УДК 519.682

© *М. В. Кучуганов*

СИСТЕМЫ РЕЛЯЦИОННЫХ ПРЕОБРАЗОВАНИЙ: ПРАВИЛА И КРИТЕРИЙ РЕАЛИЗУЕМОСТИ

В статье определяются и исследуются основные конструкции и семантика языка описания действий (action description language), предназначенного для описания и анализа преобразований отношений моделей ситуаций (реляционных преобразований).

Основное отличие описываемого языка KSL (Knowledge Specification Language) от традиционных (STRIPS, ADL, PDDL и т. п.) — использование кроме традиционных (STRIPS-like) правил их теоретико-множественных композиций. Это существенно повышает выразительность языка.

Точная характеризация основных свойств реляционных преобразований на языке логики предикатов первого порядка (FOL), но без использования дополнительных конструкций ситуационного исчисления, дает возможность сформулировать и доказать естественный критерий реализуемости (непротиворечивости) системы правил реляционных преобразований и, соответственно, явно описывать и исправлять логические противоречия рассматриваемой системы преобразований.

Ключевые слова: языки описания действий, STRIPS, ADL, ситуационное исчисление.

Введение

В статье определяются и исследуются основные конструкции и семантика языка описания действий (action description language), предназначенного для описания и анализа преобразований отношений моделей ситуаций (реляционных преобразований). Традиционные формализмы описания ситуаций и действий, упоминаемые в статье, подробно, с примерами описаны в обзоре [9]. Основное отличие описываемого языка преобразований KSL (Knowledge Specification Language) от традиционных, таких как STRIPS [3], ADL [7,8], PDDL [4,6], и им подобных языков заключается в том, что правила преобразования не определяют преобразование модели явно, а являются правилами вычисления спецификации (множества эффектов) преобразования.

Так как на практике изменения модели (базы фактов) ситуации могут быть весьма дорогостоящими или даже необратимыми (например, в системах управления реальными объектами), применение правила преобразования разделено на две стадии (фазы):

1) метавычисления: чтение информации, анализ исходной модели (ситуации) и вычисление (с помощью несложных функций) спецификации преобразования модели; спецификация преобразования — это множество замкнутых литер (не обязательно непротиворечивое) определяющее, что удаляется или добавляется в модель при преобразовании;

2) изменение модели: применение спецификации преобразования к модели, запись информации.

Отказ от непосредственного определения преобразований позволяет существенно, хотя и консервативно, расширить язык правил. Поскольку применению правила в конкретной модели соответствует просто множество литер, легко определяются пересечение, объединение, разность (или дополнение) и инверсия (спецификаций) правил. Использование кроме традиционных (STRIPS-like) правил их теоретико-множественных композиций существенно повышает выразительность языка. Основной результат — теорема о нормальной форме правила преобразования. Правила общего вида сводятся (транслируются) к правилам без теоретико-множественных операций, которые можно описать на языке ADL. Это означает, что описания действий на языке KSL могут быть легко включены в уже существующие системы планирования (поиска решений).

Точная характеризация основных свойств реляционных преобразований на языке логики предикатов первого порядка (FOL), но без использования дополнительных конструкций ситуационного исчисления, дает возможность сформулировать и доказать естественный критерий реализуемости (непротиворечивости) системы правил реляционных преобразований и, соответственно, явно описывать и исправлять логические противоречия рассматриваемой системы преобразований.

Далее, изложение построено следующим образом.

В § 1 определяются основные семантические понятия — спецификация отношений объектов (knowledge specification) — это формальное описание информации о состоянии мира, об эффектах действий и т. п. и операция суперпозиции (updating) спецификаций.

В § 2 определяются синтаксис и семантика правил преобразований спецификаций — реляционных преобразований.

В § 3 рассматриваются основные свойства реляционных преобразований. Они формулируются в терминах логики предикатов первого порядка и доказывается теорема о нормальной форме правила реляционного преобразования.

В § 4 определяется понятие *системы* реляционных преобразований = аксиомы (FOL-ограничения на множества возможных состояний) + правила (описания возможных действий) и понятие *реализации* системы преобразований. Доказываются критерий реализуемости и корректность и полнота FOL-характеризации системы реляционных преобразований.

В § 5 описывается пример, демонстрирующий выразительные возможности языка реляционных преобразований.

§ 1. Спецификации отношений и преобразований

Для описания систем отношений и их преобразований используем классическую логику предикатов первого порядка (FOL). Определим основные понятия.

Определение 1. *Сигнатура* (*алфавит нелогических символов*) языка логики первого порядка — это кортеж $\Sigma = [C, P]$, где C — непустое множество констант, P — непустое множество предикатных символов (различной арности).

