Knuth Bendix completion algorithmを用いた マルウェアログ統合解析の高速化

安藤　類央†　　三輪　信介‡

†情報通信研究機構ネットワークセキュリティ研究所
184-8795 東京都小金井市貫井北町４－２－１
ruo@nict.go.jp

‡情報通信研究機構ネットワークセキュリティ研究所／北陸 StarBED 技術センター
923-1211　石川県能美市旭台 2 丁目 12 番地

あらまし 本論文では knuth bendix completion algorithm を用いたマルウェアログの統合と高速化の手法を提案する。マルウェアのトラフィックやメモリダンプなどのログを一階述語論理の形式に変換し、導出を行うことでプロセスベースでの振る舞い解析のためのログ抽出の自動化を可能にする。さらに knuth bendix completion algorithm を用いて推論過程の高速化を行ない、同時に処理系の停止性と合流性を保障する。評価実験ではデータセットで提供された複数種類のログの統合解析の自動化を行ない、計算コストと提案手法の効果を測定評価した。

Faster analysis of malware log using Knuth Bendix completion algorithm

Ruo Ando†　　Shinsuke Miwa‡

†National Institute of Information and Communication Technology
4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795 Japan
ruo@nict.go.jp

‡National Institute of Information and Communication Technology
StarBED Technology Center
2-12, Asahidai, Nomi-city, Ishikawa-pref., 923-1211 JAPAN

Abstract In this paper we propose a faster log processing method for analyzing malware using knuth bendix completion algorithm. Diversified log string of malware is translated into a representation of FoL (First order Logic) formulation and resolved to discover hidden behavior of malware. Besides, we apply reasoning stragety for term rewriting called as knuth bendix completion algorithm for ensuring termination and confluency. Knuth bendix completion includes some inference rules such as lrpo (the lexicographic recursive path ordering) and dynamic demodulation. Proposed system enables us to fasten reasoning process and assure the termination and confluent analysis.

1　はじめに

近年のモニタリング、フィルタリング技術の進展により、ネットワークトラフィックだけで

なく各種リソースアクセスのログが採取可能になった。しかしながら、これらのログを統合し、イベントに関する情報を抽出するための効果的な解析手法はまだ提案されていない。本論文では、単一化、導出 [1]、項書き換え [2]、包摂 [3] などの論理演算を含む自動推論（Mechanized reasoning）を用いて、マルウェアによるソケット、ファイル、メモリなどの多様なアクセスログを統合し、情報抽出、解析の自動化を行う手法 [4][5] [6][7] を提案する。ログ統合時には、支持集合ベース [8] の充足性判定処理を行う。

更に本論文では、Knuth Bendix comletion algorithm[9] を用いて統合解析システムに合流性と停止性を保障し、計算コストを削減する方法を提案する。提案手法では若干の例外が示されたが、超導出を用いて Knuth-Bendix comletion algorithm を適用することで計算コストが最小になることが明らかになった。

2 提案手法

本節では、ログの統合と自動解析に必要な、自動推論の中心的手法である resolution（導出）演算と、問題の定式化について述べる。

2.1 導出法

節 Cls1 と Cls2 がリテラル L1,L2 を持つ場合、導出節 CR は下記によって得られる。$C_R = (C_1\sigma \backslash L_1\sigma) \cup (C_2\sigma L_2\sigma)$ ここで、σ は、リテラル L1 と L2 を等しくする単一化演算子である。σ は、最汎単一化子（most general unifier）の場合もある。$Lit_1 \in Cls_1$ は $Lit_n \in Cls_n$ でも可能であり、複数の節から導出する方法を超導出という。二項導出、超導出の計算コストの実験結果については 5 節で述べる。

2.2 問題の定式化

ログ L からイベント E を発見することを、節集合 S から論理式 P が恒真であることを導出するとする。これは節集合 S に P の否定:P を付加して空節を導くことと同義になる。検体 x のイベントの集合 S(x) と T(x) があり、R(a) が起ったことで S(x) が生じ、P(x) の結果になった場合、$\forall x((S(x) \bigvee T(x))) \rightharpoonup P(x))$ $\forall x((S(x) \bigvee R(x))) \neg R(a)$ 以上 3 つの節に:P(a) を付加して導出を行うと、空集合が得られることになる。これをプログラムの形式で表現すると、

```
#set 1
S(x). T(x). R(a).
#set 2
-S(x) | T(x) -> P(x).
-S(x) | -R(x).
```

