

状態監視用エキスパートシステムにおける履歴情報の分割管理法の提案

6P-1

橋本 和夫

松本 一則

小花 貞夫

KDD 研究所

1. はじめに

時間的に変化するシステム状態の推定を行なうため、すべての時間情報を知識ベースに蓄積する、TMM^[1] や timegraph-II^[2] などの時間情報の管理手法が提案されている。これらの手法は、各状態相互の時間的な制約充足を検査するため計算量が大きく、扱うことのできる情報量には限界がある^[3]。従って、連続的な状態監視が必要となる、プロセスやネットワークの監視などのエキスパートシステムに適用することは難しい。

そこで、筆者らは、最新のシステム状態のみを管理し、過去の状態については、必要に応じて履歴情報をもとに復元する方式^[4]を検討している。本稿では、履歴情報からの状態復元時間を短縮する履歴情報の分割管理法を提案する。

2. 履歴管理に基づく時間推論システムの構成

本稿では、図1の構成を取ることを前提とする。

- (1) 監視系は、状態差分を検出した時に、状態管理部にイベントを通知する。このため、 $\langle report(\phi), t \rangle$ を時刻 t で、状態 $\neg\phi$ から状態 ϕ への状態遷移が通知されたことを示す述語とすると、通知の論理学的意味は (1) 式のように定義できる。

$$\forall t_1, \phi [\langle report(\phi), t_1 \rangle \equiv \exists t_2 \forall t [(t_2 < t < t_1) \wedge \langle \neg\phi, t \rangle] \wedge \exists t_3 \forall t' [(t_1 < t' < t_3) \wedge \langle \phi, t' \rangle]] \quad (1)$$

- (2) 履歴管理部は、イベント通知を履歴として蓄積する。
- (3) 問題解決器は、質問応答のためのインタフェースであり、状態管理部が復元するシステム状態に基づいて、診断などの推論を行なう。
- (4) 状態管理部は、監視系からのイベント通知に基づき、最新のシステム状態を更新すると同時に、イベントを履歴管理部に通知する。問題解決器の要求に応じて履歴情報の検索を行ない、任意の過去の時刻のシステム状態を復元する。

3. 履歴情報からのシステム状態復元方法

STRIPS^[5] は、監視対象を初期状態から目的状態へ導くための動作のシーケンスを求める定理証明器であるが、ここでは、履歴情報から任意の(過去の)時刻の

“A proposal of Log Partitioning Method for State Monitoring Expert Systems”, Kazuo Hashimoto, Kazunori Matsumoto and Sadao Obana: Kokusai Denshin Denwa Co. Ltd.

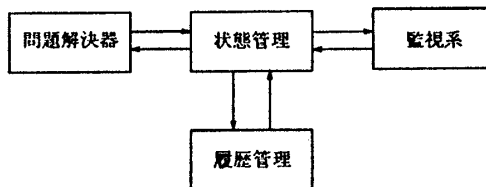


図1: 履歴管理に基づく時間推論システムの構成

システム状態を復元するために用いる。以下では、状態復元アルゴリズムを示し、履歴情報を主記憶上に置く場合とファイルに置く場合の復元時間を、示す。

3.1 状態復元アルゴリズム

ある時刻 t_n におけるシステム状態を Δ_n とし、 $E_n = \{ \langle e_i, t_i \rangle \mid i = 1, 2, 3, \dots, n \}$ を、イベント生起時間順序によって整列されたスタックとする。最新時刻 t_n に於けるイベント $\langle e_n, t_n \rangle$ をスタックの起点として、スタック距離を定める。例えば、時刻 t_k ($t_1 \leq t_k \leq t_n$) に於けるイベント $\langle e_k, t_k \rangle$ までのスタック距離は $(n - k)$ とする。このとき、スタック E_n を逆順にたどり過去の時刻 t_k のシステム状態 Δ_k を復元する方法は以下の通り。

