ClamAVのAC法をダブル配列により拡張したアンチウイルスエンジンの提案

松浦 寛生† 蔵満 琢麻† 望月 久稔†
†大阪教育大学

1. はじめに

コンピュータウイルスによる被害が増加し，対策として様々なアンチウイルスソフトが開発されている．アンチウイルスソフトには，ウイルスパターンを登録して対象のデータを照合するソフトや，ヒューリスティックな学習によりウイルスを検知するソフト，プログラムの動作を監視し，動作の異常を検知するソフトなどが存在する．オープンソースのアンチウイルスソフトClamAntiVirus(以下，ClamAV)は，登録したウイルスパターンと対象ファイルを照合するソフトである[1].

本論文では，ClamAVの照合に用いられるAC法[2]をダブル配列[3]により拡張し，照合手法に合わせてアンチウイルスエンジンを最適化したシステムを提案する．

2. アンチウイルスソフトClamAV

2.1 ウイルスパターンの照合方法

AC法はマシンACと呼ばれる複数パターンマッチングマシンを用いて照合する手法で，ClamAVにおいても使用される．

ClamAVに登録するウイルスパターンには，部分的に不定なパターンを表現するためにワイルドカードを用いる．しかし，不定なパターンはマシンACに登録することができない．そのためClamAVは，ウイルスパターンにおいて不定ではない部分の3バイト以下をパターンキーとしてマシンACに登録し，各節点において検出するキーに該当するウイルスパターンを各節点のOutput集合として登録する．ワイルドカードは，キー検出時において該当するウイルスパターンと対象データを1バイトずつ照合することで処理される．以下，Output集合として登録したウイルスパターンをVパターンと呼ぶ．Vパターンの照合はキー検出時の処理であるため，マシンACにおける対象データの照合ポインタは進まない．よって，Vパターン照合回数の増加によりClamAVにおける照合速度は低下する．

また，ClamAVにおけるマシンACは配列構造によりGoto関数を実現しているため，マシンACにおける各節点にはすべての遷移種に対応した遷移先を保持する空間が必要となる．そのため，すべてのウイルスパターンをマシンACに登録すると，節点数の増加により照合時に必要な空間が膨大になる．そこで，ClamAVはワイルドカードを用いないウイルスパターンをハッシュ法により処理し，マシンACに登録するキー数を制限する．これにより，ClamAVは1つの照合対象データに対してAC法とハッシュ法による2度の照合を要する．

2.2 ウイルスデータベースの構築

ClamAVは，マシンACにおける節点間のリンクや，ハッシュ法における節点間のリンクをポインタ管理により実現する．そのため，ClamAVはシステムを起動する度にファイルからウイルスパターンを読込み，ウイルスデータベースを構築する必要がある．

The Improvement of ClamAV using MachineAC extended with Double-Array Structure
† Nobutaka MATSUURA, Takuma KURAMITSU, Hisatoshi MOCHIZUKI
Osaka Kyoiku University

3. ダブル配列により拡張したAC法

パターンの効率的な照合を実現してシステムの性能を向上させるために，マシンACにおけるGoto関数をダブル配列により実現し，Goto関数に決定性をもたせたマシンACを提案する．

3.1 データ構造

ダブル配列は配列Base，Checkを用いてトライ構造を実現する手法で，節点sから遷移種aによる遷移先節点tを式(1)により定義できる[4]．以下，節点sにおけるBase値，Check値をそれぞれ$B[s]$，$C[s]$とする．

$$\begin{cases} t \leftarrow B[s] + a \\ C[t] = a \end{cases} \quad (1)$$

提案マシンにおけるGoto関数，Output関数を以下のように定義する．ただし，Output関数内で用いる'#'はパターンに使用する遷移種の最大値+1として定義し，Goto関数とOutput関数を拡張するためにダミー節点を付加する．

