社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2003－ＳＬＤＭ－112　(11)
2003／11／28

時間制約付き制御フローグラフの弱双模倣等価性検証およびWebセキュリティ検査への応用

佐々木俊　谷本匡亮　中田明夫　東野輝夫

大阪大学大学院情報科学研究科　〒560-0043　大阪府豊中市待兼山町 1-3

E-mail:　{s-sasaki, tanimoto, nakata, higashino}@ist.osaka-u.ac.jp

あらまし　近年、実時間制約を伴うハードウェア・ソフトウェアプログラムの効率的な開発手法が求められている。プログラム開発においては、プロファイラを用いた分岐構造の変更などが行われるが、そのような場合においても、入出力など外部観測点の間に指定された実時間制約が保存されていることを検証したい。そのような検証には、Global Timed Bisimulation の検証が有用である。本論文では、実時間制約が与えられたプログラムから制御構造と時間制約を抽出したモデルである、Timed CFG(Control Flow Graph)の Global Timed Bisimulation 等価性検証を、時間強双模倣等価性検証に帰着して行う手法を提案する。Web ブラウザの Timing Attack に対する脆弱性検証への応用例も示す。

キーワード　実時間制約，Timed CFG，Global Timed Bisimulation，時間強双模倣，Timing Attack

Global Timed Bisimulation on Timed Control Flow Graph And Application to Checking for Timing Attacks on Web Privacy

Suguru SASAKI　Tadaaki TANIMOTO　Akio NAKATA　and　Teruo HIGASHINO

Graduate School of Information Science and Technology, Osaka University

1-3 Machikaneyama-chou, Toyonaka-shi, Osaka, 560-0043 Japan

E-mail:　{s-sasaki, tanimoto, nakata, higashino}@ist.osaka-u.ac.jp

Abstract.　In recent years, the effective development methodology for software with real-time constraints is desired. Programmers sometimes change blanch structure in their codes to improve performance. This may change timing of I/O action of programs. Thus, programmers want to check whether timing behavior of I/O action is equivalent between original codes and optimized ones. In such verification, we think that verification of global timed bisimulation equivalence is useful. In this paper, we propose timed CFGs (Control Flow Graph), which represent control structure with real-time constraints of real-time software. And we show that verification of global timed bisimulation equivalence for timed CFGs can result in existing parametric verification of strong timed bisimulation equivalence. We also apply our approach to checking for timing attacks on Web privacy.

Keyword　Real Time Constraints，Timed CFG，Global Timed Bisimulation，Strong Timed Bisimulation，Timing Attack

1. 背景

近年、携帯電話などの組み込みシステムは勿論の事、輸送、医療、防衛などの比較的大規模なシステムでさえ、その多くが実時間制約を伴うものとなって来ており、実時間システムの効率的な開発手法がますます重要となって来ている。また、実時間制約を持つプログラムの開発においては、実時間制約を保障するためのコード実行時間の最適化や、分岐構造の組み換えによる最適化などがプロファイラ等を用いて行われる。

実時間性と外部観測性を同時に考慮したプロセスの等価性として Timed Weak Bisimulation 等価が提唱されている[7][8]が、内部動作の分岐による違いを考慮するため、プログラムの分岐構造組み換えによる最適化が実施されると、本来等価であるところを等価でないと判定してしまう場合がある。

そこで、本論文では、実時間性と外部観測性を同時に考慮し、かつ内部動作の分岐構造の違いを無視した等価性である Global Timed Bisimulation 等価[2]の等価性判定手法の提案を

行う。まず、入出力動作とその実行時間に関する実時間制約のみが与えられたスケルトン・コード（図1参照）、及び具体的な内部動作と入出力動作間の時間制約を内部動作の時間制約へ詳細化した実装プログラムを読み込み、内部表現としてCFG（Control Flow Graph）に時間遷移の情報を付加した中間表現（Timed CFG）にそれぞれ変換する。CFGはソフトウェア・コンパイラで広く用いられている中間表現であり、これを対象とする事で特定のプログラム言語に依存しない検証手法の提案が可能となると考えている。次に、実時間制約を保存しながら内部動作を1つにまとめる等の変換を行うことにより、EFSMに実時間拡張を行ったモデル（Timed EFSM）を得る。これに既存のStrong Timed Bisimulation等価性判定法を用いる事で、スケルトン・コードと実装プログラムの等価性判定、即ちスケルトン・コードに記述されている入出力動作及びそれらの実時間制約の情報が実装コードで保持されているかの検証、が可能となる。

```
/* 0≦T≦20 */
(read a)
... = a;
...
(write b)
b = ...;
...
```

