Tạp chí Tin học và Điều khiển học, T. 14, S. 2, (1998), (53- 60)

LÝ THUYẾT ĐIỂM BẤT ĐỘNG TRONG CƠ SỞ DỮ LIỆU SUY DẪN

PTS. LÊ MẠNH THẠCH [1]

Tóm tắt nội dung.

Bài báo dựa trên cơ sở lý thuyết điểm bất động của một chuyển dịch trên một dàn đầy đủ để xác lập ngữ nghĩa thích hợp cho chương trình Datalog và các trường hợp mở rộng của nó: trong trường hợp Datalog ngữ nghĩa là mô hình nhỏ nhất (minimal model), trong trường hợp Datalog mở rộng có phủ định xếp tầng thì ngữ nghĩa được xác lập là một mô hình hoàn thiện (perfect model), còn trong trường hợp phủ định tổng quát thì là mô hình lạm phát (inflationary model).

I. MỘT SỐ KHÁI NIỆM CƠ BẢN.

Trong khuôn khổ bài báo tôi xin phép không nhắc lại ở đây một số khái niệm trong ngôn ngữ lôgic bậc nhất như term, atom, literal, fact,... (đọc giả có thể xem ở [C1]).

Chương trình logic: Chương trình logic là tập hợp hữu hạn, khác rỗng P các công thức bậc nhất dạng: $s_1 \wedge s_2 ... \wedge s_k \rightarrow p$ ($k \geq 0$), trong đó p là một atom, s_i ($i = 1, ...k$) là các literal, $\wedge$ ký hiệu ghép hội logic, $\rightarrow$ ký hiệu phép kéo hteo logic.

Thay cho công thức trên, chúng ta sử dụng cách viết như trong Datalog và Prolog, $p{:-}\ s_1 \wedge s_2 ... \wedge s_k$.

- Nếu $k > 0$ thì công thức được gọi là một quy tắc (rule).

- Nếu $k = 0$ thì công thức trở thành p: -, gọ là một nhân tử (fact).

Phép thể hiện Herbrand: Cho chương trình logic P chúng ta ký hiệu:

- L_P là ngôn ngữ bao gồm tất cả các ký hiệu tham gia trong P và giả thiết tập ký hiệu hằng trong đó là hữu hạn và khác rộng.

- HB là tập hợp các nguyên tố cơ sở của L_P và gọi là cơ sở Herbrand của P.

Một tập con I của HB là một phép thể hiện Herbrand của chương trình P theo nghĩa một nguyên tố cơ sở p có giá trị true nếu $p \in I$.

---

[1] Khoa CNTT,Đại học Khoa học Huế

Mô hình Herbrand: Cho chương trình logic P, r là một quy tắc của P, I là thể hiện nào đó của P.

1. I làm thoả mãn quy tắc r nếu mỗi một hiện thành cơ sở của r mà vế phải có giá trị true thì vế phải của nó cũng có giá trị true. Ký hiệu I ặ r.

2. I nlà mô hình của P nếu I làm thoả mãn mọi quy tắc r trong P. Ký hiệu I ặ P,

3. I là mô hình cực tiểu của P nếu I là mô hình của P và không tồn tại mô hình nào khác của P được chứa trong I.

4. I là mô hình nhỏ nhất của P nếu I là mô hình của P và mọi mô hình khác của P đều chứa I.

Phép dịch chuyển: Giả sử L = (V, ≤) là một dàn, và T là một ánh xạ từ V vào V thì T được gọi là một phép dịch chuyển trên V. Hơn thế nữa, T được gọi là đơn điệu nếu và chỉ nếu a ≤ b kéo theo T(a) ≤ T(b).

1. Cho phép dịch chuyển T trên dàn L = (V, ≤) nếu tồn tại a ∈ V sao cho t(a) = a thì a được gọi là điểm bất động của T.

2. Nếu a là điểm bất động của T và không tồn tại điểm bất động b ≠ a sao cho b ≤ a thì a được gọi là điểm bất động cực tiểu.

