足不可能なものについては，自動的に複雑な論理式を得るための手法を提案でき提案できたが，充足可能なものについては, 指針を述べるにとどまった．充足可能性判定において，充足不可能なものを示すほうが難しく，評価のために重要であるとはいえ，充足可能な論理式の自動生成については今後の課題である．

参考文献

[1] P. Balsiger, A. Heuerding, and S. Schwendimann, “A benchmark method for the propositional modal logics K,KT,S4,” J. of Automated Reasoning, vol. 24, no. 3, pp. 297–317, 2000.

[2] E.M. Clarke, O. Grumberg, and D. Peled, Model Checking, Mit Press, 2000.

[3] M.B. Dwyer, G.S. Avrunin, and J.C. Corbett, “Paterns in property specifications for finite-state verification,” Proc. of 21st International Conference on Software Engineering, May, 1999.

[4] E.A. Emerson, “Temporal and modal logic,” in Handbook of Theoretical Computer Science, vol. B, chap. 16, pp. 995–1072, Elsevier and MIT Press, 1990.

[5] E.A. Emerson, and E.M. Clarke, “Using branching-time temporal logic to synthesize synchronization skeltons,” Science of Computer Programming, vol. 2, no. 3, pp. 241–266, 1982.

[6] M. Hagiya, K. Takahashi, M. Yamamoto, and T. Sato, “Analysis of synchronous and asynchronous cellular automata using abstraction by temporal logic,” Proc. of 7th International Symposium on Functional and Logic Programming, LNCS, vol. 2998, pp. 7–21, 2004.

[7] M. Huth, and M. Ryan, Logic in Computer Science: modelling and reasoning about systems, Cambridge University Press, 2004.

[8] M. Lange, and C. Stirling, “Focus games for satisfiability and completeness of temporal logic,” Proc. of 16th Annual IEEE Symposium on Logic in Computer Science, pp. 357–365, 2001.

[9] Z. Manna, and P. Wolper, “Synthesis of communicating process from temporal logic specifications,” ACM Transactions on Programming Languages and Systems, vol. 6, no. 1, pp. 68–93, 1984.

[10] K. Takahashi, and M. Hagiya, “Abstraction of graph transformation using temporal formulas,” Supplemental Volume of International Conference on Dependable Systems and Networks (DSN-2003), pp. W-65–W-66, 2003.

[11] Y. Tanabe, T. Takai, T. Sekizawa, and K. Takahashi, “Preconditions of properties described in CTL for statements manipulating pointers,” Supplemental Volume of International Conference on Dependable Systems and Networks (DSN2005), pp.228–234, June 28-July 1, 2005.

[12] 田辺良則，高橋孝一，山本光晴，佐藤貴洋，戸沢晶彦，萩谷昌己, “BDD による実装が可能な様相論理の充足可能性判定手続き”, 第 7 回プログラミングおよびプログラミング言語ワークショップ, PPL2005, pp.5-16, 2005.

[13] 田辺良則，高橋孝一，山本光晴，佐藤貴洋，萩谷昌己, “BDD を用いた 2 方向 CTL 論理式充足可能性決定手続きの実装,” コンピュータソフトウェア. (掲載予定)

[14] A. Tozawa, “On binary tree logic for XML and its satisfiability test,” 第 6 回プログラミングおよびプログラミング言語ワークショップ, PPL2004, 2004.

時相論理の充足可能性判定器のためのベンチマーク用論理式生成法
(算譜科学研究速報)
発行日：2005 年 9 月 22 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

A method to generate benchmark formulae
for temporal logic satisfiability checkers (preliminary version)
(in Japanese)
(Programming Science Technical Report)
Sep. 22, 2005
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