図1．スケルトン・コード例

以降、2章では、Timed CFGの定義、及びラベル付遷移システムとの対応付けによる操作意味の定義を与え、3章で、Timed CFG上のGlobal Timed Bisimulationの定義を述べる。4章では、Timed CFGの変換アルゴリズムの提案、及び変換アルゴリズムがGlobal Timed Bisimulation等価を保存する事の説明を行い、5章で、Timed CFGのGlobal Timed Bisimulation等価性判定法を提案し、Timing Attackに対する脆弱性の検証に適用可能である事を述べる。最後に6章において、結論と将来の課題を述べる。

2. Timed CFG

本章では、Timed CFGの定義、及びその操作的意味定義を与える。尚、操作的意味定義は、実時間拡張を行ったEFSM（Timed EFSM）と対応させる事により行う。

2.1. Timed CFGの定義

定義1　図2のようなグラフを時間制御フローグラフ（Timed Control Flow Graph, 以下Timed CFG）と呼ぶ。　□

Timed CFGは分岐枝の開始と終了を表すノードと，ループの開始と終了を表すノードを含む。特に，クロック境界ノードを枝上に付加することで、クロック・ティックを表現する。各クロック境界には線形不等式による実時間制約が付加されており、特に1単位時間消費の場合には時間制約の記述を省略できる。尚、本論文では、再帰は対象外とする。再帰以外の関数呼び出しに関しても、全てインライン展開する事で除去可能である為、本論文では扱わない。

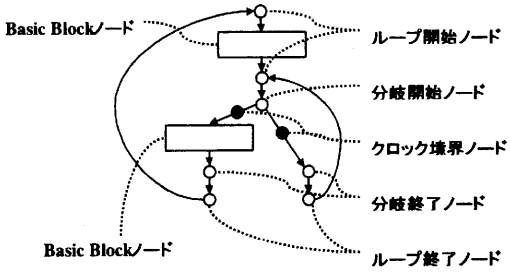

図2．Timed CFGの表現形式

2.2. Timed CFGの操作的意味定義

CFGの動作は各ノードを状態、ノード間の有向枝を遷移とみなした状態機械に対応するものとする。また、分岐は非決定性分岐を表すとする。各ノードの時間的な動きは次の通りである。まず、分岐開始、分岐終了、ループ開始、ループ終了の各ノードに関しては時間を消費せず、後続ノードに制御が移るものとする。次に、クロック境界ノードにおいては、$e1 \leqq t \leqq e2$ （$e1, e2$ は整数定数、変数および加減算のみを含む式。但し、先行するクロック境界ノードで用いられている時間変数は参照不可）という時間制約が付加されている場合は、この制約を満たすある時間 t 経過後に、後続ノードに制御が移るものとする。時間制約が何も付加されていない場合は、1単位時間を消費して後続ノードに制御が移るものとする。

形式的には各CFGを以下に定義する実時間EFSM（Timed EFSM）に対応させることによって、動作を定義する。まず、変数の集合を Var とし、Var に含まれる変数を用いた実数プレスブルガ式の全体集合を $Pred(Var)$ とする。

定義2　実時間**EFSM**とは5字組$(S, Act, PVar, E, s_{init})$である。但し、$S$ は状態の有限集合、Act は動作名の有限集合、$PVar \subseteq Var$ はパラメータ変数の有限集合、$E \subseteq S \times Act \times Var \times Pred(Var) \times S$ は遷移関係の有限集合、$s_{init} \in S$ は初期状態である。任意の $s,s' \in S$, $a \in Act$, $d \in Var$, $P \in Pred(Var)$ に対して、$(s,a,d,P,s') \in E$ を

$$s \xrightarrow{a@?d[P]} s'$$

と略記する。　□

実時間EFSMのセマンティクスは、状態 s と変数への代入 σ の組(s, σ)を状態とするラベル付遷移システムとして定義される。例えば、遷移 $s \xrightarrow{a@?t[t \leqq 5+p]} s'$ と代入 $\sigma=(p=2,t=0)$ に対して、以下のような遷移関係が定義される。

