ISSN 2079-3316 **ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ** **т. 9**, № 1(36), с. 53–83

УДК 519.682

М. В. Кучуганов

Рекурсивные определения реляционных преобразований

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

Основное отличие описываемого языка от традиционных языков описания действий (STRIPS, ADL и т.п.) заключается в использовании, кроме традиционных (STRIPS-like) правил, их теоретико-множественных композиций и рекурсии — это существенно повышает выразительность языка.

Описывается функция для вычисления эффектов действий, определенных рекурсивно и доказывается ее частичная корректность.

Ключевые слова и фразы: языки описания действий, STRIPS, ADL, ситуационное исчисление, реляционные преобразования.

Введение

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

Традиционные формализмы описания ситуаций и действий подробно, с примерами, описаны в книге [1]. Основное отличие описываемого языка KSL (Knowledge Specification Language) от традиционных, таких как STRIPS [2], ADL [3, 4] и им подобных языков описания действий заключается в том, что операторы преобразования не определяют преобразование модели явно, а являются правилами вычисления спецификации (множества эффектов) преобразования.

10.25209/2079-3316-2018-9-1-53-83

Применение оператора преобразования разделено на 2 стадии (фазы):

(1) метавычисления — чтение информации, анализ исходной модели (ситуации) и *вычисление (с помощью несложных функций) множества эффектов действия* — спецификации преобразования модели. Спецификация преобразования — это множество основных литер (не обязательно непротиворечивое) определяющее, что удаляется или добавляется в модель при преобразовании;

(2) изменение модели — *применение спецификации преобразования к модели*, запись информации.

Это позволяет существенно увеличить выразительность языка описания действий, так как появляется возможность определять (и вычислять) множество эффектов действия с помощью *теоретико-множественных операций* и *рекурсии.*

В статье [5] описываются синтаксис и семантика простых (не рекурсивных) определений реляционных преобразований и их логические свойства.

В данной статье описываются синтаксис и семантика рекурсивных определений реляционных преобразований и их вычисление.

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 операторов реляционных преобразований.

В §3 описывается *соглашение о функциональной нотации*, позволяющее существенно упростить запись формул и операторов.

В §4 рассматриваются основные свойства простых операторов реляционных преобразований - они формулируются в терминах логики предикатов первого порядка (FOL). Доказывается теорема о нормальной форме простого оператора — простые операторы общего вида преобразуются, посредством нормализации (трансляции), к операторам без теоретико-множественных операций, которые можно описать на языке ADL.

В §5 определяются синтаксис и семантика рекурсивных определений реляционных преобразований. Доказывается теорема о существовании наименьшей неподвижной точки системы позитивных определений.

В §6 описывается функция для вычисления операторов, определенных с использованием рекурсии. Доказывается, что она корректно вычисляет значение оператора в наименьшей неподвижной точке системы определений.

В §7 приводится пример системы, действия в которой (с побочными эффектами) удобно описывать с помощью рекурсии и демонстрируется вычисление рекурсивно определенного реляционного преобразования.

§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) — *это кортеж* $\Sigma = [C, P, A]$, *где* C *есть непустое множество констант,* P *— непустое конечное множество предикатных символов (различной арности),* A *— непустое конечное множество* символов преобразований *(различной арности).*

Будем обозначать арность символов $p \in P$ и $F \in A$ через $|p|$ и $|F|$ соответственно.

Формулы языка логики первого порядка 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 $(\forall, \exists)$.

Пусть $\mathsf{T}(\Sigma)$ обозначает множество термов, а $\mathsf{L}(\Sigma)$ — множество формул (язык) сигнатуры Σ.

Кортежи (конечные последовательности) термов будем обозначать буквами:

$\mathbf{a}, \mathbf{b}, \mathbf{c}$ — кортежи констант;

$\mathbf{x}, \mathbf{y}, \mathbf{z}$ — кортежи переменных;

$\mathbf{t}$ — кортежи произвольных термов.

Если арность кортежа $\mathbf{t}$ однозначно определяется контекстом, то $|\mathbf{t}|$ будет обозначать количество элементов (длину) кортежа.