3. Nếu a là điểm bất động của T và nếu mọi điểm bất động b ≠ a của T ta có a ≤ b thì a được gọi là điểm bất động nhỏ nhất.

Phép dịch chuyển T trên dàn L = (V, ≤) được gọi là lạm phát (*Inflationary*) nếu với mọi x ∈ V x ≤ T(x). Nếu dàn L là đầy đủ thì dãy $T(m_O)$, $T^2(m_O)$, ... hội tụ và giới hạn của nó được gọi là điểm bất động lạm phát, ở đây $m_O = \inf(V)$.

Dễ dàng nhận thấy rằng nếu V là tập hữu hạn thì (P(V), ⊆) là một dàn đầy đủ.

II. MỘT SỐ ĐỊNH LÝ QUAN TRỌNG.

Định lý 1. ([c1]). Nếu T là phép dịch chuyển đơn điệu trên dàn đầy đủ L = (V, ≤) thì T có điểm động nhỏ nhất, và điểm bất động là giới hạn của dãy $T(m_O)$, $T^1(m_O)$, $T^2(m_O)$, ..., ở đây $m_O = \inf(V)$.

Cho chương trình logic P chúng ta đưa vào phép dịch chuyển T_P và J_P như sau:

1. Với mọi $I \in P(HB)$ $T_P(I)$ là tập hợp tất cả các atom cơ sở A sao cho tồn tại một hiện hành cơ sở của một quy tắc trong P có dạng A: $\rightarrow B_1 \wedge B_2 \wedge \ldots \wedge B_k$.

2. Với mọi $I \in P(HB)$ $J_P(I) = I \cup T_P(I)$.

Định lý 2: I là mô hình của P khi và chỉ khi $T_P(I) \subseteq I$.

Chứng minh. I là mô hình của P khi và chỉ khi I làm thoả mãn mọi quy tắc của P. Điều này chỉ xảy ra khi và chỉ khi $T_P(I) \subseteq I$.

Định lý 3: Nếu I là điểm bất động cực tiểu của T_P thì I là mô hình cực tiểu của P.

Chứng minh: Nếu I là điểm bất động cực tiểu của T_P thì $I = T_P(I)$ do đó I là mô hình của T_P (theo định lý 2). Mặt khác, nếu I là điểm bất động cực tiểu của T_P thì nguyên tố cơ sở $A \in I$ tồn tại ít nhất một hiện hành cơ sở của một quy tắc r trong P có dạng: A: - $B_1 \wedge B_2 \wedge \ldots \wedge B_k$, trong đó $A \neq B_i$ $(i = 1, \ldots, k)$ và I - A r. Như vậy I là mô hình cực tiểu của P.

Định lý 4. Nếu T_P là đơn điệu thì điêm bất động nhỏ nhất của T_P là mô hình nhỏ nhất của P.

Chứng minh: Nếu T_P là đơn điệu thì theo định lý 1 tồn tại điểm bất động nhỏ nhất I của của T_P và theo định lý 3 nó là mô hình cực tiểu của P. Mặt khác do T_P đơn điệu nên từ định lý 2 suy ra một mô hình của P đều là điểm bất động của T_P. Vì vậy I được chưa trong mọi mô hình của P, hay I là mô hình nhỏ nhất.

Chương trình logic nếu không chứa dấu phủ định ở vế phải.

Định lý 5: Nếu P là dương thì T_P là đơn điệu.

Chứng minh: Lấy $I, J \in P(HB)$ và $I \subseteq J$. Nếu $A \in T_P(I)$ tồn tại một hiện hành dạng: A: f $B_1 \wedge B_2 \wedge \ldots \wedge B_k$ của một quy tắc nào đó trong P sao cho I ặ $B_1 \wedge B_2 \wedge \ldots \wedge B_k$ có nghĩa là $B_i \in I$, do đó $B_i \in J$ $\forall i$, $i = (1, \ldots, k)$. Điều này kéo theo $A \in T_P(J)$.