$$(s, \sigma) \xrightarrow{7} (s,(p=2,t=7)) \xrightarrow{a} (s',(p=2,t=7))$$

$\xrightarrow{7}$ は7単位時間の経過を表し、$\xrightarrow{a}$ は動作 a が実行され

ることを表す。また、状態$(s, \sigma+t)$で状態(s, σ)からt単位時間が経過した状態を表すとする。

最後に、CFGから実時間EFSMへの対応づけを定義する。具体的にはCFG Gに対して、以下のような実時間EFSM $M(G)$を定義する。$M(G)$の制御状態集合はGのノード集合とし、Gの各ノードに対して以下のように$M(G)$の遷移を対応づける。時間を消費しないノードに関してはそのノードに対応する状態sから後続のノードに対応する状態s'への遷移$s \xrightarrow{a@?t[t=0]} s'$を対応づける。同様に、時間制約$e1 \leqq t \leqq e2$を持つノードに関しては遷移$s \xrightarrow{a@?t[e1 \leqq t \leqq e2]} s'$を、時間制約を持たないクロック境界ノードに関しては遷移$s \xrightarrow{a@?t[t=1]} s'$を対応づける。

3. Global Timed Bisimulation

本章では、文献[2]で提案されたGlobal Timed Bisimulationの定義、及びGlobal Timed BisimulationとTimed Weak Bisimulationの関係を述べる（詳細は文献[2]を参照）。

3.1. Global Timed Bisimulationの定義

Global Timed Bisimulationの定義に先立って、まずTimed Weak Bisimulationの定義を与える。

定義3 **Timed Weak Transition Relation** $\rightarrow_w$は、ラベル付遷移システム$P=(CS, Act \cup \mathbf{R}+ \cup \{\tau\}, CE, (s0, \sigma 0))$の状態$(s, \sigma)$ (s', σ')に対して以下のように定義される（$\mathbf{R}+$：非負実数）。

(1) $\xrightarrow{\tau}_w \stackrel{def}{:=} (\xrightarrow{0} \xrightarrow{\tau})^*$

(2) $(s, \sigma) \xrightarrow{t}_w (s', \sigma')$ $(t \in \mathbf{R}+)$

$\stackrel{def}{:=} \exists t1, t2, \ldots, tn \in \mathbf{R}+;\ \Sigma t_i = t\ \wedge$

$\exists (s1, \sigma 1), (s1, \sigma 1'), (s2, \sigma 2), (s2, \sigma 2'), \ldots,$

$(sn, \sigma n), (sn, \sigma n')$

$s.t.\ (s, \sigma) \xrightarrow{\tau}_w (s1, \sigma 1) \xrightarrow{t1} (s1, \sigma 1') \ldots$

$\xrightarrow{\tau}_w (sn, \sigma n) \xrightarrow{tn} (sn, \sigma n') \xrightarrow{\tau}_w (s', \sigma')$

(3) $\xrightarrow{a}_w (a \in Act) := \xrightarrow{\tau}_w \xrightarrow{a} \xrightarrow{\tau}_w$ □

定義4 ラベル付遷移システム上の二項関係Rが**Timed Weak Bisimulation**であるとは

「$P R Q \Rightarrow \forall \alpha \in Act \cup \mathbf{R}+ \cup \{\tau\}$に対して

(1) $P \xrightarrow{\alpha} P' \Rightarrow \exists Q';\ Q \xrightarrow{\alpha}_w Q' \wedge P' R Q'$

(2) $Q \xrightarrow{\alpha} Q' \Rightarrow \exists P';\ P \xrightarrow{\alpha}_w Q' \wedge P' R Q'$」

が成立する事である。尚、

「$\exists R$: Timed Weak Bisimulation; $P R Q$」

を「$P \equiv_{twb} Q$」と書く。 □

以下では、Static Generalized Transition Relation $\rightarrow_{gt}$、及びDynamic Generalized Transition Relation $\xrightarrow{\alpha}_{gt}$ $(\alpha \in Act \cup \mathbf{R}+)$の定義を述べ、最後にGlobal Timed Bisimulationの定義を述べる。

定義5 **Static Generalized Transition Relation** $\rightarrow_{gt}$、及び**Dynamic Generalized Transition Relation** $\xrightarrow{\alpha}_{gt}$ $(\alpha \in Act \cup \mathbf{R}+)$は、ラベル付遷移システムP、P'に対して以下のように定義される。

(1) Pの Transition Relationの保存

(i) $P \rightarrow P'$ ならば、$P \rightarrow_{gt} P'$

(ii) $\exists \alpha \in Act \cup \mathbf{R}+; P \xrightarrow{\alpha} P'$ ならば、$P \xrightarrow{\alpha}_{gt} P'$