- (1) (i) $S \leftarrow \Delta_n$
(ii) $E \leftarrow E_n$
- (2) もし $E = \{ \}$ ならば失敗して停止する
- (3) (i) $e \leftarrow E$ 中の最新イベント
(ii) $t_e \leftarrow e$ の生起時刻
- (4) もし $t_e \leq t_k$ ならば
(i) S を求めるシステム状態 Δ_k として、停止する
otherwise (もし $t_e > t_k$ ならば)
(i) $L_{DEL} \leftarrow e$ の削除リストに指定される状態
(ii) $L_{ADD} \leftarrow e$ の追加リストに指定される状態
(iii) $S \leftarrow (S \cup L_{DEL} - L_{ADD})$
(iv) $E \leftarrow (E - \langle e, t_e \rangle)$
- (5) (2)へ戻る

3.2 主記憶上に履歴情報を置く場合の復元時間

3.1 のアルゴリズムを、OPS 系の推論エンジンで実行する場合を想定し、スタック距離 k のシステム状態 Δ_k を復元する処理時間を $T_{主記憶}(k)$ とするとき、状態を表す Working Memory (WM) 要素の削除と書き込みの操作時間が支配的になると考えられる。そこで、

T_{del}	WM 要素の平均削除時間
T_{add}	WM 要素の平均書き込み時間
N_{del}	—イベント当りの WM 要素の平均削除個数
N_{add}	—イベント当りの WM 要素の平均書き込み個数

として、 $T_{主記憶}(k)$ を (2) で近似する。

$$T_{主記憶}(k) = k \times T_{\alpha} \quad (2)$$

$$T_{\alpha} = (T_{del} \times N_{del} + T_{add} \times N_{add})$$

3.3 ファイルに履歴情報を置く場合の復元時間

N	イベント履歴のスタック長
E_{kN}	$\{ \langle e_i, t_i \rangle \mid (k-1)N+1 \leq i \leq kN \}$ からなるイベント履歴
Δ_{kN}	時刻 t_{kN} における基本的状態の集合
$ \Delta_{kN} $	Δ_{kN} を構成する基本的状態の個数

とする時、履歴 LOG_k は、時刻 $t_{(k-1)N+1}$ から t_{kN} までのイベント履歴と、最終イベント $\langle e_{kN}, t_{kN} \rangle$ によって生じた状態 Δ_{kN} から (3) 式で定義する。

$$LOG_k \equiv \Delta_{kN} \cup E_{kN} \quad (3)$$

この LOG_k を状態管理部がロードする時に要する時間 T_{load} を、(4) 式で近似する。最新状態 Δ_n からスタック距離 $(kN+l)$ の状態 $\Delta_{n-(kN+l)}$ を、 LOG_k を用いる場合の復元時間は、ファイルからの書き込み時間が支配的になるため、 LOG_k の検索時間を無視し、 $|\Delta_{kN}| = |\Delta_0|$ ($k \geq 0$) と仮定して、時間 $T_{file}(kN+l)$ を (5) で近似する。

$$T_{load} = (|\Delta_0| + N) \times T_{add} \quad (4)$$

$$T_{file}(kN+l) = T_{load} + T_{主記憶}(l) \quad (5)$$

4. 履歴情報の分割管理法の提案

主記憶上の履歴からの復元時間 (2) は、スタック距離 k に比例して増加するが、スタック長 N のファイルからの復元時間 (5) は、一定時間内に留まる。このため、あるスタック長 N_{max} 以上では、ファイルによって履歴管理を行なう方が、復元時間を短縮できる。従って、 N_{max} 個までのイベント履歴を主記憶上のスタックに持ち、スタック距離 k が N_{max} 以上のイベントに対しては、 N_{max} 個のイベント履歴 $E_{kN_{max}}$ と $\Delta_{kN_{max}}$ から構成するファイル LOG_k を用いる、履歴情報を分割管理法が有利となる。