となる。その他本提案手法では、項書き換えや包摂処理を行い、ログからイベントと関連情報を抽出する。

3 Knuth-Bendix Completion

Knuth-Bendix 完備化アルゴリズム (Knuth-Bendix Completion Algorithm) は、ある節集合（等式の集合）を完備性を充足する項書き換えシステムに変換するためのアルゴリズムである。変換が成功した場合、項書き換えシステムは、停止性と合流性を持つことが保障される。項書き換えシステムの動作の際は、書き換えが収束すること、複数に分岐することが予想される書き換え結果が、一意の結果になることが重要である。

3.1 危険対

項書き換えシステムが停止する場合（生成節が有限）である場合、危険対の正規形を求めることで合流性を確かめることができる。

- 書き換えシステムの危険対をすべて求める。
- 危険対 (m, n) について正規形 $(m\downarrow, n\downarrow)$ を求める。
- $m\downarrow = q\downarrow$ であるか確認する。
- すべての (p, q) について $m\downarrow = q\downarrow$ であれば項書き換えが一意の結果になると保障される。

実際の推論過程では、書き換えシステムに上記の性質を持っているか事前にチェックすることはなく、処理過程のループにおいて、逐次停止性と合流性が保障されるように処理を行う。

3.2 LRPO

lrpo (lexicographic recursive path ordering) は、項書き換えのうち、交換法則による節生成を制限するもので、Knuth-Bendix Completion を保証する演算規則である。項の集合に対し、アルファベットなどの順序を与えて書き換えの制限を行う。

Lexicographic Ordering 1 Sは $\angle$ の *strict partial ordering* であり、$S(*)$ S の要素の文字列の集合であるとする。このとき、$a(1)..a(n) \prec b(1)..b(n)$ は $a(1) \prec b(1)$ であれば $S(*)$ の *Lexicographic Ordering* である。

Lexcographic recursive path ordering は、項 a, b が高階になった場合に適用可能である。

3.3 dynamic demodulation

dynamic demoulation とは、書き換えられた節を、lrpo であれば再び適用するものである。$x\varphi(y\varphi z) = (x\varphi y)\varphi z$ から、 $(x\varphi y)\varphi z = x\varphi(y\varphi z)$ を導出した際に、前の $x\varphi(y\varphi z) = (x\varphi y)\varphi z$ を、lex order である $(x\varphi y)\varphi z \succ x\varphi(y\varphi z)$ から適用を検討する。

4 適用アルゴリズム

4.1 支持集合戦略

支持集合戦略は、１９６５年に Wos らによって提案されたものである。この計算戦略は制限戦略の１つで、自動推論プログラムに目標とする解空間に関係ないところを探索せずに、対象としている問題に集中させるようにする。節集合 S、T があり、S-T が充足可能であるとき、T は S の支持集合である。このとき、支持集合に属さない節同士では導出を行わず、支持集合に属する節との間で、導出を行う方針を支持集合戦略という。

支持集合 1 H は S の充足可能な支持集合である。このとき、$S = \cup H$ *and* $H \cap T = \phi$ である場合、T も支持集合になる。

4.2 超導出

超導出は１９６５年に Robinson らによって提唱された手法で、通常の導出系の手法では１対の節から順次導出を行うのに対して、２個以上の節に対して導出を行う。超導出の意味は、何段階もの２項導出にあたる作業を１つにまとめたもので、通常の２項導出に比べて、多くの導出が起こるという事を指す。

4.3 包摂

定理証明を用いた推論プロセスでは、目標とする節を導出する過程で、いくつかの節が保持され、新しい節が生成された時点で、過去に保持された節との間で、改めて定理が適用される。この保持されている節のうち、より一般的な節を残す処理を包摂という。

4.4 デモジュレーション

デモジュレーションとは、あらかじめ等価代入を行うための節を定理証明系に加えて、処理節群の簡略化あるいは正準化を行う処理である。本論文ではデモジュレーションを pcap データのポート番号の処理に適用した。

5 評価実験

評価実験では、MARS data set 2010 から検体８種類をランダムに選択し、２種類の推論規則（超導出と２項導出）と、Knuth Bendix completion algorithm を適用した際の効果を測定した。生成節数の測定項目は生成節数合計、書き換え節数、包摂節数、そして保持された節であ