(2) 内部遷移の同時実行

$P = \Sigma_\alpha \alpha; P_{\alpha i} + \Sigma_{i \in I} \tau; P_i \rightarrow_{gt} \Sigma_{i \in I} \tau; P_i$

(3) 非決定性の除去（同一アクション実行時）

$P = \Sigma_\alpha \alpha; P_{\alpha i} + a; P_a + \Sigma_{i \in I} a; P_i \rightarrow_{gt} \Sigma_\alpha \alpha; P_{\alpha i} + a; P_a$

(4) 内部動作の実行

$P_\beta \rightarrow_{gt} P'_\beta \Rightarrow P = \Sigma_\alpha \alpha; P_{\alpha i} + \beta; P_\beta$

$\rightarrow_{gt} P = \Sigma_\alpha \alpha; P_{\alpha i} + \beta'; P'_\beta$

$(\beta \in Act \cup \{\tau\}$ または $\beta = delay(t) (t \in \mathbf{R}+))$

(5) 外部観測性のあるアクションによる遷移の同時実行

$\forall a \in Act, \forall I \subseteq I_a$ with $I_a = \{i; P \xrightarrow{a}_w P_i\} \Rightarrow P \xrightarrow{a}_{gt} \Sigma_{i \in I} \tau; P_i$

(6) 内部遷移における時計の同期

$\forall i \in I: P_i \xrightarrow{t} P'_i \Rightarrow$

$P = \Sigma_\alpha \alpha; P_{\alpha i} + \Sigma_{i \in I} \tau; P_i \xrightarrow{t}_{gt} \Sigma_{i \in I} \tau; P'_i$

ここで、$\rightarrow$は$\rightarrow_{gt}$の反射及び推移に関する閉方であり、$P \xrightarrow{t} P'$は$t = \Sigma t_i$に対して

$P \rightarrow \cdot \xrightarrow{t1}_{gt} \cdot \cdot \cdot \xrightarrow{tn}_{gt} \cdot \cdot \cdot \rightarrow P'$

を表し、任意の$a \in Act$に対して、$\xrightarrow{a}$は

$\rightarrow \cdot \xrightarrow{a}_{gt} \cdot \rightarrow$ に等しい。 □

定義6 ラベル付遷移システムP、Qの関係Rが**Global Timed Bisimulation**であるとは

「$P R Q \Rightarrow \forall \alpha \in Act \cup \mathbf{R}+ \cup \{\varepsilon\}$に対して

(1) $P \xrightarrow{\alpha}_{gt} P' \Rightarrow \exists Q'; Q \xrightarrow{\alpha} Q' \wedge P' R Q'$

(2) $Q \xrightarrow{\alpha}_{gt} Q' \Rightarrow \exists Q'; P \xrightarrow{\alpha} P' \wedge P' R Q'$」

が成立する事である。ここで、$\xrightarrow{\varepsilon} := (\xrightarrow{\tau})^*$である。尚、

「$\exists R$: Global Timed Bisimulation; $P R Q$」

を「$P \equiv_{gtb} Q$」と書く。 □

3.2. Global Timed BisimulationとTimed Weak Bisimulationの関係

下記定理が成立する(詳細は文献[2]参照)。

定理1 $P \equiv_{twb} P' \Rightarrow P \equiv_{gtb} P'$

（P、P'はラベル付遷移システム）

また、Global Timed Bisimulationでは、下記に示す内部分岐に関する結合法則が成立する(詳細は文献[2]参照)。

$P \equiv_{gtb} P' \wedge P' \equiv_{gtb} P''$

ここで、$P = \tau; (\tau; P_1 + \tau; P_2) + \tau; P_3$

$P' = \tau; P_1 + \tau; (\tau; P_2 + \tau; P_3)$

$P'' = \tau; P_1 + \tau; P_2 + \tau; P_3$

よって、内部動作の分岐による違いを識別する Timed Weak Bisimulation では等価とならない実時間 EFSM が、下記に示すように Global Timed Bisimulation では等価となる。

$Q_1 \equiv_{gtb} Q_2$

$Q_1 = \tau;\delta;(\tau;a + \tau;b) + \tau;\delta;(\tau;c + \tau;d)$

$Q_2 = \delta;(\tau;a + \tau;b + \tau;c + \tau;d)$

$R_1 \equiv_{gtb} R_2$

$R_1 = \delta;(\tau;a + \tau;2\delta;(\tau;b + \tau;c))$

$R_2 = \tau;\delta;a + \tau;3\delta;b + \tau;3\delta;c$

4. Property Preserving Transformation on Timed CFG

本章では、時間制約を保存しつつ内部動作を徐々に削除する変換規則の定義を述べ、その変換規則が Global Timed Bisimulation を保存する事を示し、変換規則を実装するアルゴリズムが停止する事を示す