Определение 2. Литеры *сигнатуры* Σ *есть формулы вида* $p(\mathbf{t})$ *(*позитивные*) или* $\neg p(\mathbf{t})$ *(*негативные*), где* $p \in P$, $\mathbf{t} \in \mathsf{T}(\Sigma)^{|p|}$.

Далее $\mathsf{F}(\Sigma)$, $\mathsf{CF}(\Sigma)$, $\mathsf{Neg}(\Sigma)$ обозначают множества литер, *основных* (не содержащих вхождений переменных) литер и множество негативных основных литер сигнатуры Σ соответственно.

Символы преобразований используются только при описании преобразований (см. §2).

Определение 3. Спецификация (преобразования) отношений *сигнатуры* Σ *(*Σ*-спецификация) есть множество* $\alpha \subseteq \mathsf{CF}(\Sigma)$ *основных литер языка* $\mathsf{L}(\Sigma)$.

Далее $\mathsf{Spec}(\Sigma) = 2^{\mathsf{CF}(\Sigma)}$ — множество Σ-спецификаций.

На спецификациях, как множествах (литер), определены обычные операции $\cup$ (объединение), $\cap$ (пересечение), $\sim$ (дополнение до множества $\mathsf{CF}(\Sigma)$), а также операции инверсии и суперпозиции:

Определение 4. Инверсия Σ*-спецификации* $\alpha \in \mathsf{Spec}(\Sigma)$ *— это операция*

$$-\alpha : \mathsf{Spec}(\Sigma) \to \mathsf{Spec}(\Sigma) = \{p(\mathbf{t}) \mid \neg p(\mathbf{t}) \in \alpha\} \cup \{\neg p(\mathbf{t}) \mid p(\mathbf{t}) \in \alpha\}.$$

Определение 5. Суперпозиция Σ *-спецификаций* $\alpha, \beta \in \mathsf{Spec}(\Sigma)$ *— это операция*

$$\alpha * \beta : \mathsf{Spec}(\Sigma) \times \mathsf{Spec}(\Sigma) \to \mathsf{Spec}(\Sigma) = (\alpha \setminus -\beta) \cup \beta.$$

Операция суперпозиции обладает следующими свойствами.

Утверждение 1 (Алгебраические свойства суперпозиции).

Структура $\boldsymbol{SP}(\Sigma) = [\mathsf{Spec}(\Sigma), *, \emptyset]$ *является локально конечной полугруппой идемпотентов с единицей* $\emptyset$ *и законом* сокращения*: для любых* $\alpha, \beta \in \mathsf{Spec}(\Sigma)$ *выполняется*

$$\alpha * \beta * \alpha = \beta * \alpha.$$

Доказательство. В записи формул будем опускать некоторые пары скобок. Порядок выполнения операций восстанавливается исходя из того, что операция $\cup$ имеет самый низкий приоритет.

Единица: для любого $\alpha \in \mathsf{Spec}(\Sigma)$

$$\alpha * \emptyset = (\alpha \setminus -\emptyset) \cup \emptyset = \alpha;$$
$$\emptyset * \alpha = (\emptyset \setminus -\alpha) \cup \alpha = \alpha.$$

Идемпотентность: для любых $\alpha \in \mathsf{Spec}(\Sigma)$

$$\alpha * \alpha = (\alpha \setminus -\alpha) \cup \alpha = \alpha.$$

Ассоциативность: для любых $\alpha, \beta, \gamma \in \mathsf{Spec}(\Sigma)$

$$\begin{aligned}(\alpha * \beta) * \gamma =& \alpha \setminus -\beta \ \cup \ \beta) \setminus -\gamma \ \cup \ \gamma \\ =&(\alpha \setminus -\beta) \setminus -\gamma \ \cup \ \beta \setminus -\gamma \ \cup \ \gamma \\ =&\alpha \setminus -(\beta \cup \gamma) \ \cup \ \beta * \gamma \\ =&\alpha \setminus -(\beta * \gamma \ \cup \ \beta \cap -\gamma) \ \cup \ \beta * \gamma \\ =&(\alpha \setminus -(\beta * \gamma)) \setminus -(\beta \cap -\gamma) \ \cup \ \beta * \gamma.\end{aligned}$$