Формулы сигнатуры Σ строятся обычным образом: из термов (констант $c_i \in C$ и переменных), предикатных символов $p_i \in P$, предиката равенства $=$, логических констант $(true, false)$, связок $(\wedge, \vee, \neg)$ и кванторов $(\forall, \exists)$.

Далее, $L(\Sigma)$ — множество формул (язык) сигнатуры Σ; $BA(\Sigma), F(\Sigma), BF(\Sigma)$ — множества основных (замкнутых) атомарных формул, литер (атомарных формул и их отрицаний) и основных (замкнутых) литер сигнатуры Σ соответственно.

Определение 2. *Спецификация* (*преобразования*) *отношений* сигнатуры Σ (Σ-спецификация) — это множество $\alpha \subseteq BF(\Sigma)$ основных литер языка $L(\Sigma)$.

Далее, $SP(\Sigma) = 2^{BF(\Sigma)}$ означает множество Σ-спецификаций. На спецификациях как множествах (литер) определены обычные операции $\cup$ (объединение), $\cap$ (пересечение), $\setminus$ (разность), а также следующие операции.

Определение 3. *Инверсия* Σ-спецификации $\alpha \in SP(\Sigma)$ — это операция

$$-\alpha : SP(\Sigma) \to SP(\Sigma) = \{P \in BF(\Sigma) \mid \neg P \in \alpha\} \cup \{\neg P \in BF(\Sigma) \mid P \in \alpha\}.$$

Определение 4. *Суперпозиция* Σ-спецификаций $\alpha, \beta \in SP(\Sigma)$ — это операция

$$\alpha * \beta : SP(\Sigma) \times SP(\Sigma) \to SP(\Sigma) = (\alpha \setminus -\beta) \cup \beta.$$

Структура $\mathbf{SP}(\Sigma) = [SP(\Sigma), *, \emptyset]$ обладает хорошими алгебраическими свойствами: она является локально-конечной полугруппой идемпотентов с единицей $\emptyset$.

Определение 5. Σ-спецификация $\alpha \in SP(\Sigma)$ *совместна* iff $\alpha \cap -\alpha = \emptyset$.

Определение 6. *Моделью* сигнатуры Σ называется максимальная (по отношению $\subseteq$) совместная Σ-спецификация.

Далее, $M(\Sigma)$ — множество моделей сигнатуры Σ.

Отношение истинности $\mu \models \phi$ замкнутой формулы ϕ в модели μ языка $L(\Sigma)$ определяется как обычно.

§ 2. Правила преобразования отношений

Опишем синтаксис и семантику правил определения (спецификаций) реляционных преобразований.

Определение 7. Множество $R(\Sigma)$ *правил преобразования отношений* языка $L(\Sigma)$ определяется индуктивно следующим образом:

1) $\emptyset$ — это правило преобразования;

2) если $\delta \in F(\Sigma)$ — литера языка $L(\Sigma)$, то $\{\delta\}$ — это правило преобразования;

3) если ρ_1, ρ_2 — правила преобразования, $\phi \in L(\Sigma)$, то $if\ \phi\ then\ \rho_1\ else\ \rho_2\ fi$ — *условная композиция правил с условием* ϕ — это правило преобразования (правила вида $if\ \phi\ then\ \rho_1\ else\ \emptyset\ fi$ сокращенно записываем $if\ \phi\ then\ \rho_1\ fi$);

4) если ρ_1, ρ_2 — правила преобразования, то $(\rho_1 \cup \rho_2)$ — *объединение правил* — это правило преобразования;

5) если ρ_1, ρ_2 — правила преобразования, то $(\rho_1 \cap \rho_2)$ — *пересечение правил* — это правило преобразования;

6) если ρ_1, ρ_2 — правила преобразования, то $(\rho_1 \setminus \rho_2)$ — *разность правил* — это правило преобразования;

7) если ρ — правило преобразования, то $(-\rho)$ — *инверсия правила* — это правило преобразования;

8) если ρ_1, ρ_2 — правила преобразования, то $(\rho_1 * \rho_2)$ — *суперпозиция правил* — это правило преобразования;

9) если ρ — правило преобразования, x — переменная, то $(\bigcup x\rho)$ и $(\bigcap x\rho)$ — *универсальные объединение и пересечение правил* соответственно — это правила преобразования.

Объединение и пересечение правил фактически (см. определение 9 ниже) представляют различные способы параллельного исполнения действий ρ_1 и ρ_2; другим интересным вариантом является операция $(\rho_1 \setminus -\rho_2) \cup (\rho_2 \setminus -\rho_1)$.