ダブル配列における遷移は式(1)により定義できるため，Base値を写像することで節点の遷移情報を写像できる．よって，Goto遷移が存在しない節点においてはFailulre遷移先のBase値を写像することで遷移を定義する．ここで，式(1)による遷移先が未定義である場合，Goto関数により遷移すべき節点tをトライ構造上の深さが1，もしくは2である節点に限定し，Goto関数に決定性をもたせることができる．式(1)による遷移先が未定義である場合，節点sが初期節点であるか，遷移すべき節点tのトライ構造上の深さが3以上であるときにGoto関数を拡張するためのダミー節点を作成する．ダミー節点の節点番号dを$B[s]+a$とし，$B[d]$に遷移すべき節点tのBase値を写像する．ここで，ダミー節点dは節点tの遷移情報をもつため，節点sから節点tへの遷移を擬似的に定義できる．そこで，Goto関数を以下のように拡張することで，提案マシンはFailure関数を用いることなく対象データを照合することができる．

関数 Goto(節点 s，遷移種 a) 戻り値：遷移先節点
式(1)に示す節点tが定義されていればtを返す．未定義であれば$t \leftarrow B[B[$初期節点$]]+C[s]]+a$とし，$C[t]=a$であればtを返し，そうでなければ$B[$初期節点$]+a$を返す．

関数 Output(節点 s) 戻り値：Output集合のインデクス
ダミー節点$s_o \leftarrow s$+'#'とし，$C[s_o]$='#'であればOutput集合のインデクス$B[s_o]$を返し，そうでなければOutput集合が存在しないことを示すFalseを返す．

例1: 図1に，パターン集合P={"ACBAC", "BA", "BAB"}を登録した提案マシンを示す．ここで，遷移種'A', 'B', 'C', '#'の内部表現値をそれぞれ0, 1, 2, 3とし，Output集合{"ACBAC"}, {"BA"}, {"BAB"}のインデクスをそれぞれ1, 2, 3とする．また，図1のマシンACにおいて，節点1を初期節点とし，Base値，Check値がともに空欄である節点を未使用節点とする．

図1における節点4，14はGoto関数を拡張するためのダミー節点に該当し，それぞれ初期節点1，節点13における遷移情報

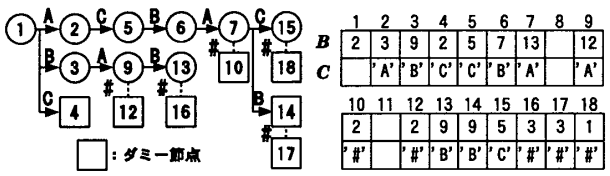

図1 パターン集合Pを登録した提案マシン

をもつ．これにより節点1から節点1，節点7から節点13へのGoto関数による遷移を擬似的に定義できる．同様に，節点10，12，16，17，18はOutput関数を拡張するためのダミー節点に該当し，それぞれ節点7，9，13，14，15におけるOutput集合のインデクスを管理する．これによりOutput集合のインデクスを，ダミー節点におけるBase値を参照することで得られる．

図1のマシンACを用いて，対象データ“ACBABC”を照合する場合について考える．まず初期節点1において，Goto(1, ‘A’)=2より，節点2へ遷移する．ここで，Output(2)=Falseより，節点2において検出するOutput集合は存在しない．同様に，節点5, 6, 7と遷移し，$C[7+'\#'(10)]='\#'$より，$B[10]=2$に該当するOutput集合{“BA”}を検出する．次に，Goto(7, ‘B’)=14より節点14へ遷移し，Output(14)=3に該当するOutput集合{“BAB”}を検出する．従来のマシンACではダミー節点14が存在しないため，節点7からFailure関数により節点9に遷移する必要がある．その後，Goto(14, ‘C’)=4より，節点4へ遷移し，対象データの末尾まで照合したので終了する．