4.1. Transformation の定義

まず、変換規則の実時間 EFSM 上でのシンボリックな定義を与える。尚、本論文で対象としている Timed CFG では、時間制約が線形不等式のみで与えられ、時間パラメータが一つしか存在しないため、下記に示す可換環が定義可能となり、それ故、時間制約を保存しつつ内部動作を削除する変換規則が定義可能となる。

定義 7 $\mathbf{X} := \{\{t; c1 \leqq t \leqq c2\}; c1,c2 \in \mathbf{R}+\}$ に対して、$\mathbf{X}$ 上の二項演算 Θ を以下で定義する。

$x = \{t; x1 \leqq t \leqq x2\}, y = \{t; y1 \leqq t \leqq y2\} \in \mathbf{X}$

$x \Theta y \overset{def}{:=} \{t; \exists t1,t2 \text{ s.t. } t = t1 + t2 \wedge t1 \in x \wedge t2 \in y\}$

$= \{t; x1+y1 \leqq t \leqq x2+y2\} \in \mathbf{X}$ □

今、$\mathbf{Pred} := \{X_1 \vee X_2 \vee \cdots \vee X_n; X_1,\ldots,X_n \in \mathbf{X}, n \in \mathbf{N}\}$、($\mathbf{N}$：自然数)とすると、$\vee$ は **Pred** 上の自然な二項演算となっており、それ故、二項演算 Θ を改めて $pred_1, pred_2 \in \mathbf{Pred}$ に対して

$pred_1 \Theta pred_2 := \{t; \exists t1,t2 \text{ s.t. } t = t1 + t2 \wedge$

$t1 \in pred_1 \wedge t2 \in pred_2\}$

と定める事で、Θ は **Pred** 上の演算に拡張可能となる。また、下記定理が成立する。

定理 2 **Pred** 上の二項演算 Θ 及び $\vee$ は $\{t:=0\}$ を単位元、空集合 Φ を零元として **Pred** 上の可換環を成す。即ち、下記が成立する。

$\forall t,s,u \in \mathbf{Pred}$,

(1) 交換法則　$t\Theta s = s\Theta t$

(2) 結合法則　$t\Theta(s\Theta u) = (t\Theta s)\Theta u$

(3) 単位元（$1 := \{t==0\} \in \mathbf{Pred}$）　$t\Theta 1 = t$

(4) 分配法則　$(t \vee s)\Theta u = t\Theta u \vee s\Theta u$

$t\Theta(s \vee u) = t\Theta s \vee t\Theta u$

(5) 零元（$0 := \Phi$）　$t \vee 0 = t$

$t\Theta 0 = 0$

定義 8　$x = \{t; x1 \leqq t \leqq x2\} \in \mathbf{X}$ に対して、ループ演算 l を

$l(x) := \{lt; t \in x, l \in \mathbf{N} \cup \{0\}\}$

$= \{t; t=0\} \vee_{k \in \mathbf{N}} \{t; kx1 \leqq t \leqq kx2\} \in \mathbf{Pred}$

で定める。また、$x = \{t; x1 \leqq t \leqq x2\}, y = \{t; y1 \leqq t \leqq y2\} \in \mathbf{X}$ に対して、

$l(x \vee y) := l(x) \Theta l(y) \in \mathbf{Pred}$

と定める事で **Pred** 上の演算として定義する。 □

定理 2 より、Pred 上の二項演算 Θ は交換可能であり、また $\vee$ も交換可能である事に注意すると、$l(x)\Theta l(y)$ は時間制約 x でのループと時間制約 y でのループに於いて、それぞれの任意の実行回数の組み合わせに対応する時間経過を表現している事がわかる。一方、左辺の $l(x \vee y)$ は時間制約 x または時間制約 y でループを実行した場合の時間経過を表しており、これは $l(x)\Theta l(y)$ が表す時間経過と一致する。従って、ループ演算の Pred 上での定義は Well defined である。