Определение 8. *Пример* (*частный случай*) правила $\rho(x_1, \ldots, x_k) \in R(\Sigma)$, где $x_1, \ldots, x_k$ — все свободные переменные правила, — правило $\rho(c_1, \ldots, c_k)$, где $c_1, \ldots, c_k \in C$.

В модели $\mu \in M(\Sigma)$ пример правила $\rho = \rho(c_1, \ldots, c_k) \in R(\Sigma)$ *однозначно* определяет спецификацию преобразования $sp\big(\mu, \rho(c_1, \ldots, c_k)\big) \in SP(\Sigma)$, а именно, дадим следующее определение.

Определение 9. *Спецификация преобразования* $sp(\mu, \rho)$ в модели $\mu \in M(\Sigma)$ по примеру правила $\rho = \rho(c_1, \ldots, c_k) \in R(\Sigma)$ определяется индуктивно следующим образом:

1) $sp(\mu, \emptyset) = \emptyset$;

2) $sp\big(\mu, \{\delta(c_1, \ldots, c_k)\}\big) = \{\delta(c_1, \ldots, c_k)\}$, где $\delta(c_1, \ldots, c_k) \in BF(\Sigma)$;

3) $sp(\mu, if\ \phi\ then\ \rho_1\ else\ \rho_2\ fi) = if\ \mu \models \phi\ then\ sp(\mu, \rho_1)\ else\ sp(\mu, \rho_2)$;

4) $sp\big(\mu, (\rho_1 \cup \rho_2)\big) = sp(\mu, \rho_1) \cup sp(\mu, \rho_2)$;

5) $sp\big(\mu, (\rho_1 \cap \rho_2)\big) = sp(\mu, \rho_1) \cap sp(\mu, \rho_2)$;

6) $sp\big(\mu, (\rho_1 \setminus \rho_2)\big) = sp(\mu, \rho_1) \setminus sp(\mu, \rho_2)$;

7) $sp\big(\mu, (-\rho)\big) = -sp(\mu, \rho)$;

8) $sp\big(\mu, (\rho_1 * \rho_2)\big) = sp(\mu, \rho_1) * sp(\mu, \rho_2)$;
9) $sp\Big(\mu, \big(\bigcup x \rho(x)\big)\Big) = \bigcup \big\{ sp\big(\mu, \rho(c)\big) \mid c \in C \big\}$,
$sp\Big(\mu, \big(\bigcap x \rho(x)\big)\Big) = \bigcap \big\{ sp\big(\mu, \rho(c)\big) \mid c \in C \big\}$.

Итак, любой пример правила $\rho = \rho(c_1, \ldots, c_k) \in R(\Sigma)$ в любой модели $\mu \in M(\Sigma)$ однозначно определяет спецификацию преобразования $sp(\mu, \rho)$. Таким образом, можно дать следующее определение.

Определение 10. *Функция преобразования* правила $\rho(x_1, \ldots, x_k)$ — это функция

$$\rho(x_1, \ldots, x_k)(\mu) : C^k \times M(\Sigma) \to SP(\Sigma) = sp\big(\mu, \rho(x_1, \ldots, x_k)\big).$$

Определение 11. Правила $\rho_1, \rho_2 \in R(\Sigma)$ *равны* (обозначается $\rho_1 = \rho_2$) iff $\forall \mu \in M(\Sigma)$ $\forall c_1, \ldots, c_k \in C$ $\rho_1(c_1, \ldots, c_k)(\mu) = \rho_2(c_1, \ldots, c_k)(\mu)$, то есть ρ_1, ρ_2 равны как функции.

§ 3. Свойства и нормализация правил

Рассмотрим основные свойства правил. Для этого используем следующие понятия.

Определение 12. *Позитивная* $\Phi_p^+[\rho]$ и *негативная* $\Phi_p^-[\rho]$ *характеристические формулы* n-арного предиката $p \in P$ сигнатуры Σ в правиле $\rho(x_1, \ldots, x_k) \in R(\Sigma)$ — это формулы $\Phi^+(x_1, \ldots, x_k, y_1, \ldots, y_n) \in L(\Sigma)$ и $\Phi^-(x_1, \ldots, x_k, y_1, \ldots, y_n) \in L(\Sigma)$ такие, что для всех $\mu \in M(\Sigma)$, для всех $a_1, \ldots, a_k, b_1, \ldots, b_n \in C$

$$\mu \models \Phi^+(a_1, \ldots, a_k, b_1, \ldots, b_n) \ \ iff \ \ p(b_1, \ldots, b_n) \in \rho(a_1, \ldots, a_k)(\mu)$$

и

$$\mu \models \Phi^-(a_1, \ldots, a_k, b_1, \ldots, b_n) \ \ iff \ \ \neg p(b_1, \ldots, b_n) \in \rho(a_1, \ldots, a_k)(\mu)$$

соответственно.

