04 Декабря 2012

Инсерционное моделирование 1

Лекция 14

Предикатный трансформер VRS требования

Предикатный трансформер

$$\mathrm{pt}(\alpha, \beta) = \gamma$$

Используется для вычисления переходов в символьном генераторе трасс

$$\gamma = \beta\,?$$

$$\mathrm{pt}(x = y, z > 0) = z > 0\,?$$

$$\mathrm{pt}(x = y, z > 0) = ((x = y)\,\&\, z > 0)$$

Конкретные трассы

Конкретное состояние: все атрибуты имеют конкретные значения.
Переходы для конкретных состояний:

$$s \xrightarrow{B} s',$$

$$B = \forall x(\alpha \rightarrow < P > \beta), \beta = R \wedge U \wedge Q$$

$$s \models \alpha', s' \models R' \wedge U' \wedge Q'$$

R есть параллельное присваивание, R' есть конъюнкция равенств
U есть оператор параллельного обновления списков (add_to, remove_from);
U' конъюнкция равенств для списков;
Q есть формула, Q' есть конкретизированная формула;
α' есть конкретизированное предусловие;
Все используемые значения должны быть определены
(если допускается неопределенность).
Только атрибуты, изменяемые с помощью R и U, а также входящие в Q
могут изменить свое значение.

Выполнение присваиваний

$$R = (u_1 := v_1) \wedge (u_2 := v_2) \wedge \ldots$$

$$R' = \varphi(u_1 := v_1) \wedge \varphi(u_2 := v_2) \wedge \ldots$$

$$\varphi(x := y) = (x = s(y))$$

$$\varphi(f(x) := y) = (f(s(x)) = s(y))$$

Модификация списков

remove для пустого списка не определен, протокол не применим

$$U = u_1 \wedge u_2 \wedge \ldots$$

$$U' = \varphi(u_1) \wedge \varphi(u_2) \wedge \ldots$$

$$L = (a_1, a_2, \ldots) \Rightarrow \varphi(\text{add\_to\_head}(L, y)) = (L = (s(y), a_1, a_2, \ldots))$$

$$L = (\ldots, a_{n-1}, a_n) \Rightarrow \varphi(\text{add\_to\_tail}(L, y)) = (L = (\ldots, a_{n-1}, a_n, s(y)))$$

$$L = (a_1, a_2, \ldots) \Rightarrow \varphi(\text{remove\_from\_head}(L)) = (L = (a_2, \ldots))$$

$$L = (\ldots, a_{n-1}, a_n) \Rightarrow \varphi(\text{remove\_from\_tail}(L)) = (L = (\ldots, a_{n-1}))$$

Символьные трассы

$$s \xrightarrow{B} s',$$
$$B = \forall x(\alpha \rightarrow < P > \beta), \beta = R \wedge U \wedge Q$$
$$s \models \alpha', s' \models R' \wedge U' \wedge Q'$$

Символьное состояние: формула
Переходы для символьных состояний:

$$s \xrightarrow{B} s',$$
$$B = \forall x(\alpha \rightarrow < P > \beta), \beta = R \wedge U \wedge Q$$
$$s \wedge \alpha \neq 0, s' \Leftrightarrow \exists x \, \mathrm{pt}(s \wedge \alpha, R \wedge U \wedge Q)$$
$$s' = B(s)$$

Переходы символьной модели детерминированы!

R есть параллельное присваивание;
U есть параллельное обновление списков;
Q есть формула;
В процессе проверки выполнимости и применения предикатного трансформера, параметры рассматриваются как новые атрибуты.

Неполные списки

$$L = (a_1,\ldots,a_n,\mathrm{Nil}), n > 0$$

$$L = (a_1,\ldots,a_n,\mathrm{Nil}; b_1,\ldots,b_m,\mathrm{Nil}), m, n > 0$$

$$L = (\mathrm{Nil}; b_1,\ldots,b_m,\mathrm{Nil})$$

$$L = (a_1,\ldots,a_n,\mathrm{Nil};\mathrm{Nil})$$

$$L = (\mathrm{Nil};\mathrm{Nil})$$

$$L = \mathrm{Nil}$$

L **– атрибутное выражение типа список**
L=u **– высказывание о возможных значениях этого выражения (см. комментарий)**
Состояние среды:

$$\gamma = \exists v(L(v) \wedge F(v))$$

L **– состояние очередей,** ***F*** **– формула без очередей**

Требования к предикатному трансформеру

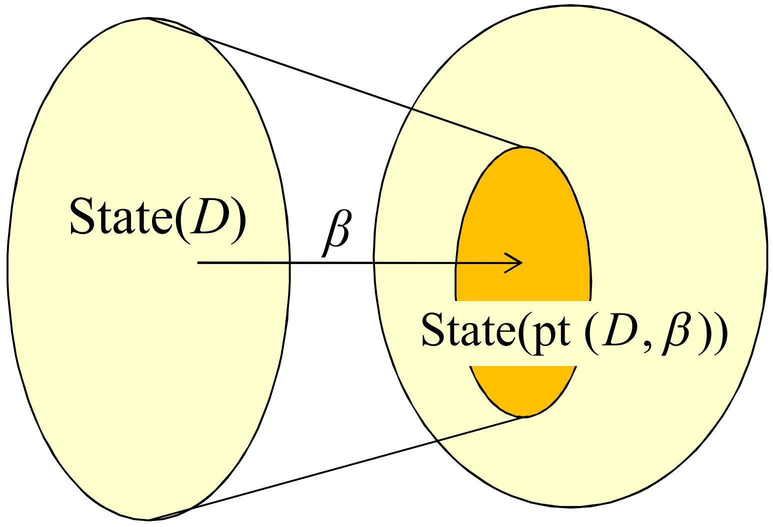
$$\mathrm{State}(D) = \{s \in C \mid s \models D\}$$

Формула D покрывает состояние s (множество состояний)
C – множество всех конкретных состояний

$$\mathrm{Nstate}(S, \beta) = \{s' \mid \exists (s \in S)(s \xrightarrow{\beta} s')\}$$

$$S \subseteq C, s \xrightarrow{\beta} s' \Leftrightarrow s \xrightarrow{1-><\Delta>\beta} s'$$

здесь $$\mathrm{Nstate}(\mathrm{State}(D), \beta) = \mathrm{State}(\mathrm{pt}(D, \beta))$$



Свойства предикатного трансформера

Конкретные трассы

$$1)\ \subseteq \quad \begin{tikzcd} s \arrow[r, "\beta"] \arrow[d, "\models"'] & s' \arrow[d, dashed, "\models"] \\ D \arrow[r, dashed, "\beta"] & \mathrm{pt}(D,\beta) \end{tikzcd} \qquad 2)\ \supseteq \quad \begin{tikzcd} s \arrow[r, dashed, "\beta"] \arrow[d, "\models"'] & s' \arrow[d, dashed, "\models"] \\ D \arrow[r, "\beta"] & \mathrm{pt}(D,\beta) \end{tikzcd}$$

Символьные трассы

$$\mathrm{Nstate}(\mathrm{State}(D),\beta) = \mathrm{State}(\mathrm{pt}(D,\beta))$$