B-038

効率のよいコース探査のための抽象化
Abstraction for Effective Course Exploration

大矢野 潤[†]
Jun Oyano

1. はじめに

WWW の普及にともない，インターネット上で多様なサービスが提供されている．しかし，複数のサービスが相互に依存している場合には，全文検索やリンク集に基づいた検索システムでは不十分である．

例えば，簿記の e-Learning 教材において，出席や課題，質問の提出に電子メールが用いられる場合には，学習者がまず履修する必要があるのは「簿記」ではなく電子メールの利用方法を含む適当な「情報リテラシ」のコースであることがわかる．このように，それぞれいくつかの前提条件を持つ教材を複数組み合せて利用する場合には，学習者の現在の状態から最終的な目標を達成するために修得可能な教材の履修順序列を検索の対象とする仕組みが必要である．

筆者は，状態が動的に変化するプロセスとデータベースの更新を同一視し，モデル検査の技術 [1] を応用することにより，時制論理式を用いたアクション系列を検索の対象とする枠組を提案してきた [3]．しかし，このアプローチには状態数爆発や，検索結果として得られる具体的な状態遷移列は大部分が冗長なものになるという問題点が指摘される．

本論文では模倣前順序関係 (Simulation Preorder Relation) を用いて構成された抽象領域に対してモデル検査を行い，そこで得られた反例と抽象化時に得られる証拠から具体的な系列を得るための枠組を提案する．この仕組みにより，モデル検査技法を時制データベース検索に応用する際の効率化を計る．

2. 時制論理式を用いた問い合わせ

2.1 タプル論理と遷移システム

通常，データベースシステムは，問い合わせを発行した時刻でデータベースの状態を固定し，その状態における問い合わせ式の充足集合を検索結果として返す．本論文では，この問い合わせ式を基本論理式と呼ぶことにし，タプル論理式を仮定する．タプル論理はタプル T とその属性 A から作られる $T.A$ を基本構成子とした，次のような原子式 B をもつ．

$$B ::= (T.A = T.A) \,|\, (T.A = C) \,|\, R(T)$$

ここで，$T.A = T.A$，$T.A = C$ は属性値間の，もしくは属性と定数との比較を，$R(T)$ は T がある関係 R の要素であることを示す．

一方，時制論理のモデルとしてラベルつき遷移システム (LTS) を用いる．LTL M は状態の集合 S，状態間の遷移関係 $\langle s, i, \tilde{s}\rangle \in \xrightarrow{i}$ $(s, \tilde{s} \in S, i \subseteq \mathrm{AP})$，状態とその状態で成立している命題の集合との対応を行うラベルづけ $\mathcal{L} : S \to 2^{\mathrm{AP}}$ からなる．ただし，命題の集合 AP をタプル論理の基本論理式の集合とする．

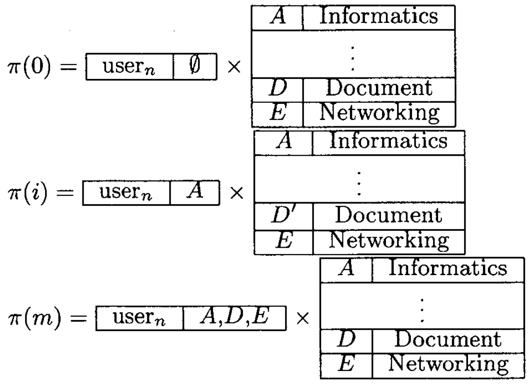

図 1: 状態列 π のテーブル表現

ここで，タプル論理のタプル T と LTS の状態 $s \in S$ を同一視することを考える．例えば，タプル T で表される学生は属性 A として英語の成績をもち，その値が 80 点であったとすると，$T.A = 80$ と記述できる．一方 LTS では，状態 s がそのラベルとして命題 $A = 80$ を持つことは $A = 80 \in \mathcal{L}(s)$ と記述する．さらにタプルの JOIN（×）は合成された遷移システム（後述）の状態の一つとして取り扱う．

例えば，あるユーザ user$_n$ の初期状態 $\pi(0)$ には履修履歴がなく（$\emptyset$），時刻 j の状態 $\pi(j)$ では，科目 A が履修済み，科目 D が履修中（D'），さらに $\pi(m)$ で A, D, E が履修済みであるという状態変化の系列 π は図 1 のように表現することができる．ただし，状態遷移列（パス）π において，j 番目の状態を $\pi(j)$ と記述するものとする．