Так как $-(\beta \cap -\gamma) = -\beta \cap \gamma \subseteq \beta * \gamma$, то имеем

$$(\alpha * \beta) * \gamma = \alpha \setminus -(\beta * \gamma) \ \cup \ \beta * \gamma = \alpha * (\beta * \gamma).$$

Сокращение: для любых $\alpha, \beta \in \mathsf{Spec}(\Sigma)$

$$\begin{aligned}\alpha * \beta * \alpha =&(\alpha \setminus -\beta \ \cup \ \beta) \setminus -\alpha \ \cup \ \alpha \\ =&(\alpha \setminus -\beta) \setminus -\alpha \ \cup \ \beta \setminus -\alpha \ \cup \ \alpha \ = \ \beta * \alpha.\end{aligned}$$

Локальная конечность: для любого $S \subseteq \mathsf{Spec}(\Sigma)$ мощность его замыкания S^* относительно операции $*$ не превосходит мощности множества конечных последовательностей элементов S *без повторений* и если множество S конечное, то и S^* также конечное. □

Определение 6. Σ*-спецификация* $\alpha \in \mathsf{Spec}(\Sigma)$ совместна *если*

$$\alpha \cap -\alpha = \emptyset.$$

Определение 7. Модель *сигнатуры* Σ *есть максимальная (по отношению* $\subseteq$*) совместная* Σ*-спецификация.*

Пусть $\mathsf{CSpec}(\Sigma)$ обозначает множество совместных спецификаций, а $\mathsf{M}(\Sigma)$ — множество моделей сигнатуры Σ.

Любую совместную спецификацию $\alpha \in \mathsf{CSpec}(\Sigma)$ можно дополнить до модели. При этом может быть использован оператор построения *«замкнутого мира»*

$$\mathsf{cwa}(\alpha) = \mathsf{Neg}(\Sigma) * \alpha.$$

Каждому предикатному символу $p \in P$ в каждой модели $\mu \in \mathsf{M}(\Sigma)$ соответствует отношение такой же арности.

Определение 8. Интерпретация *предикатного символа* $p \in P$ *в модели* $\mu \in \mathsf{M}(\Sigma)$ *есть отношение*

$$\mathsf{Rel}(p, \mu) = \{\mathbf{c} \mid p(\mathbf{c}) \in \mu\}.$$

Очевидно, что

Утверждение 2 (Преобразования моделей).
Если $\mu \in \mathsf{M}(\Sigma), \alpha \in \mathsf{CSpec}(\Sigma)$, *то* $(\mu * \alpha) \in \mathsf{M}(\Sigma)$.

Используя это свойство, преобразования моделей (действия) определим следующим образом.

Определение 9. Реляционное преобразование *сигнатуры* Σ *арности* k *есть отображение*

$$r : C^k \to (\mathsf{M}(\Sigma) \to \mathsf{Spec}(\Sigma)).$$

Далее $\mathsf{RT}(\Sigma)$ обозначает множество реляционных преобразований сигнатуры Σ.

Определение 10. Действие *k-арного реляционного преобразования* $r \in \mathsf{RT}(\Sigma)$ *есть* частичная *функция* $\mathsf{App}_r : C^k \to (\mathsf{M}(\Sigma) \to \mathsf{M}(\Sigma))$ *такая, что для любой модели* $\mu \in \mathsf{M}(\Sigma)$, *для всех* $\mathbf{c} \in C^{|\mathbf{x}|}$ *выполняется*

$$\mathsf{App}_r(\mathbf{x}, \mu) = if\ \ r(\mathbf{x}, \mu) \in \mathsf{CSpec}(\Sigma)\ \ then\ \mu * r(\mathbf{x}, \mu).$$

Таким образом, каждое реляционное преобразование *однозначно* определяет частичное преобразование моделей.

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

Определение 11. *Формулы* $\phi(\mathbf{x}), \psi(\mathbf{x}) \in \mathsf{L}(\Sigma)$ эквивалентны *(обозначается* $\phi(\mathbf{x}) \Leftrightarrow \psi(\mathbf{x})$*) если для любой модели* $\mu \in \mathsf{M}(\Sigma)$, *для всех* $\mathbf{c} \in C^{|\mathbf{x}|}$ *выполняется*

$$\mu \models \phi(\mathbf{c}) \quad \Longleftrightarrow \quad \mu \models \psi(\mathbf{c}).$$

Здесь и далее знак $\Longleftrightarrow$ означает «тогда и только тогда, когда».