定義 9　T-EFSM 上での下記変換を定義する。

ここで、　Abs : T-EFSM → T-EFSM とする。

(1) $s0 \xrightarrow{\tau @?t[P]} s1 \xrightarrow{\alpha @?t[Q]} s2$

$\xrightarrow{Abs} s0 \xrightarrow{\alpha @?t[P \Theta Q]} s2$ ($\alpha \in Act \cup \{\tau\}$, $P, Q \in \mathbf{Pred}$)

(2) $s0 \xrightarrow{\tau @?t[P]} s0 \wedge s0 \xrightarrow{\alpha 1 @?t[Q_1]} s1 \wedge$

$s0 \xrightarrow{\alpha 2 @?t[Q_2]} s2 \wedge \cdots \wedge s0 \xrightarrow{\alpha n @?t[Q_n]} sn$

$\xrightarrow{Abs} s0 \xrightarrow{\alpha 1 @?t[l(P) \Theta Q_1]} s1 \wedge$

$s0 \xrightarrow{\alpha 2 @?t[l(P) \Theta Q_2]} s2 \wedge \cdots \wedge$

$s0 \xrightarrow{\alpha n @?t[l(P) \Theta Q_n]} sn$

($\alpha 1, \alpha 2, \cdots, \alpha n \in Act \cup \{\tau\}$, $P, Q_1, \ldots, Q_n \in \mathbf{Pred}$)

(3) $s0 \xrightarrow{\alpha 1 @?t[P_1]} s1 \xrightarrow{\alpha 2 @?t[P_2]} s2 \cdots\cdots \xrightarrow{\alpha n @?t[P_n]} sn \wedge$

$s0 \xrightarrow{\beta 1 @?t[Q_1]} s1', \xrightarrow{\beta 2 @?t[Q_2]} s2', \cdots\cdots \xrightarrow{\beta n @?t[Q_n]} sn$

$\xrightarrow{Abs} s0 \xrightarrow{\alpha 1 @?t[P_1]} s1 \xrightarrow{\alpha 2 @?t[P_2]} s2 \cdots\cdots \xrightarrow{\alpha n @?t[P_n]} sn$

($\alpha 1, \alpha 2, \cdots, \alpha n, \beta 1, \beta 2, \cdots, \beta n \in Act \cup \{\tau\}$;

$\alpha_i = \beta_i \wedge P_1, P_2, \cdots P_n, Q_1, Q_2, \cdots, Q_n \in \mathbf{Pred}$; $P_i = Q_i$) □

(1)が時間制約を保存しつつ連続する遷移を一つの遷移に纏める変換、(2)が内部動作のみからなるループの変換、(3)が分岐の開始と終了が一致していて且つ遷移列の構造と時間制約が等しいものは一つの遷移列のみ残すという変換を表している。

上記定義に対応する、ラベル付遷移システム上での変換規

則の定義を下記に述べる。

定義１０　ラベル付遷移システムP上での下記変換を定義する。ここで、P_1は再帰を含まないラベル付遷移システム。

(1) $P := P_1;delay(t1);\tau;delay(t2);\alpha;P_1'$
$\Rightarrow Abs(P) = P_1;delay(t1+t2);\alpha;P_1'$　for $\forall \alpha \in Act \cup \{\tau\}$

(2) $P := P_1;P_2 \quad P_2 := delay(t);\tau;P_2 + P_3$
$\Rightarrow Abs(P) = P_1;\Sigma_{l \in N \cup \{0\}} delay(lt);\tau;P_3$

(3) $P := P_1;(delay(t);\alpha;P_2 + delay(t);\alpha;P_2);P_3$
$\Rightarrow Abs(P) = P_1;delay(t);\alpha;P_2;P_3$　for $\forall \alpha \in Act \cup \{\tau\}$ □

4.2. Preserved Property by the Transformation

前節で定義した変換規則においては、Global Timed Bisimulationが保存される。証明は帰納法等を用いて行うことが出来るが、紙面の都合上割愛させていただく。以下では、その結果のみ示す（$\alpha \in Act \cup \{\tau\}$）。

補題１　P_1、P_2をラベル付遷移システムとする。

(1) $\forall \rho \in Act \cup \{\varepsilon\} \cup \mathbf{R}+$ に対して、
$\exists P_\alpha$：ラベル付遷移システム
s.t. 「$P_1 \overset{\rho}{\rightarrow}_w P_\alpha$ ←→ $P_2 \overset{\rho}{\rightarrow}_w P_\alpha$」
$\Rightarrow P_1 \equiv_{twb} P_2$