2.2 時制論理とモデル検査

時制論理は事象の一般的な順序関係を記述することができるため，並行システムの振舞いやそのプロトコルの時間的性質の記述に多く用いられる．

ここでは説明のために CTL*と呼ばれる時制論理の一部を導入する．$M, s \models \phi$ を “モデル M はその状態 s で論理式 ϕ を充足する” と解釈すると，CTL*論理式の意味は次のように定義される．なお，$\mathbf{A}$，$\mathbf{G}$ はそれぞれ $\mathbf{A}\phi \equiv \neg\mathbf{E}\neg\phi$，$\mathbf{G}\phi \equiv \neg\mathbf{F}\neg\phi$ とする．

$$\begin{array}{rcl}
M, s \models p & \Leftrightarrow & p \in \mathcal{L}(s) \\
M, s \models \neg\phi & \Leftrightarrow & M, s \not\models \phi \\
M, s \models \mathbf{E}\phi & \Leftrightarrow & M, \pi \models \phi \text{ となる } \pi(\pi(0) = s) \text{ が存在} \\
M, \pi \models \neg\phi & \Leftrightarrow & M, \pi \not\models \phi \\
M, \pi \models \mathbf{F}\phi & \Leftrightarrow & \text{ある } k \geq 0 \text{ が存在して } M, \pi(k) \models \phi
\end{array}$$

時制論理によって記述された性質を対象となるシステムが充足しているかを検証する枠組としてモデル検査

[†]千葉商科大学 政策情報学部

(Model Checking) [1] がある．モデル検査機は，検査対象としてのシステムのモデルとそのシステムが保持することが期待される性質を表す論理式をその入力とし，肯定的な結果が得られた場合には "yes" を，そうでなければ論理式を充足しない状態列を反例として出力する．

ここで，モデル検査機にシステム M と "どの選択肢をとっても常に $\neg\phi$ が成り立つ" という意味の論理式 $\mathbf{AG}\neg\phi$ を入力として与え，出力として反例となるパス π が得られたとすると，π はモデル M においての論理式 $\neg(\mathbf{G}\phi)$ を充足する具体例となる．意味論の議論から，$M,\pi \models \neg(\mathbf{G}\neg\phi) \Leftrightarrow M,\pi \models \mathbf{F}\phi$ となり，この π について $\pi(0) \models \mathbf{EF}\phi$ であることがわかる．$\pi(0)$ をモデル M の初期状態に限定すると，$\mathbf{AG}\neg\phi$ は，"初期状態から開始して，いつか ϕ が成り立つまでのパス π" を検索するための問い合わせ式となる．

以下，本論文では論理を $\mathbf{AG}\neg\phi$，かつ ϕ を基本論理式に限定する．これにより，モデル検査に必要な計算量をモデルの状態数に対して線形とすることができる [1]．

3. 抽象領域上の問い合わせ

ここでは，複数の LTS を組み合わせた場合に起こる状態爆発を防ぐために，模倣前順序関係を用いた抽象化技法を用いる．ここではモデルを Moore 機械に限定した同期遷移システムを用いる．目標はユーザとコースの集合を合成したシステム User‖Courses の抽象領域である User′‖Courses′ に対してモデル検査を行い，ここで得られた反例から，User の動作系列を得ることである．以下で用いる定義，記法はおもに [2] から得ている．

3.1 Moore 機械

Moore 機械 $M \equiv \langle S, s_0, I, O, L, R\rangle$ とは次の要素から作られる．S:状態の集合，$s_0 \in S$:初期状態，I:入力命題の集合，O:出力命題の集合（ただし $I \cap O = \emptyset$），$L : S \rightarrow 2^O$ は状態に出力命題の集合でラベルづけする関数，$R \subseteq S \times 2^I \times S$ を遷移関係とする‡．

次に Moore 機械 $P \equiv \langle S^P, s_0^P, I^P, O^P, L^P, R^P\rangle$ と $Q \equiv \langle S^Q, s_0^Q, I^Q, O^Q, L^Q, R^Q\rangle$ の合成 $P\|Q$ を定義する．$O^P \cap O^Q = \emptyset$ のとき P と Q が合成可能であるといい，このとき $P\|Q \equiv \langle S, s_0, I, O, L, R\rangle$ を次のように定義する．$S = S^P \times S^Q$，$s_0 = \langle s_0^P, s_0^Q\rangle$，$I = (I^P \cup I^Q)\backslash(O^P \cup O^Q)$，$O = O^P \cup O^Q$．さらに，すべての $\langle p,q\rangle, \langle\tilde{p},\tilde{q}\rangle \in S$，$i \subseteq I$ について，$L(\langle p,q\rangle) = L^P(p) \cup L^Q(q)$，$R(\langle p,q\rangle, i, \langle\tilde{p},\tilde{q}\rangle)$ を $R^P(p,(i \cup L^Q(q)) \cap I^P, \tilde{p})$ かつ $R^Q(q,(i \cup L^P(p)) \cap I^Q, \tilde{q})$ と定義する．