§2. Операторы реляционных преобразований

Опишем синтаксис и семантику операторов реляционных преобразований и их основные виды.

Определение 12. *Множество* $\mathsf{R}(\Sigma)$ операторов (правил) реляционных преобразований *сигнатуры* Σ *определяется индуктивно следующим образом:*

(1) *если* $p \in P$ *и* $\mathbf{t} \in \mathsf{T}(\Sigma)^{|p|}$, *то* $\{p(\mathbf{t})\} \in \mathsf{R}(\Sigma)$ *(*атомарное преобразование*);*

(2) *если* $\rho \in \mathsf{R}(\Sigma), \xi \in \mathsf{R}(\Sigma)$ *и* $\phi \in \mathsf{L}(\Sigma)$, *то* $(if\ \phi\ then\ \rho\ else\ \xi\ fi) \in \mathsf{R}(\Sigma)$ *(*условная композиция операторов с условием ϕ*);*

(3) *если* $\rho \in \mathsf{R}(\Sigma)$ *и* $\xi \in \mathsf{R}(\Sigma)$, *то* $(\rho \cup \xi) \in \mathsf{R}(\Sigma)$ *(*объединение*);*

(4) *если* $\rho \in \mathsf{R}(\Sigma)$ *и* $\xi \in \mathsf{R}(\Sigma)$, *то* $(\rho \cap \xi) \in \mathsf{R}(\Sigma)$ *(*пересечение*);*

(5) *если* $\rho \in \mathsf{R}(\Sigma)$, *то* $(\sim \rho) \in \mathsf{R}(\Sigma)$ *(*дополнение*);*

(6) *если* $\rho \in \mathsf{R}(\Sigma)$, *то* $(-\rho) \in \mathsf{R}(\Sigma)$ *(*инверсия*);*

(7) *если* $\rho \in \mathsf{R}(\Sigma)$, x *— переменная, то* $(\bigcup x\rho) \in \mathsf{R}(\Sigma)$ *(*универсальное объединение*);*

(8) *если* $\rho \in \mathsf{R}(\Sigma)$, x *— переменная, то* $(\bigcap x\rho) \in \mathsf{R}(\Sigma)$ *(*универсальное пересечение*);*

(9) *если* $F \in A$, $\mathbf{t} \in \mathsf{T}(\Sigma)^{|F|}$, *то* $F(\mathbf{t}) \in \mathsf{R}(\Sigma)$ *(*применение преобразования*).*

Символы преобразований (action names) являются по сути функциональными переменными различной арности.

Определение 13. Интерпретация *символов преобразования сигнатуры* Σ *есть отображение* $J : A \to \mathsf{RT}(\Sigma)$, *сопоставляющее каждому символу преобразования* $F \in A$ $|F|$*-арное реляционное преобразование* $r \in \mathsf{RT}(\Sigma)$.

Далее $\mathsf{RT}(\Sigma)^A$ обозначает множество интерпретаций символов преобразования сигнатуры Σ.

Каждая интерпретация $J \in \mathsf{RT}(\Sigma)^A$ *однозначно* определяет интерпретацию операторов $\rho \in \mathsf{R}(\Sigma)$.

Определение 14. Реляционное преобразование $sp(J, \rho) \in \mathsf{RT}(\Sigma)$ *оператора* $\rho \in \mathsf{R}(\Sigma)$ *при интерпретации* $J \in \mathsf{RT}(\Sigma)^A$ *определяется индуктивно следующим образом:*

(1) $sp(J, \{p(\mathbf{t})\}) = \{p(\mathbf{t})\}$;

(2) $sp(J, if\ \phi\ then\ \rho\ else\ \xi\ fi) = if\ \mu \models \phi\ then\ sp(J,\rho)\ else\ sp(J,\xi)$;

(3) $sp(J, \rho \cup \xi) = sp(J,\rho) \cup sp(J,\xi)$;

(4) $sp(J, \rho \cap \xi) = sp(J,\rho) \cap sp(J,\xi)$;

(5) $sp(J, \sim \rho) = \sim sp(J,\rho)$;

(6) $sp(J, -\rho) = -sp(J,\rho)$;

(7) $sp(J, \bigcup x\rho(x)) = \bigcup\{sp(J,\rho(c)) \mid c \in C\}$;

(8) $sp(J, \bigcap x\rho(x)) = \bigcap\{sp(J,\rho(c)) \mid c \in C\}$;

(9) $sp(J, F(\mathbf{t})) = J(F)(\mathbf{t})(\mu)$.