Характеристические формулы (характеристики) правил легко вычисляются рекурсивно в соответствии с семантикой правил (см. определение 9), а именно, имеет место следующее утверждение.

Утверждение 1 (вычисление характеристических формул правил). *Для любого n-арного предиката $p \in P$ сигнатуры Σ имеем*

1) $\Phi_p^+[\emptyset] = \Phi_p^-[\emptyset] = false;$
2) $\Phi_p^+\big[\{p(t_1, \ldots, t_n)\}\big] = \big((y_1 = t_1) \wedge \ldots \wedge (y_n = t_n)\big)$, $\quad \Phi_p^-\big[\{p(t_1, \ldots, t_n)\}\big] = false,$
$\Phi_p^+\big[\{q(t_1, \ldots, t_n)\}\big] = \Phi_p^-\big[\{q(t_1, \ldots, t_n)\}\big] = false$, *если* $p \neq q$,
$\Phi_p^+\big[\{\neg p(t_1, \ldots, t_n)\}\big] = false, \quad \Phi_p^-\big[\{\neg p(t_1, \ldots, t_n)\}\big] = \big((y_1 = t_1) \wedge \ldots \wedge (y_n = t_n)\big)$,
$\Phi_p^+\big[\{\neg q(t_1, \ldots, t_n)\}\big] = \Phi_p^-\big[\{\neg q(t_1, \ldots, t_n)\}\big] = false$, *если* $p \neq q$;
3) $\Phi_p^+[if\ \varphi\ then\ \rho_1\ else\ \rho_2\ fi] = \big(\varphi \wedge \Phi_p^+[\rho_1]\big) \vee \big(\neg\varphi \wedge \Phi_p^+[\rho_2]\big)$,
$\Phi_p^-[if\ \varphi\ then\ \rho_1\ else\ \rho_2\ fi] = \big(\varphi \wedge \Phi_p^-[\rho_1]\big) \vee \big(\neg\varphi \wedge \Phi_p^-[\rho_2]\big)$;
4) $\Phi_p^+[\rho_1 \cup \rho_2] = \Phi_p^+[\rho_1] \vee \Phi_p^+[\rho_2]$, $\quad \Phi_p^-[\rho_1 \cup \rho_2] = \Phi_p^-[\rho_1] \vee \Phi_p^-[\rho_2]$;
5) $\Phi_p^+[\rho_1 \cap \rho_2] = \Phi_p^+[\rho_1] \wedge \Phi_p^+[\rho_2]$, $\quad \Phi_p^-[\rho_1 \cap \rho_2] = \Phi_p^-[\rho_1] \wedge \Phi_p^-[\rho_2]$;
6) $\Phi_p^+[\rho_1 \setminus \rho_2] = \Phi_p^+[\rho_1] \wedge \neg\Phi_p^+[\rho_2]$, $\quad \Phi_p^-[\rho_1 \setminus \rho_2] = \Phi_p^-[\rho_1] \wedge \neg\Phi_p^-[\rho_2]$;
7) $\Phi_p^+[-\rho] = \Phi_p^-[\rho]$, $\quad \Phi_p^-[-\rho] = \Phi_p^+[\rho]$;
8) $\Phi_p^+[\rho_1 * \rho_2] = \big(\Phi_p^+[\rho_1] \wedge \neg\Phi_p^-[\rho_2]\big) \vee \Phi_p^+[\rho_2]$, $\quad \Phi_p^-[\rho_1 * \rho_2] = \big(\Phi_p^-[\rho_1] \wedge \neg\Phi_p^+[\rho_2]\big) \vee \Phi_p^-[\rho_2]$;
9) $\Phi_p^+[\bigcup x\rho] = \exists x\ \Phi_p^+[\rho]$, $\quad \Phi_p^-[\bigcup x\rho] = \exists x\ \Phi_p^-[\rho]$,
$\Phi_p^+[\bigcap x\rho] = \forall x\ \Phi_p^+[\rho]$, $\quad \Phi_p^-[\bigcap x\rho] = \forall x\ \Phi_p^-[\rho]$.