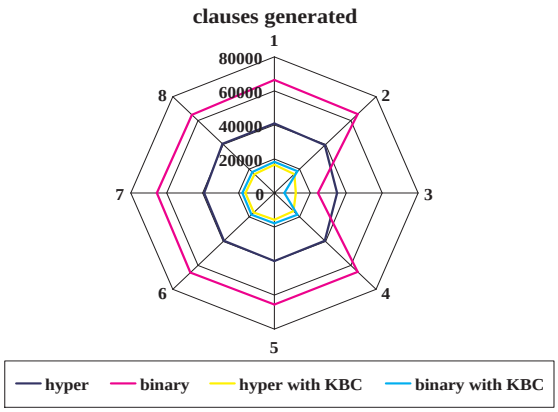

図 1: 超導出、2 項導出、及びＫＢＣを用いた際の生成された節数

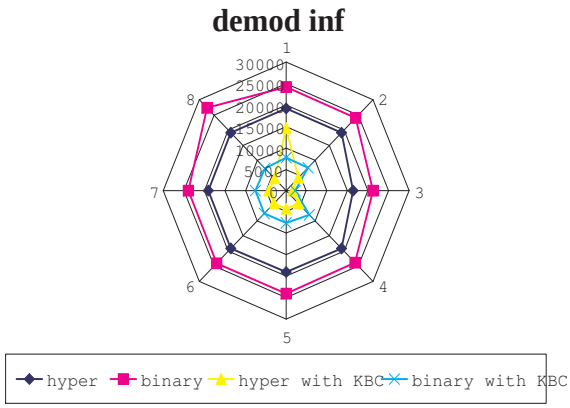

図 2: 超導出、2 項導出、及びＫＢＣを用いた際の生成された変換節（demodulator) の数

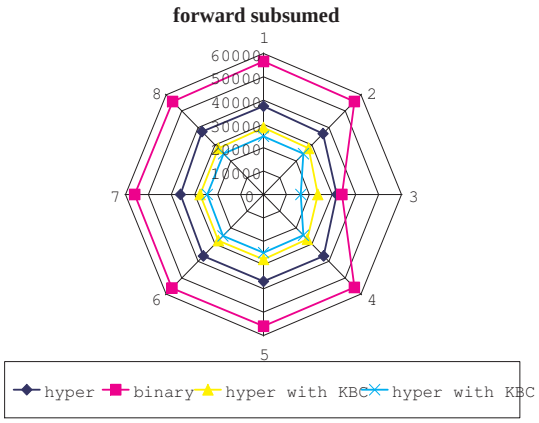

図 3: 超導出、2 項導出、及びＫＢＣを用いた際の包摂 (forward subsumed) された節の数

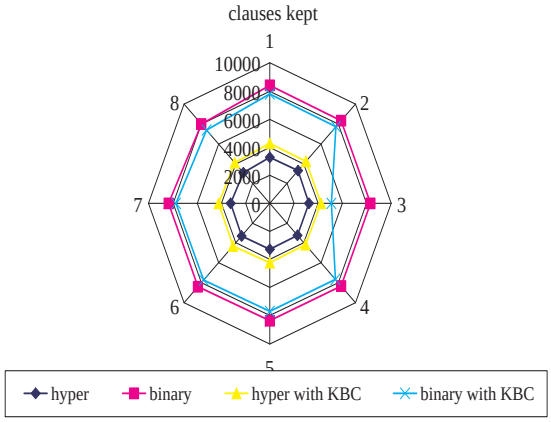

図 4: 超導出、2 項導出、及びＫＢＣを用いた際の保持された節の数

る。概ね生成節数の順序であるが、２項導出（ＫＢＣ適応なし）が一番多く、超導出（ＫＢＣ適用なし)、２項導出（ＫＢＣ適用)、超導出（ＫＢＣ適用）の順で高速な推論（生成節数が少ない）が可能になる。この順序は、包摂節数と保持された節数に関してはすべての検体に当てはまるが、生成節数合計と書き換え節数の場合にいくつかの例外があった。

検体３ (1012F) の生成された節の総数については、KBC 適用の有無に関わらず、２項導出が超導出の計算コストを下回っている。また書き換え節数は、ＫＢＣ適用時の２項導出と超導出の計算コストが同程度の結果になっている。検体１ (09EE) については、ＫＢＣ適用時の超導出の生成節数 (14624) が、大きく２項導出 (7625) を上回っている。上記のような例外があるが、評価実験から概ね推論規則に超導出を用いて、KBC を適用することで最小の計算コストでログを統合解析することができる事が明らかになった。

6　まとめと今後の課題

本論文では knuth bendix completion algorithm を用いたマルウェアログの統合と高速化の手法を提案する。マルウェアのトラフィックやメモリダンプなどのログを一階述語論理の形式に変換し、導出を行うことでプロセスベースでの振る舞い解析のためのログ抽出の自動化を

可能にする。さらに knuth bendix completion algorithm を用いて推論過程の高速化を行ない、同時に処理系の停止性と合流性を保障する。評価実験ではデータセットで提供された複数種類のログの統合解析の自動化を行ない、計算コストと提案手法の効果を測定評価した。

今後の予定としては、unification と term rewriting を用いた手法を難読化ウィルスの解析 [10]、特に悪意のある Java Scritpt の解析 [11] に適用する事などが挙げられる。

参考文献

[1] LarryWos: The Problem of Explaining the Disparate Performance of Hyperresolution and Paramodulation. J. Autom. Reasoning 4(2): 215-217 (1988)

[2] Larry Wos, George A. Robinson, Daniel F. Carson, Leon Shalla, ”The Concept of Demodulation in Theorem Proving ”, Journal of Automated Reasoning, 1967.