Таким образом, каждому реляционному оператору $\rho \in \mathsf{R}(\Sigma)$ соответствует функционал

$$\rho(J) : \mathsf{RT}(\Sigma)^A \to \mathsf{RT}(\Sigma) = sp(J,\rho).$$

Определение 15. *Операторы* $\rho, \xi \in \mathsf{R}(\Sigma)$ равны *(обозначается* $\rho = \xi$*) если* $\rho(J) = \xi(J)$*, т.е.* $\forall J \in \mathsf{RT}(\Sigma)^A\ sp(J,\rho) = sp(J,\xi)$.

Будем использовать следующие обозначения для часто используемых операторов:

$\{L_1, \dots, L_k\} = \bigcup_{i \in 1..k} \rho_{L_i}$, где
  $L_i \in \mathsf{F}(\Sigma)$ - литеры,
  $\rho_{L_i} = \{p(\mathbf{t})\}$, если $L_i = p(\mathbf{t})$ и
  $\rho_{L_i} = -\{p(\mathbf{t})\}$, если $L_i = \neg p(\mathbf{t})$— *множество литер*;

$\top = \bigcup_{p \in P}(\top_p^+ \cup \top_p^-)$, где
  $\top_p^+ = \bigcup \mathbf{y}\{p(\mathbf{y})\}$,
  $\top_p^- = \bigcup \mathbf{y}\{\neg p(\mathbf{y})\}$ — *top-оператор*,
  $\top(\mu) = \mathsf{CF}(\Sigma)$ для всех $\mu \in \mathsf{M}(\Sigma)$;

$\bot = \sim \top$ — *bottom-оператор*,
  $\bot(\mu) = \emptyset$ для всех $\mu \in \mathsf{M}(\Sigma)$;

$\phi? = if\ \phi\ then\ \top\ else\ \bot\ fi$ — *оператор проверки условия, тест*;

$if\ \phi\ then\ \rho\ fi = \phi? \cap\ \rho$ — *сокращенный условный оператор*;

$I = \bigcup_{p \in P}(I_p^+ \cup I_p^-)$, где
  $I_p^+ = \bigcup \mathbf{y}(if\ p(\mathbf{y})\ then\ \{p(\mathbf{y})\})$,
  $I_p^- = \bigcup \mathbf{y}(if\ \neg p(\mathbf{y})\ then\ \{\neg p(\mathbf{y})\})$,
  $I(\mu) = \mu$ для всех $\mu \in \mathsf{M}(\Sigma)$ — *тождественный оператор.*

Определение 16. *Оператор $\rho \in \mathsf{R}(\Sigma)$ называется*

(1) простым*, если ρ не содержит символов применений преобразований;*

(2) позитивным*, если ρ не содержит символов применений преобразований в области действия операций дополнения и универсального пересечения;*

(3) конъюнктом*, если ρ является пересечением (конечного числа) простых операторов и операторов вида $F(\mathbf{t}), \sim F(\mathbf{t}), -F(\mathbf{t})$, где $F \in A$.*

Будем использовать обозначения:

$\mathsf{R}^0(\Sigma)$, $C\mathsf{R}^0(\Sigma)$ — для множеств простых и *замкнутых* (не содержащих свободных вхождений переменных) простых операторов сигнатуры Σ;

$\mathsf{R}^+(\Sigma)$, $C\mathsf{R}^+(\Sigma)$ — для множеств позитивных и замкнутых позитивных операторов сигнатуры Σ;

$K\mathsf{R}(\Sigma)$, $K\mathsf{R}^+(\Sigma)$ — для множеств конъюнктов и позитивных конъюнктов сигнатуры Σ.

§3. Функциональная нотация

Для сокращения записи формул и операторов удобно *соглашение о функциональной нотации*: предикатные символы $p \in P$ арности $n+1$ будем использовать и как функциональные символы арности n и строить с их помощью термы более сложные, чем константы и переменные.