(2) $\forall \rho \in Act \cup \{\varepsilon\} \cup \mathbf{R}+$ に対して、
$\exists P_\alpha$：ラベル付遷移システム
s.t. 「$P_1 \overset{\rho}{\rightarrow}_{gt} P_\alpha$ ←→ $P_2 \overset{\rho}{\rightarrow}_{gt} P_\alpha$」
$\Rightarrow P_1 \equiv_{gtb} P_2$

補題２　$P := delay(t1);\tau;delay(t2);\alpha;P_2 \Rightarrow P \equiv_{twb} Abs(P)$

定理２　$P := P_1;delay(t1);\tau;delay(t2);\alpha;P_2$
$\Rightarrow P \equiv_{gtb} Abs(P)$

補題３　$P := (delay(t);\alpha;P_2 + delay(t);\alpha;P_2);P_3$
$\Rightarrow P \equiv_{twb} Abs(P)$

定理３　$P := P_1;(delay(t);\alpha;P_2 + delay(t);\alpha;P_2);P_3$
$\Rightarrow P \equiv_{gtb} Abs(P)$

補題４　$P := delay(t);\tau;P_1 + P_2 \Rightarrow P \equiv_{gtb} Abs(P)$

定理４　$P := P_1;P_2 \wedge P_2 := delay(t);\tau;P_2 + P_3$
$\Rightarrow P \equiv_{gtb} Abs(P)$

4.3. Transformation Algorithm

本節にて、４．１にて定義した変換規則の具体的なアルゴリズムを与え、その停止性を示す。

定理５　$\forall$P:ラベル付遷移システム、$\exists n \in N$ s.t. $Abs^n(P)$は内部動作を含まない

証明　定義１４のAbs()の定義からAbs()が内部遷移の数を減少させる事、及びPに含まれる内部動作遷移の数に関する帰納法により示される。 ■

系１　下記アルゴリズムが停止する。

```
Input P; /* Timed CFG P の入力 */
Do {
P = MargeConcequtiveTran(P);      /*連続遷移のマージ*/
P = DeleteRedundantBlanch(P);     /*分岐の削除*/
P = LoopTran(P);                  /*ループの変換*/
}while(P が変化)
```

5. Global Timed Bisimulation Checking on Abstracted Timed CFG

本章にて、提案するTimed CFGのGlobal Timed Bisimulation等価性判定がTimed Strong Bisimulation等価性判定に帰着される事を述べ、適用事例として、連続時間でのTiming Attackの検出手法を述べる。

5.1. Algorithm

２つのTimed CFG P及びQに対して系１のアルゴリズムを適用すると、双方ともGlobal Timed Bisimulation等価で且つ内部動作を含まないAbs(P)及びAbs(Q)を得る。一方、文献[6]の結果よりAbs(P)とAbs(Q)がTimed Strong Bisimulation等価となるためのパラメータ条件が得られる。Timed Strong Bisimulation等価であればGlobal Timed Bisimulation等価であり、またGlobal Timed Bisimulation等価性の推移律より、求めたパラメータ条件はPとQがGlobal Timed Bisimulation等価であるためのパラメータ条件となる。

定理６　P,Q：Timed CFG
$\exists n,m$; $Abs^n(P)$：内部動作を含まない
$\wedge$ $Abs^m(Q)$：内部動作を含まない
$\wedge$ $Abs^n(P) \equiv_{tsb} Abs^m(Q)$
$\Rightarrow P \equiv_{gtb} Q$