最後に Moore 機械上の全順序模倣関係を定義する．Moore 機械 P，Q と二項関係 $\Theta_{P\preceq_S Q} \subseteq S^P \times S^Q$ が存在し，次の 3 つの条件を満たす時，Q は P を模倣，もしくは抽象化するといい，$P \preceq_S Q$ と記述する．

i）$(s_0^P, s_0^Q) \in \Theta_{P\preceq_S Q}$，ii）すべての $(p,q) \in \Theta_{P\preceq_S Q}$ について $L^P(p) \cap O^Q = L^Q(q)$，iii）すべての $(p,q) \in \Theta_{P\preceq_S Q}$ とすべての $R^P(p,i,\tilde{p})$ となる $i \subseteq I$，$\tilde{p} \in S^P$ に関して，$R^Q(q,(i \cup L^P(p)) \cap I^Q, \tilde{q})$，$(\tilde{p},\tilde{q}) \in \Theta_{P\preceq_S Q}$ なる $\tilde{s} \in S^Q$ が存在する．また，$\Theta_{P\preceq_S Q}$ を特に証拠 (witness) と呼ぶ．

‡[2] では，さらに Nonblocking，Finite Nondeterminism の制限があるが本論文では省略する．

3.2 抽象領域の構成と詳細化

$P \preceq_S Q$ と P 上のパスの集合が Q 上のパスの集合に含まれることは同値であり [2]，$Q, s_0^Q \models \mathbf{A}\phi$ から $P, s_0^P \models \mathbf{A}\phi$ を帰結して良い．しかし，Q のパスは必ずしも P のパスではないため，一般に，$Q, s_0^Q \not\models \mathbf{A}\phi$ から $P, s_0^P \not\models \mathbf{A}\phi$ を導くことはできない．このため，モデル検査で得られた反例 π^Q を抽象化の段階で得られた証拠 $\Theta_{P\preceq_S Q}$ と抽象化のルールに基づき P のパスに対応づける．この対応づけが失敗した場合には，π^Q を棄却する．逆に対応づけが成功した場合には詳細化されたパス π^P を新たな反例として採用する．

具体的な詳細化の方法について述べる．

1）抽象領域が，$P \preceq_S Q$ の証拠 $\Theta_{P\preceq_S Q}$ から導き出されている場合には $\Theta_{P\preceq_S Q}$ を利用して Q のパス π^Q に対応する P のパスを検索する．

2）$P \preceq_S P'$，$Q \preceq_S Q'$ の事実から $P\|Q \preceq_S P'\|Q'$ が導出され，反例 π が得られたとする．この時，合成の定義より π に対して，P' のパス $\pi^{P'}$ がそれぞれ存在する．ここで $\pi(j) = \langle\pi^{P'}(j), \pi^{Q'}(j)\rangle$ とすると $\pi = \pi(0) \xrightarrow{i(0)} \ldots \pi(j) \xrightarrow{i(j)} \pi(j+1) \ldots$ のそれぞれの $i(j)$ に対して $i^{P'}(j) = (i(j) \cup L^{Q'}(\pi^{Q'}(j))) \cap I^{P'}$，$i^{Q'}(j) = (i(j) \cup L^{P'}(\pi^{P'}(j))) \cap I^{Q'}$，$\langle\pi^{P'}(j), \pi^{Q'}(j)\rangle$ が存在する．この $i^{P'}(j)$ と $\pi^{P'}(j)$ から構成されるパス $\pi^{P'}$ に対応する P のパスの存在を証拠 $\Theta_{P\preceq P'}$ を用いて調査する．$Q \preceq_S Q'$ についても同様の手続きを行う．

3）$P\|Q' \preceq_S P'$ と $P'\|Q \preceq_S Q'$ から $P\|Q \preceq_S P'\|Q'$ が導出された場合 [2] には 2）と同様に $P'\|Q'$ のパス π から P' に対応するパス $\pi^{P'}$ が導かれる．この $\pi^{P'}$ が $P\|Q'$ のパスになっていることを調べる．このとき，すでに $P'\|Q'$ のパスから Q' に対応するパスの存在はわかっているため，P に対応するパスの存在性のみ調べれば良い．$P'\|Q \preceq_S Q'$ の場合も同様に行う．

4. まとめ

本論文ではモデル検査の枠組を，データベース問い合わせに時制論理式を応用するための提案を行った．この際，計算量と探査するモデルの状態数をおさえるための時制論理式を制限し，さらに模倣前順序を用いたモデルの抽象化とパスの詳細化を行う方法を示した．

ここでの詳細化手続きは，それぞれのモデルの持つ証拠を再利用するため，分散されたコースをそれぞれ階層的に調べることができる．このことは，XML などそれぞれ独立したサイトで提供される文書から構成される分散データベースやコースの検索に有効である．

参考文献

[1] E.M. Clarke, O. Grumberg and D.A. Peled, Model Checking, The MIT Press, 1999.

[2] T. A. Henzinger, S. Qadeer, S. K. Rajamani and S. Tasiran, An Assume-Guarantee Rule For Checking Simulation, ACM Transactions on Programming Languages and Systems, Vol. 8, No. 1 pp. 111-123, Jan 1999 Languages pp. 12-25, ACM Press, 2000.

[3] 大矢野 潤，分散環境にある文書からのコース抽出と時制を用いた問い合わせについて，第 2 回情報科学技術フォーラム（FIT2003）D-016 第 2 分冊 pp.33-34，Sept 2003.