Итак, важнейшие свойства правил преобразования определяются формулами FOL. Такая точная и полная характеризация правил позволяет многое. В частности, на языке FOL описывается такое важное свойство правил преобразования, как

Определение 13. *Характеристика совместности* правила преобразования $\rho = \rho(x_1, \dots, x_k) \in R(\Sigma)$ по n-арному предикату $p \in P$ — это формула языка $L(\Sigma)$

$$CONS[p,\rho](x_1, \dots, x_k) = \forall\, y_1 \dots y_n \neg\big(\Phi^+[p,\rho] \wedge \Phi^-[p,\rho]\big).$$

Очевидно следующее утверждение.

Утверждение 2 (совместность спецификации правила). *Пусть $\mu \in M(\Sigma)$ — модель, $\rho = \rho(x_1, \dots, x_k) \in R(\Sigma)$ — правило преобразования. Тогда спецификация $sp\big(\mu, \rho(c_1, \dots, c_k)\big)$ совместна iff для любого (n-арного) предиката $p \in P$ $\mu \models CONS[p,\rho](c_1, \dots, c_k)$.*

Особо важным является следующее свойство правил.

Определение 14. (*Слабейшее*) *предусловие* $wp\big[\rho(x_1, \dots, x_k), \phi\big]$ предложения (замкнутой формулы) $\phi \in L(\Sigma)$ по правилу $\rho(x_1, \dots, x_k) \in R(\Sigma)$ — это формула $\psi(x_1, \dots, x_k) \in L(\Sigma)$ такая, что для всех $\mu \in M(\Sigma)$, для всех $c_1, \dots, c_k \in C$ таких, что $\rho(c_1, \dots, c_k)(\mu)$ совместно (т. е. $\mu * \rho(c_1, \dots, c_k)(\mu) \in M(\Sigma)$), выполнено

$$\mu \models \psi(c_1, \dots, c_k) \;\; iff \;\; \mu * \rho(c_1, \dots, c_k)(\mu) \models \phi.$$

Суть понятия слабейшего предусловия и его использование подробно описаны в увлекательной книге [2]. Предусловия правил вычисляются рекурсивно с использованием их характеристических формул, а именно имеет место следующее утверждение.

Утверждение 3 (вычисление предусловия правила преобразования). *Для любого правила преобразования $\rho \in R(\Sigma)$ имеем*

1) $wp(\rho, true) = true, \quad wp[\rho, false] = false;$
2) $wp\big[\rho, p(t_1, \dots, t_n)\big] = \Phi^+[p,\rho](t_1, \dots, t_n) \vee \big(p(t_1, \dots, t_n) \wedge \neg\Phi^-[p,\rho](t_1, \dots, t_n)\big)$, *где $p \in P$ — n-арный предикатный символ, $t_1, \dots, t_n$ — термы сигнатуры Σ;*
3) $wp[\rho, \phi_1 \vee \phi_2] = wp[\rho, \phi_1] \vee wp[\rho, \phi_2];$
4) $wp[\rho, \phi_1 \wedge \phi_2] = wp[\rho, \phi_1] \wedge wp[\rho, \phi_2];$
5) $wp([\rho, \neg\phi] = \neg wp[\rho, \phi];$
6) $wp[\rho, \exists\, x\phi] = \exists\, x \; wp[\rho, \phi];$
7) $wp[\rho, \forall\, x\phi] = \forall\, x \; wp[\rho, \phi].$

И наконец, с помощью характеристических формул правил, легко осуществляется их преобразование в более простую, «нормальную» форму.

Определение 15. *Атомарное правило* $\rho(x_1, \dots, x_k) \in R(\Sigma)$ сигнатуры Σ — это правило вида

$$\bigcup y_1 \dots y_n \left(if \;\, \phi(x_1, \dots, x_k, y_1, \dots, y_n) \; then \; \big\{\delta(y_1, \dots, y_n)\big\} \; fi\right),$$

где $\phi \in L(\Sigma)$, $\delta \in F(\Sigma)$ — n-арная литера языка $L(\Sigma)$.

Далее, через $AR(\Sigma)$ обозначаем множество атомарных правил сигнатуры Σ.

Подобные (с разными ограничениями) правила преобразования используются в STRIPS [3], ADL [7, 8] и многих других, более современных, системах планирования действий и поиска решений задач (см., например, [4–6] и т. д.).

Теорема 1 (нормальная форма правила). *Для любого* $\rho(x_1,\ldots,x_k) \in R(\Sigma)$ *существуют и эффективно вычисляются атомарные правила* $\rho_1(x_1,\ldots,x_k),\ldots,\rho_m(x_1,\ldots,x_k) \in AR(\Sigma)$ *такие, что для всех* $c_1,\ldots,c_k \in C$*, для всех* $\mu \in M(\Sigma)$

$$\rho(c_1,\ldots,c_k)(\mu) = \rho_1(c_1,\ldots,c_k)(\mu) \cup \ldots \cup \rho_m(c_1,\ldots,c_k)(\mu).$$

Теорема о нормальной форме правила, с точностью до обозначений, совпадает с теоремой 1 в [1] о нормальной форме ADL-правил.