Определение 17. *Множество $\mathsf{T}_F(\Sigma)$* квазитермов *сигнатуры Σ определяется индуктивно следующим образом:*

(1) если $t \in \mathsf{T}(\Sigma)$, то t — квазитерм;

(2) если $p \in P$ — унарный предикатный символ, то p — квазитерм;

(3) если $p \in P$ — предикатный символ арности $n > 1$, $t_1, \ldots, t_{n-1}$ — квазитермы, то $p(t_1, \ldots, t_{n-1})$ — квазитерм;

(4) если $p \in P$ — бинарный предикатный символ, t — квазитерм, то $p^{-1}(t)$ — квазитерм.

Далее $\mathsf{L}_F(\Sigma)$ и $\mathsf{R}_F(\Sigma)$ обозначают, соответственно, множества *расширенных* (содержащих вхождения квазитермов) формул и операторов сигнатуры Σ.

Количество аргументов предикатного символа в расширенной формуле (операторе) однозначно определяет, что именно он обозначает в данном вхождении: отношение (см. определение 8), либо многозначную функцию.

Расширенные формулы и операторы — это *сокращенные записи* обычных, смысл котрорых задаёт *расшифровка*. Достаточно определить функцию расшифровки с точностью до имен связанных переменных.

Сначала определим вспомогательную функцию «расшифровки терма».

Определение 18. Расшифровка DC_t *расширенных формул вида $y = t$, где t — квазитерм, y — переменная, которая не входит в t, определяется индуктивно следующим образом:*

(1) *если $t \in \mathsf{T}(\Sigma)$, то*

$$\mathrm{DC}_t(y = t) \;=\; (y = t);$$

(2) *если $t = p$, где $p \in P$ — унарный предикатный символ, то*

$$\mathrm{DC}_t(y = p) \;=\; p(y);$$

(3) *если $t = p(t_1, \ldots, t_{n-1})$, где $p \in P$ — предикатный символ арности $n > 1$, $t_1, \ldots, t_{n-1}$ — квазитермы, то*

$$\mathrm{DC}_t(y = p(t_1, \ldots, t_{n-1})) = \exists y_1 \ldots y_{n-1}\,(\mathrm{DC}_t(y_1 = t_1) \wedge \ldots \\ \wedge \mathrm{DC}_t(y_{n-1} = t_{n-1}) \wedge p(y_1, \ldots, y_{n-1}, y)),$$

где переменные $y_1, \ldots, y_{n-1}$ не входят в t;

(4) *если $t = p^{-1}(t')$, где $p \in P$ — бинарный предикатный символ, t' — квазитерм, то*

$$\mathrm{DC}_t(y = p^{-1}(t')) = \exists y_1\,(\mathrm{DC}_t(y_1 = t') \wedge p(y, y_1)),$$

где переменная y_1 не входит в t.

Пример 1. *Пусть p_1, p_2, p_3 — унарный, бинарный и тернарный предикатные символы сответственно. Тогда*

$$\begin{aligned}
\mathrm{DC}_t(y =&p_3(p_2(p_1), x)) \\
&= \exists y_1 y_2 (\mathrm{DC}_t(y_1 = p_2(p_1)) \wedge \mathrm{DC}_t(y_2 = x) \wedge p_3(y_1, y_2, y)) \\
&= \exists y_1 y_2 (\exists y_3 (\mathrm{DC}_t(y_3 = p_1) \wedge p_2(y_3, y_1)) \wedge (y_2 = x) \\
&\qquad\qquad\qquad\qquad\qquad\qquad \wedge p_3(y_1, y_2, y)) \\
&= \exists y_1 y_2 (\exists y_3 (\mathrm{DC}_t(y_3 = p_1) \wedge p_2(y_3, y_1)) \wedge (y_2 = x) \\
&\qquad\qquad\qquad\qquad\qquad\qquad \wedge p_3(y_1, y_2, y)) \\
&= \exists y_1 y_2 (\exists y_3 (p_1(y_3) \wedge p_2(y_3, y_1)) \wedge (y_2 = x) \wedge p_3(y_1, y_2, y)).
\end{aligned}$$

Упрощая формальную расшифровку, получаем

$$\mathrm{DC}_t(y = p_3(p_2(p_1), x)) = \exists y_1 y_3 (p_1(y_3) \wedge p_2(y_3, y_1) \wedge p_3(y_1, x, y)).$$

Определим функцию расшифровки атомарных формул и операторов.