III. CHƯƠNG TRÌNH DATALOG VÀ DATALOG¬

Chúng ta xét một tập con của chương trình logic không chứa ký hiệu hàm. Nếu P không chứa phủ định ở vế phải của bất kỳ quy tắc nào thì gọi là chương trình Datalog, ký hiệu là *Datalog*, nếu ngược lại gọi là chương trình *Datalog* có phủ định và ký hiệu là *Datalog¬*.

Trong chương trình *Datalog* hoặc Datalog¬ chúng ta tách riêng các công thức là fact và gọi tất cả các hiện hành của chúng là EDB (cơ sở dữ liệu ban đầu). Các mệnh đề tham gia trong các fact được gọi là mệnh đề EDB. Tập các quy tắc còn gọi là chương trình chính tắc. Mệnh đề ở đầu của các quy tắc trong chương trình gọi là mệnh đề IDB.

Đồ thị phụ thuộc D-graph: Giả sử P là chương trình *Datalog* hoặc *Datalog¬*. Đồ thị phụ thuộc D(P) của P là đồ thị định hướng G = <V, N> trong đó V là tập các mệnh đề tham gia trong P, N là tập các cặp có thứ tự (p, q), ở đây p, g ∈ V sao cho tồn tại trong P một quy tắc với đầu có mệnh đề q và trong thân có mệnh đề p.

Nếu trong D(P) có chu trình thì nói rằng P là đệ quy.

Ví dụ 1: Cho chương trình:

$$r(a): f.$$

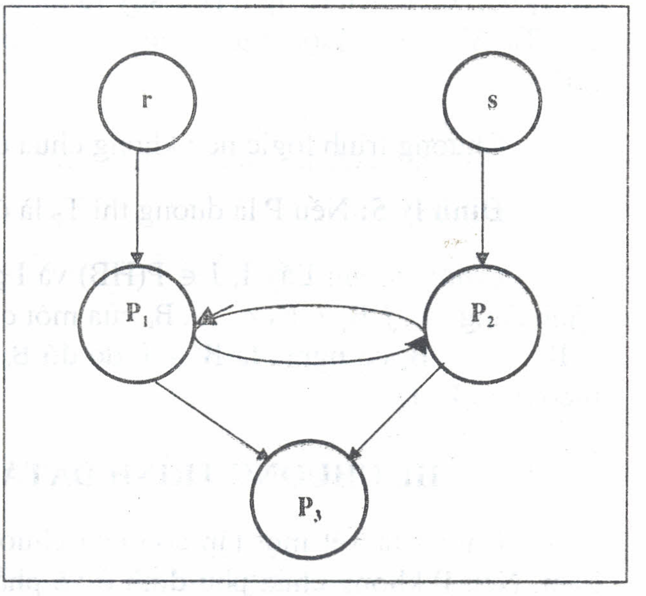
$$s(b): =.$$

$$p_1(x): f\ r(x) \wedge \neg p_2(x).$$

$$p_2(x): f\ s(x) \wedge \neg p_1(x).$$

$$p_3(x): f\ p_1(x) \wedge p_2(x).$$

Đồ thị phụ thuộc là như sau:



Datalog¬ xếp tầng.

Chương trình *Datalog*¬ P được gọi là xếp tầng nếu đối với mỗi quy tắc p(.): *f*...¬g(.)... không tồn tại một đường đi trong D(P) từ p đến q.

Nếu P là *Datalog*¬ xếp tầng thì tồn tại một thuật toán gắn số tầng t(p) cho mọi mệnh đề p trong P sao cho:

- Nếu p(.): *f*... g(.)... là một quy tắc trong P thì t(p) ≥ t(q).

- Nếu p(.): *f*...¬g(.)... là một quy tắc trong P thì t(p) > t(q).