Это означает, что KSL-правила преобразований с теоретико-множественными операциями, которые удобно использовать для описания действий, преобразуются посредством нормализации (трансляции) в ADL-правила и могут быть легко включены в уже существующие системы планирования действий (поиска решений).

§ 4. Системы реляционных преобразований

Полученные результаты позволяют сформулировать и доказать естественный критерий непротиворечивости (реализуемости) *систем* реляционных преобразований на языке FOL без использования дополнительных конструкций ситуационного исчисления.

Определение 16. *Система реляционных преобразований* сигнатуры Σ — это пара $R = [T, A]$, где T — множество предложений (замкнутых формул) языка $L(\Sigma)$ (*теория*), A — множество (примеров) правил реляционных преобразований $R(\Sigma)$ (*правила*).

Определение 17. *Интерпретация* сигнатуры $\Sigma = [C, P]$ — это функция $J : C \to U$, где U — непустое множество (универсум).

Каждая интерпретация однозначно сопоставляет системе преобразований $R = [T, A]$ сигнатуры $\Sigma = [C, P]$ систему $J(R) = [J(T), J(A)]$ сигнатуры $J(\Sigma) = \Sigma_U = [U, P]$, где

$J(T) = \big\{\phi\big(J(c)\big) \mid \phi(c) \in T$ — формула с константами $c \in C\big\}$,

$J(A) = \big\{\rho\big(J(c)\big) \mid \rho(c) \in A$ — правило с константами $c \in C\big\}$.

Определяя некоторую систему реляционных преобразований, мы подразумеваем, что существует в идеале *реализация* этой системы

Определение 18. *Реализация* системы $R = [T, A]$ реляционных преобразований сигнатуры Σ — это интерпретация $J : C \to U$ такая, что выполняются следующие свойства:

1) *нетривиальность*: множество $SJ(T) = \big\{\mu \in M(\Sigma_U) \mid \forall\,\phi \in J(T)\ \mu \models \phi\big\}$ (всех) моделей теории $J(T)$ сигнатуры Σ_U (*пространство моделей*) непустое;

2) *замкнутость*: $\forall\,\mu \in SJ(T)\ \forall\,\rho \in A\ \big(\mu * sp(\mu,\rho) \in SJ(T)\big)$.

Очевидно, что не всякая система реляционных преобразований имеет реализацию.

Определение 19. *Характеризация* (*FOL-теория*) системы $R = [T, A]$ реляционных преобразований сигнатуры Σ — это множество $T_A = \bigcup T_i$, $i \in N$, предложений языка $L(\Sigma)$, где:

1) $T_0 = T \cup TCON$, где $TCON = \big\{\forall\,x_1,\ldots,x_k\ CONS\big[p, \rho(x_1,\ldots,x_k)\big] \mid \rho(x_1,\ldots,x_k) \in A,$ $p \in P\big\}$ — аксиомы совместности;

2) $T_{i+1} = T_i \cup WP(T_i)$, где $WP(T_i) = \big\{\forall\,x_1,\ldots,x_k\ wp\big[\rho(x_1,\ldots,x_k), \phi\big] \mid \rho(x_1,\ldots,x_k) \in A,$ $\phi \in T_i\big\}$ — аксиомы замкнутости.

Оказывается, этого достаточно, чтобы описать основные логические свойства систем реляционных преобразований. Следующие теоремы являются очевидным следствием определений выше и теорем о непротиворечивости, корректности и полноте FOL.

Теорема 2 (критерий реализуемости). *Система реляционных преобразований* $R = [T, A]$ *сигнатуры* Σ *имеет реализацию* iff *теория* T_A *совместна, т. е. имеет модель.*

И наконец, имеет место следующая теорема.

Теорема 3 (корректность и полнота характеризации). *Пусть* $R = [T, A]$ *— система реляционных преобразований сигнатуры* Σ. *Тогда для любой замкнутой формулы* $\phi \in L(\Sigma)$ *имеют место следующие свойства:*

1) *корректность: если* $T_A \models \phi$, *то для любой реализации* $J(R)$ $\forall \mu \in SJ(R)$ $\mu \models \phi$;

2) *полнота: если для любой реализации* $J(R)$ $\forall \mu \in SJ(R)$ $\mu \models \phi$, *то* $T_A \models \phi$.