5.2. Application to check Vulnerability from Timing Attack on Web

Timing Attack on Web[10]とは、以下に示すネットワーク上の攻撃を指す。

(i) 悪意を持ったWebがダウンロード形式のアプレットを用意。

(ii) アプレットがダウンロード後実行されると、Webキャッシュ上に対象とするWebが登録されているかを悪意を持ったWebから観測。

即ちキャッシュヒット時とミス時の応答速度の違いに注目した攻撃である。

さて、文献[10]によれば、Timing Attack に対する脆弱性の検証は下記等価性検証により実現可能である。

「∀Π:被攻撃者の Web キャッシュ記録,

$B \setminus H \equiv_{twb} (B \| \Pi) \setminus H$ 」

尚、ここで、

B：アプレットの動作を表すオートマトン、

H：B 上の Web キャッシュヒットの動作、

B‖Π：B とΠの積オートマトン、

B＼H：B 上で動作 H を禁止した LTS

であり、特に厳しい条件として、Π として常に悪意ある Web が興味を持つ Web サイトが Web キャッシュヒットするものを用いて検証を実施すれば良い。

連続時間の Timed Weak Bisimulation の等価性判定法が未だ報告されていないため、離散時間での検証のみが可能となっている。しかし、現実の世界は連続時間であり、連続時間での検証が必要である。本論文が提案する Global Timed Bisimulation 等価性判定法により、連続時間での検証法へ既存検証法を拡張可能となる。また、本論文が提案する Global Timed Bisimulation 等価性判定法においては、得られたパラメータ条件式にループ回数のパラメータが時間パラメータの係数となるため、一般的にはプレスブルガでは無いという問題があるが、得られたパラメータ式の否定を取り、ループ回数のパラメータを係数として持つ時間パラメータを新たな時間パラメータに置換する事でプレスブルガ式に変換可能であり、得られたプレスブルガ式が充足されれば、Timing Attack に対する脆弱性があると判断可能である。従って、本論文が提案する Global Timed Bisimulation 等価性判定法は Timing Attack に対する脆弱性の検証に適用可能であると考えられる。

6. Conclusion and Future Directions

本論文では、Timed CFG を提案し、Timed CFG 上で Global Timed Bisimulation を保存しつつ内部動作を含まないよう変換するアルゴリズムを提案した。それにより、Global Timed Bisimulation 等価性判定法を既存の Strong Timed Bisimulation 等価性判定法に帰着させた。また、提案する等価性判定法が Timing Attack に対する脆弱性の検証に適用可能である事を示した。

一方、提案した変換アルゴリズムは、時間オートマトン上の(パラメトリック)モデル検査法における抽象化に利用可能であると考えられるため、今後 Global Timed Bisimulation 等価な動作を識別できない適当な時相論理式のクラスが存在する事を示していく計画である。また、ループ回数の変数を係数として持つ時間パラメータの条件式を如何に扱うかも今後の研究課題である。

参考文献

[1] Jia Xu, "On Inspection and Verification of Software with Timing Requirements", IEEE Trans. on Software Engineering, Vol.29, No.8, pp.705-720, Aug. 2003.

[2] David de Frutos, Natalia López and Manuel Núñez, " Global Timed Bisimulation: An Introduction", In Proceedings of Joint Conference on Formal Description Techniques for Distributed Systems and Communication Protocols XII, and Protocol Specification, Testing, and Verification XIX (FORTE/PSTV'99), pages: 401-416, 1999.

[3] M. C. B. Hennessy and H. Lin. "Symbolic bisimulations", Theoretical Computer Science, 138:353–389, 1995..

[4] Z. Li and H. Chen. "Computing strong/weak bisimulation equivalences and observation congruence for value-passing processes", In Tools and Algorithms for the Construction and Analysis of Systems (TACAS), pages 300-314, 1999.

[5] Rajeev Alur, Costas Courcoubetis, and Thomas A. Henzinger. The observational power of clocks. Proceedings of the Fifth International Conference on Concurrency Theory (CONCUR), Lecture Notes in Computer Science 836, Springer-Verlag, 1994, pp. 162-177.

[6] Akio Nakata, Teruo Higashino and Kenichi Taniguchi: "Time-Action Alternating Model for Timed LOTOS and its Symbolic Verification of Bisimulation Equivalence", in Proc. of IFIP TC6/WG6.1 Joint Int'l Conf. on Formal Description Technique for Distributed Systems and Communication Protocols, and Protocol Specification, Testing, and Verification (FORTE/PSTV'96), Kaiserslautern, Germany , Chapman&Hall, pp.279-294, Oct. 1996.

[7] K.G. Larsen and Y. Wang. "Time Abstracted Bisimulation: Implicit Specifications and Decidability", In Proceedings of MFPS'93, 1993.

[8] Akio Nakata, Teruo Higashino and Kenichi Taniguchi: "LOTOS enhancements to specify time constraints among nonadjacent actions using 1st-order logic," In Formal Description Techniques, VI, pp.451-466, Elsevier Science Publishers B.V. (North-Holland), 1994.

[9] J. Sifakis, S. Tripakis, S. Yovine. "Building models of real-time systems from application software", Proceedings of the IEEE, Special issue on modeling and design of embedded, 91(1):100-111, January 2003.

[10] Focardi, R., Gorrieri, R., Lanotte, R., Maggiolo-Schettini, A., Martinelli, F., Tini, S., Tronci, E. "Formal Models of Timing Attacks on Web Privacy.", Electronic Notes in Theoretical Computer Science 62, 2002