Thuật toán xếp tầng:

for each predicate p do stratum [p]: = 1;

repeat for each rule r with head predicate p do

begin

for each negated subgoal or r with predicate p do

stratum [p]: = max(stramtum [p], 1 + stramtum[q]);

for each non-negated subgoal of r with predicate p do

stratum[p]: = max(stratum[p], stratum[q]);

end until there are no changes to any stratum or some stratum exceeds the number of prediate.

IV. NGỮ NGHĨA TRONG CHƯƠNG TRÌNH DATALOG VÀ DATALOG¬.

1. Rõ ràng chương trình *Datalog* là một chương trình logic dương, nên nếu P là một chương trình *Datalog* thì T_P là đơn điệu và do đó tồn tại điểm bất động nhỏ nhất và nó cũng là mô hình nhỏ nhất của P.

Giả thiết CWA(Close Wold Assumption).

Để xác định ngữ nghĩa của chương trình *Datalog*¬ chúng ta đưa giả thiết CWA như sau: P không suy dẫn logic ra A thì ta giả thiết P suy dẫn ¬ A theo CWA, ký hiệu P ặ^{CWA} ¬ A.

2. Giả sử ràng P là một chương trình *Datalog*¬ có α tầng. Chúng ta ký hiệu:

- T_i là tập các mệnh đề có tầng không vượt quá i.

- HB_i là tập các hiện hành cơ sở của các atom có mệnh đề trong T.

- P_i là chương trình bao gồm các quy tắc trong P với mệnh đề đầu trong T_i.

Khi đó ứng với mỗi tầng i, (i = 1, α) đặt F_{P_i} là phép dịch chuyển trên dàn đầy đủ $P(HB_i)$, ⊆ sao cho:

- Với mọi $I \in P(HB_1)$ $F_{P_1}(I) = T_{P_1}$. Rõ ràng $F_{P_1}(I)$ có điểm bất động nhỏ nhất, ký hiệu là M_1.

- Với mọi $I \in P(HB_2)$ $F_{P_2}(I) = T_{P_2}(I \cup M_1)$. F_{P_2} cũng có điểm bất động nhỏ nhất, ký hiệu là M_2.

- Tương tự như vậy ta có: với mọi $I \in P(HB_i)$ $F_{P_i}(I) = T_{P_i}(I \cup M_1 \cup M_2 \cup \ldots \cup M_{k-1})$, ở đây M_j là điểm bất động nhỏ nhất của F_{P_j}.

Định lý 6.

a) $M = M_1 \cup M_2 \cup \ldots \cup M_\alpha$ *là mô hình cực tiểu của P.*

b) Nếu M` là mô hình cực tiểu của P thì $M_1 \subset M$*.*

c) M là giới hạn của dãy $T_P^{(k)}(\phi)$ *(k = 0, 1, ...). M được gọi là mô hình perfect của P.*

Chứng minh:

a) Vì

$$T_P(M) = T_P(M_1 \cup M_2 \cup \ldots \cup M_\alpha)$$
$$= T_{P1}(M_1) \cup T_{P2}(M_1 \cup M_2) \cup \ldots \cup T_{P\alpha}(M_1 \cup M_2 \cup \ldots \cup M_\alpha)$$
$$= F_{P1}(M_1) \cup F_{P2}(M_2) \cup \ldots \cup F_{P\alpha}(M_\alpha),$$

do đó M là điểm bất động của T_P, hay M là mô hình của P.

Lấy $A \in M_i$ $(1 \le i \le \alpha)$, khi đó M - {A} P_i vì vậy M là mô hình cực tiểu của P.

b) Vì P không chứa phủ định ở vế phải nên mọi thể hiện M của P đều phải chứa M_1 và do đó M' chứa M_1.

c) Rõ ràng rằng với mỗi i $(1 \le i \le \alpha)$

$$M_1 \cup M_2 \cup \ldots \cup M_i = T_P^{(i)}(\phi)\Big|_{HB_1 \cup HB_2 \cup \ldots \cup HB_i}$$

ở đây $T_P(I)|_B$ là hạn chế T_P trên tập B. Điều này rút ra rằng $T_P^{(k)}(\phi)$ hội tụ và giới hạn của nó là M.