Данные результаты можно переформулировать, используя теорему полноты FOL, в терминах отношения FOL-выводимости.

Очевидно, что системы реляционных преобразований, не имеющие реализации, содержательного смысла не имеют.

С практической точки зрения крайне желательно, чтобы рассматриваемая система преобразований была «свободна от противоречий» в следующем более сильном смысле.

Определение 20. (*Логически*) *корректная* система $R = [T, A]$ реляционных преобразований сигнатуры Σ — это система преобразований такая, что:

1) T — совместна; 2) $T \vdash TCON$; 3) $T \vdash WP(T)$.

Характерным свойством логически корректных систем реляционных преобразований является следующее свойство.

Теорема 4 (универсальность реализуемости). *Если* $R = [T, A]$ *— логически корректная система реляционных преобразований сигнатуры* Σ *и* $\mu \models T$ *— модель теории* T *с универсумом* U, *то соответствующая модели* μ Σ*-интерпретация* $J : C \to U$ *является реализацией системы* $R = [T, A]$.

Приведенные результаты позволяют описывать на языке FOL основные логические противоречия рассматриваемой системы преобразований для их последующего анализа и исправления.

§ 5. Пример: железная дорога

Очевидным следствием теоремы 1 является то, что всякое простое определение действия вида

$$\langle\text{Имя действия}\rangle\big(\langle\text{Список параметров}\rangle\big) = \langle\text{Правило реляционного преобразования}\rangle$$

имеет эквивалентное, хотя и более длинное, определение на языке ADL. Однако наличие теоретико-множественных операций в описываемом языке принципиально расширяет выразительные возможности языка, позволяя определять действия рекурсивно. Рассмотрим характерный пример системы, действия в которой невозможно описать без рекурсии.

Система

Имеется линейный (без разъездов и стрелок) железнодорожный путь, неограниченный в обе стороны. На пути находятся тепловоз, который может самостоятельно двигаться в обе стороны и n обычных вагонов, которые можно двигать — толкать или тянуть (если вагон(ы) сцеплены с тепловозом).

Одно из возможных формальных описаний этой простой системы следующее.

Виды объектов

1. Участки (секции) железнодорожного пути, занумерованные целыми числами из Z.
2. Вагоны, занумерованные числами: 0 — тепловоз, $1, \ldots, n$ — обычные вагоны.

Предикаты (*таблицы*):

1) $At(a, x)$ — тепловоз (или вагон) a находится на участке x;

2) $CON(a,b)$ — вагон (или тепловоз) a сцеплен с вагоном (тепловозом) b.

Аксиомы (*ограничения на множество возможных состояний*):

1) $\forall\, a,b,x\ \big(At(a,x) \wedge At(b,x) \supset a=b\big)$ — на каждом участке может находиться 0 или 1 вагон;

2) $\forall\, a,b\ \big(CON(a,b) \equiv CON(b,a)\big)$ — симметричное отношение;

3) $\forall\, a,b,x,y\ \big(CON(a,b) \wedge At(a,x) \wedge At(b,y) \supset (y=x+1 \vee x=y+1)\big)$ — сцепленные вагоны располагаются на соседних участках пути.

Действия:

1) $CONNECT(a,b) = \forall\, x,y\ \Big(if\big(At(a,x) \wedge At(b,y) \wedge (y=x+1 \vee x=y+1)\big)\ then$
$\big\{CON(a,b), CON(b,a)\big\}\Big)$ — сцепление вагонов;

2) $DISCONNECT(a,b) = -CONNECT(a,b)$ — расцепление вагонов, инверсия сцепления;

3) $SHIFT(a,\delta) = \bigcup x\ \Big(\big(if\, At(a,x)\ then\ \{\neg At(a,x), At(a,x+\delta)\}\big)$ // сдвиг вагона
$\cup \bigcup b\ \big(if\, At(b,x+\delta)\ then\ SHIFT(b,\delta)\big)$ // толкаем — рекурсия
$\cup \bigcup b\ \big(if\ At(b,x-\delta)\ \wedge CON(a,b)\ then\ SHIFT(b,\delta)\big)\Big)$ // тянем — рекурсия
— сдвиг вагона на соседний участок пути, $\delta \in \{-1,1\}$.

Очевидно, действия не нарушают ограничения на множество возможных состояний.

Заключение

Итак, полученные результаты позволяют описывать системы действий как системы реляционных преобразований, исследовать их логические свойства и явно, на языке FOL, описывать основные противоречия рассматриваемой системы преобразований для их последующего анализа и исправления. Приведенный пример показывает возможность рекурсивных определений реляционных преобразований. Их строгое определение и основные математические свойства будут описаны в последующих статьях.