そこで、(2) の k を $kN+l$ と置き変えて、(5) との等式を作り、 $k=1$ と置くと、 N_{max} が (6) 式によって求められることができる。また、この時の復元時間 $T(kN_{max}+l)$ は、 $0 \leq l \leq (N_{max}-1)$ に対して、(7) 式で与えられる。

$$N_{max} = \frac{|\Delta_0| \times T_{add}}{(T_{del} \times N_{del} + T_{add} \times (N_{add} - 1))} \quad (6)$$

$$T(kN_{max}+l) = \begin{cases} T_{主記憶}(l) & k=0 \\ T_{load} + T_{主記憶}(l) & k \geq 1 \end{cases} \quad (7)$$

5. 平均復元時間改善の効果

スタック距離 k のシステム状態 Δ_{n-k} に対する検索要求頻度 $P(k)$ は、 k に応じて指数関数的に減少すると仮定し、(7) 式で復元時間が与えられるときの、平均復元時間 MT を (9) で定義する。また比較のため、仮想的に主記憶上にすべて履歴を持つ場合の平均復元時間 $MT_{主記憶 \infty}$ を (10) に示す。

$$P(k) = P_0 e^{-k\lambda} \quad P_0 = (1 - e^{-\lambda}) \quad (8)$$

$$MT = \sum_{k=0}^{\infty} T(k) \times P(k)$$

$$= (|\Delta_0| + N_{max}) \frac{1}{e^{\lambda N_{max}} - 1}$$

$$- T_{\alpha} P_0 \frac{dg}{d\lambda} \frac{1}{1 - e^{-\lambda N_{max}}} \quad (9)$$

$$g(\lambda) \equiv \sum_{l=0}^{N_{max}-1} e^{-\lambda l} = \frac{1 - e^{-\lambda N_{max}}}{1 - e^{-\lambda}}$$

$$MT_{主記憶 \infty} = \sum_{k=0}^{\infty} T_{主記憶}(k) \times P(k)$$

$$= \frac{T_{\alpha}}{e^{\lambda} - 1} \quad (10)$$

$$V = \frac{MT - MT_{主記憶 \infty}}{MT_{主記憶 \infty}} \quad (11)$$

4. の履歴情報の分割によって得られる平均復元時間に関する改善度を (11) で表すと、実際の値を (12) とした場合、(13) (14) (15) より 75.8% の改善が得られる。

$$\left(\begin{array}{ccc} \lambda = 10^{-3} & N_{max} = 10^2 [\text{個}] & T_{\alpha} = 10^{-3} [\text{秒}] \\ P_0 = 10^{-3} & |\Delta_0| = 10^3 [\text{個}] & T_{add} = 10^{-3} [\text{秒}] \end{array} \right) \quad (12)$$

$$MT = 1.09 \times 10^{-4} [\text{秒}] \quad (13)$$

$$MT_{主記憶 \infty} = 4.5 \times 10^{-4} [\text{秒}] \quad (14)$$

$$V = -0.758 \quad (15)$$

6. まとめ

ここでは、履歴を用いて過去の状態を復元するシステムにおける履歴情報の分割管理法を提案し、同手法による状態復元時間改善の効果を示した。最後に、日頃ご指導いただき、KDD 研究所浦野所長、眞家次長に感謝いたします。

参考文献

- [1] Dean, T., McDermott, D.: "Temporal data base management" Artificial intelligence Vol.32, pp 1-55, 1987.
- [2] Gerevini, A., Schubert, K.: "Efficient temporal reasoning through timegraphs" IJCAI, 1993
- [3] Yampratoom, E., Allen, J.F.: "Performance of Temporal Reasoning Systems" SIG ART Bulletin Vol.4, No.3, pp 26-29
- [4] 橋本、松本、小花: "状態監視用エキスパートシステムのための履歴管理方式の提案" 電子情報通信学会 1994 年春季全国大会, D-226, 6-229.
- [5] Fikes, R.E., Nilson, N.J.: "STRIPS: A new approach to the application of theorem proving to problem solving" Artificial intelligence Vol.2, pp 189-208, 1971.