3. Trên đây chúng ta thấy rằng: nếu P là chương trình *Datalog* (dương) thì $T_P^{(k)}(\phi)$ hội tụ đến mô hình cực tiểu của P, nếu P là chương trình *Datalog¬* xếp tầng thì $T_P^{(k)}(\phi)$ hội tụ đến mô hình cực tiểu của P, nhưng trong trường hợp P không được xếp tầng thì dãy $T_P^{(k)}(\phi)$ nói chung không hội tụ.

Định lý 7. Giả sử J_P là một phép dịch chuyển lạm phát của chương trình Datalog¬ P khi đó dãy $J_P^{(k)}(\phi)$ hội tụ và giới hạn của nó là một mô hình của P, gọi là mô hình lạm phát của P. Thêm vào đó nếu P là chương trình *Datalog¬* có xếp tầng thì mô hình lạm phát trùng với mô hình perfect của P.

Chứng minh: Dễ dàng nhận thấy rằng $J_P^{(k)}(\phi)$ đơn điệu và bị chặn vì vậy tồn tại giá trị giới hạn của M của nó mà là điểm bất động nhỏ nhất caủ J_P, có nghĩa là $M = M \cup T_P(M)$ và do đó M là mô hình nhỏ nhất của P.

Nếu P là chương trình *Datalog¬* được xếp tầng và có α tầng thì với mọi k $(k > \alpha)$ $J_P^{(k)}(\phi) = T_P^{(k)}(\phi)$ và do đó mô hình lạm phát của P trùng với mô hình Perfect của nó.

Hạn chế của giả thiết CWA.

Để xác định ngữ nghĩa mô hình của các chương trình Datalog chúng ta đã sử dụng giả thiết CWA nhưng chúng ta cần chú ý rằng trong nhiều chương trình Datalog¬ việc sử dụng giả thiết CWA đã dẫn đến mẫu thuẫn như trong ví dụ sau:

Ví dụ 2: Xét chương trình P:

$$p(x) :- r(x).$$

$$p(b) :x- \neg p(a).$$

ở đây r là mệnh đề EDB có cơ sở dữ liệu tương ứng đã cho là {r(c)}, tập các hằng trong ngôn ngữ L_P là {a, b, c}

Dễ dàng thấy rằng P có hai mô hình:

$$M_1 = \{r(c), p(c), p(a)\}$$

$$M_2 = \{r(c), p(c), p(b)\}$$

và M_1 p(b); M_2 p(a) và do đó P ặ^{CWA} $\neg p(a) \wedge \neg p(b)$ trong lúc đó P $\neg p(a) \wedge \neg p(b)$.

KẾT LUẬN

Trong bài báo chúng tôi đã trình bày một cách khái quát việc xác định ngữ nghĩa mô hình của chương trình Datalog và sự mở rộng phủ định của chúng. Trong trường hợp Datalog yếu, ngữ nghĩa xác định (mô hình nhỏ nhất) trùng với ngữ nghĩa của lý thuyết chứng minh. Trong trường hợp mở rộng có sử dụng đến giả thiết CWA và như ví dụ nêu trên có thể dẫn đến mâu thuẫn. Vì vậy cần phải tìm hiểu các giả thiết mạnh hơn CWA.

TÀI LIỆU THAM KHẢO

[A1] **S. Abiteboul and V. Vianu**: *Datalog Extensions for Database Queries and Update, UNRIA Technical Report No. 900 (1988)*

[A2] **R. K.** *Apt: Logic Programming, Handbook of Theoretical Computer Science, Elsevier Pub. (1990).*

[C1] **S. Ceri, G. Gottlob and L. Tanca**: *Logic Programming and Databases, Springer - Verlag (1989)*

[C2] **C. L. Chang and R. C. T. Lee**: *Symbolic Logic and Mechanical Theorem Proving, Academic Press, New York (1973).*