СПИСОК ЛИТЕРАТУРЫ

1. Classen J., Lakemeyer G. A Semantics for ADL as progression in the situation calculus // Proceedings of the 11th Workshop on Nonmonotonic Reasoning. 2006. P. 334–341.
2. Dijkstra E.W. A discipline of programming. 1976.
3. Fikes R., Nilsson N. STRIPS: A new approach to the application of theorem proving to problem solving // Artificial Intelligence. 1971. Vol. 2. Issue 3–4. P. 189–208.
4. Ghallab M., Howe A., Knoblock C., McDermott D., Ram A., Veloso M., Weld D., Wilkins D. PDDL — the planning domain definition language. 1998. ftp://ftp.cs.yale.edu/pub/mcdermott/software/pddl.tar.gz
5. Gelfond M., Lifschitz V. Action languages // Electronic Transactions on Artificial Intelligence. 1998. Vol. 2. Issue 3–4. P. 195–210.
6. Gerevini A., Long D. BNF description of PDDL3.0. 2005. http://zeus.ing.unibs.it/ipc-5/bnf.pdf
7. Pednault E.P.D. ADL: Exploring the middle ground between STRIPS and the situation calculus // Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR'89). Toronto. Canada. 1989. P. 324–332.
8. Pednault E.P.D. ADL and the state-transition model of action // Journal of Logic and Computation. 1994. Vol. 4. № 5. P. 467–512.
9. Трофимов И.В. Языки описания доменов и задач планирования. 2006. http://ai-center.botik.ru/planning/materials/03languages.htm

Поступила в редакцию 28.02.2015

КОМПЬЮТЕРНЫЕ НАУКИ 2015. Т. 25. Вып. 1

Кучуганов Михаил Валерьевич, с.н.с, лаборатория параллельных вычислений, Удмуртский государственный университет, 426034, Россия, г. Ижевск, ул. Университетская, 1.
E-mail: qmikle3@yandex.ru

M. V. Kuchuganov

Systems of relational transformations: rules and realizability criterion

Keywords: action description languages, STRIPS, ADL, situation calculus.

MSC: 03B70, 68T27, 68T30

The paper describes and investigates basic constructions and semantics of an action description language developed to analyze transformations of relations between situation models (relational transformations). The main difference between KSL (Knowledge Specification Language) and traditional languages (STRIPS, ADL, PDDL, etc.) is the exploitation of not only traditional (STRIPS-like) rules but also their set-theoretic compositions. This greatly increases the expressiveness of a language. A clear first order logic characterization of relational transformations (without using additional constructions of a situation calculus) makes it possible to formulate and prove a natural criterion of realizability (consistency) of the system of relational transformations and, consequently, to describe and fix the logical contradictions of the given system.

REFERENCES

1. Classen J., Lakemeyer G. A Semantics for ADL as progression in the situation calculus, *Proceedings of the 11th Workshop on Nonmonotonic Reasoning*, 2006, pp. 334–341.
2. Dijkstra E.W. *A discipline of programming*, 1976.
3. Fikes R., Nilsson N. STRIPS: A new approach to the application of theorem proving to problem solving, *Artificial Intelligence*, 1971, vol. 2, issue 3–4, pp. 189–208.
4. Ghallab M., Howe A., Knoblock C., McDermott D., Ram A., Veloso M., Weld D., Wilkins D. PDDL — the planning domain definition language, 1998. ftp://ftp.cs.yale.edu/pub/mcdermott/software/pddl.tar.gz
5. Gelfond M., Lifschitz V. Action languages, *Electronic Transactions on Artificial Intelligence*, 1998, vol. 2, issue 3–4, pp. 195–210.
6. Gerevini A., Long D. BNF description of PDDL3.0, 2005. http://zeus.ing.unibs.it/ipc-5/bnf.pdf
7. Pednault E.P.D. ADL: Exploring the middle ground between STRIPS and the situation calculus, *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR'89)*, Toronto, Canada, 1989, pp. 324–332.
8. Pednault E.P.D. ADL and the state-transition model of action, *Journal of Logic and Computation*, 1994, vol. 4, no. 5, pp. 467–512.
9. Trofimov I.V. A languages of problem domain definitions, 2006 (in Russian). http://ai-center.botik.ru/planning/materials/03languages.htm

Received 28.02.2015

Kuchuganov Mikhail Valer'evich, Senior Researcher, Parallel Computation Lab, Udmurt State University, ul. Universitetskaya, 1, Izhevsk, 426034, Russia.
E-mail: qmikle3@yandex.ru