4． AC法の拡張に伴ったアンチウイルスエンジンの最適化

4.1 AC法のみによる実装

提案マシンは，1節点あたりに必要な空間がBase値，Check値を保持する空間のみとなるため，ClamAVのマシンACよりも照合時に必要な空間が小さい．そこで，すべてのウイルスパターンのキーを提案マシンに登録し，AC法のみにより照合するアンチウイルスエンジンを提案する．また，提案システムに登録するウイルスパターンのキー長を伸ばすことでOutput集合を保持する節点を分散させ，Vパターン照合回数を抑制する．これにより提案システムは1度の照合でウイルスを検知でき，Vパターン照合回数を抑制できるため高速である．しかし，キー長を伸ばすことにより照合時に必要な空間が増加するため，システムを使う環境に合わせてキー長を決定する必要がある．

4.2 ウイルスデータベースの復元

ダブル配列はカーソル管理によりデータ構造を実現する．そのため，1度作成したウイルスデータベースをファイルに保存することで，データベースをファイルから直接コア上に復元できる．これによりデータベースの構築時間を短縮できる．

5． 実験評価

提案システムの有効性を示すため，ClamAVとの比較実験をIntel Pentium Dual 1.60GHz, Fedora 7上で行った．実験では，2007年12日26日時点のウイルスパターン[1]をシステムに登録し，ファイル数22，総バイト数9.38MBのEXEファイルを照合した．実験における，システム全体の稼働時間，ウイルスパターンの照合時間，ウイルスデータベースの読込み時間，ウイルスパターンの照合時に必要な空間，マシンACの節点数，Vパターンの照合回数を表1に示す．また，実験において，提案システムに登録するウイルスパターンのキー長を5とした．

表1 実験結果

	ClamAV	提案システム
稼働時間 (s)	46.07	35.05
照合時間 (s)	43.09	33.06
読込み時間 (s)	2.81	1.89
使用空間 (MB)	26.27	25.76
マシンACの節点数	9,764	876,976
Vパターン照合回数	601,457,898	509,169,419

表1に示すように提案システムはマシンACのOutput関数により発生するVパターンの照合回数がClamAVよりも少ない．これは，マシンACに登録するウイルスパターンのキー長を伸ばしたことにより，Output集合をもつ節点が分散し，各節点において発生する照合回数を抑制できるためである．さらに，ClamAVがハッシュ法とAC法の併用により，1つの照合対象データに2度の照合を要するのに対し，提案システムはAC法による1度の照合しか行わないため，表1に示すように，ClamAVに対し約77%の時間で照合可能となる．

また表1より，提案システムにおけるウイルスデータベースの読込み時間はClamAVの約67%であり，提案システムはデータベースの読込みにおいてもClamAVより高速である．これは，提案システムにおけるマシンACをカーソル管理により実現したことで，提案システムが構築済みのデータベースをファイルから直接読込めるためである．

提案システムはAC法を拡張して実装したことで，表1に示すようにClamAVよりマシンACの節点数が増加する．しかし，提案システムのマシンACにおける1節点あたりに必要な空間はClamAVよりもはるかに小さいため，キー長5における提案システムの照合時に必要な空間は，キー長3におけるClamAVとほぼ同等であった．

以上より，提案システムはClamAVと比較して，照合時に必要な空間を増加させることなくシステム全体の稼働時間を短縮できることを示した．

6． おわりに

本論文では，ClamAVのAC法をダブル配列により拡張したアンチウイルスエンジンを提案し，実験によりClamAVのエンジンよりもシステム全体の稼働時間を短縮できることを示した．

今後の課題として，システムにヒューリスティックな検出機能を追加し，未知のウイルスにも対応したエンジンに拡張することがあげられる．

参考文献

[1] Clam antivirus. `http://www.clamav.org/`.

[2] M.J.Corasick A.V.Aho. Efficent string matching: An aid to bibliographic search. *Communications of the ACM*, Vol. 18, No. 6, pp. 333–340, 1975.

[3] J.Aoe. An efficient digital search algorithm by using a double-array structure. *IEEE Trans, on Software Engineering*, Vol. 15, No. 9, pp. 1066–1077, 1989.

[4] 矢田晋, 森田和宏, 泓田正雄, 平石亘, 青江順一. ダブル配列におけるキャッシュの効率化. *FIT2006*, pp. 71–72, 2006.