[3] Larry Wos: The Problem of Choosing the Type of Subsumption to Use. J. Autom. Reasoning 7(3): 435-438 (1991)

[4] 安藤類央, 門林雄基, 篠田陽一,「Automated deduction system を用いたマルウェア外部観測ログ解析の自動化」情報処理学会コンピュータセキュリティシンポジウム２００９ ２００９年１０月

[5] 安藤類央，門林雄基，三輪信介, 篠田陽一, ”Mechanized reasoning を用いたアクセスログの統合と解析の自動化”, 情報処理学会コンピュータセキュリティシンポジウム２０10 ２０10年１０月

[6] Ruo Ando, ”Automated Log Analysis of Infected Windows OS Using Mechanized Reasoning ”, ICONIP 2009,Neural Information Processing, 16thInternational Conference, ICONIP 2009, Bangkok, Thailand, December 1-5,2009

[7] Ruo Ando, ”Log Analysis of Exploitation in Cloud Computing Environment Using Automated Reasoning”, ICONIP(2) 2010: 337-343, Neural Information Processing, 17th International Conference

[8] Larry Wos, George A. Robinson, Daniel F. Carsonm ”Efficiency and Completeness of the Set of Support Strategy in Theorem Proving ”, Journal of Automated Reasoning, 1965.

[9] D. Knuth and P. Bendix. ”Simple word problems in universal algebras.” Computational Problems in Abstract Algebra (Ed. J. Leech) pages 263-297, 1970.

[10] Ruo Ando, Yoshiyasu Takefuji, ”Faster resolution based metamorphic virus detection using ATP control strategy ”, WSEAS TRANSACTIONS ON INFORMATION SCIENCE AND APPLICATIONS, Issue 2, Volume 3, February 2006. ISSN 1709-0832,pp260-2266,February 2006

[11] Gregory Blanc, Ruo Ando, and Youki Kadobayashi. Term-Rewriting Deobfuscation for Static Client-Side Scripting Malware Detection. In Proceedings of the 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS 2011), February 2011.

検体	推論規則	生成節合計	書き換え節数	包摂節数	保持節
09EE	Hyper	40728	19180	37454	3273
	Binary	66182	24159	56183	8382
	Hyper (KBC)	16546	14624	28175	4283
	Binary(KBC)	18230	7625	24756	7770
0EF2	Hyper	39478	19025	36221	3256
	Binary	65619	23972	55676	8334
	Hyper (KBC)	15947	4193	27522	4199
	Binary(KBC)	17895	7526	24398	7663
1012F	Hyper	37506	16356	31833	3242
	Binary	42347	21265	34031	8314
	Hyper (KBC)	11966	1590	23474	4219
	Binary(KBC)	5513	2272	16175	5064
1C16D	Hyper	40114	18950	36871	3242
	Binary	65345	23880	55432	8311
	Hyper (KBC)	15463	4172	26998	4178
	Binary(KBC)	11794	7943	24268	7643
2AA2	Hyper	40116	19025	36859	3256
	Binary	65720	23984	55760	8352
	Hyper (KBC)	15944	4194	27578	4200
	Binary(KBC)	17989	7533	24489	7676
38E0	Hyper	40083	19119	36815	3267
	Binary	65944	24088	55953	8377
	Hyper (KBC)	16062	4237	27676	4243
	Binary(KBC)	18055	7584	24568	7731
59E9	Hyper	39260	18991	36017	3242
	Binary	65564	23945	55624	8339
	Hyper (KBC)	16059	4217	27590	4223
	Binary(KBC)	18013	7556	24462	7705
79CA	Hyper	40759	18988	37668	3090
	Binary	65163	27285	55685	7952
	Hyper (KBC)	15662	4061	27498	4067
	Binary(KBC)	17308	7242	24295	7391

表 1: MARS data set 2010 の処理時の生成節数、書き換え節数、包摂節数、保持節数。８検体をランダムに選択し、２種類の推論規則とＫＢＣによる効果を測定した。