Определение 19. Расшифровка DC *расширенных атомарных формул и операторов определяется следующим образом:*

(1) *если формула имеет вид $s = t$, где s, t — квазитермы, то*

$$\mathrm{DC}(s = t)) \ = \ \exists y_s y_t (\mathrm{DC}_t(y_s = s) \wedge \mathrm{DC}_t(y_t = t) \wedge (y_s = y_t)),$$

где переменные y_s, y_t не входят в квазитермы s и t;

(2) *если формула имеет вид $p(t_1, \ldots, t_n)$, где $p \in P$ — это n-арный предикатный символ, а $t_1, \ldots, t_n$ — квазитермы, то*

$$\begin{aligned}
\mathrm{DC}(p(t_1, \ldots, t_n)) = \exists y_1 \ldots y_n (&\mathrm{DC}_t(y_1 = t_1) \wedge \ldots \wedge \mathrm{DC}_t(y_n = t_n) \\
&\wedge p(y_1, \ldots, y_n)),
\end{aligned}$$

где переменные $y_1, \ldots, y_n$ не входят в $p(t_1, \ldots, t_n)$;

(3) *если оператор $\rho(t_1, \ldots, t_n)$ имеет вид $\{p(t_1, \ldots, t_n)\}$ или $F(t_1, \ldots, t_n)$, где $p \in P$, $F \in A$ — символы арности n, $t_1, \ldots, t_n$ —*

квазитермы, то

$$\mathrm{DC}(\rho(t_1,\dots,t_n)) = \bigcup_{y_1\cdots y_n} ((\mathrm{DC}_t(y_1=t_1))? \cap \dots \cap (\mathrm{DC}_t(y_n=t_n))? \\ \cap \rho(y_1,\dots,y_n)),$$

где переменные $y_1,\dots,y_n$ *не входят в* $\rho(t_1,\dots,t_n)$*;*

Расшифровка $\mathrm{DC}(\phi)$ произвольной расширенной формулы $\phi \in \mathsf{L}_F(\Sigma)$ определяется как результат замены всех ее атомарных формул $\phi_1,\dots,\phi_n$ на соответствующие расшифровки $\mathrm{DC}(\phi_1),\dots,\mathrm{DC}(\phi_n)$.

Расшифровка $\mathrm{DC}(\rho)$ произвольного расширенного оператора $\rho \in \mathsf{R}_F(\Sigma)$ определяется как результат замены всех его атомарных формул $\phi_1,\dots,\phi_n$ и операторов $\rho_1,\dots,\rho_k$ на соответствующие расшифровки $\mathrm{DC}(\phi_1),\dots,\ \mathrm{DC}(\phi_n)$ и $\mathrm{DC}(\rho_1),\dots,\mathrm{DC}(\rho_k)$.

Пример 2. *Покажем расшифровку части реляционного оператора* RShift *из примера* 5*. Здесь* $\mathsf{At},\mathsf{Next}$ *— это бинарные предикатные символы,* RShift *— символ преобразования,* v *— переменная.*

$$\mathrm{DC}(\mathsf{RShift}(\mathsf{At}^{-1}(\mathsf{Next}(\mathsf{At}(v))))) \\ = \bigcup_{y_1} (\mathrm{DC}_t(y_1 = \mathsf{At}^{-1}(\mathsf{Next}(\mathsf{At}(v))))? \cap \mathsf{RShift}(y_1)),$$

где

$$\begin{aligned} &\mathrm{DC}_t(y_1 = \mathsf{At}^{-1}(\mathsf{Next}(\mathsf{At}(v)))) \\ &\quad = \exists y_2(\mathrm{DC}_t(y_2 = \mathsf{Next}(\mathsf{At}(v))) \wedge \mathsf{At}(y_1,y_2)) \\ &\quad = \exists y_2(\exists y_3(\mathrm{DC}_t(y_3 = \mathsf{At}(v)) \wedge \mathsf{Next}(y_3,y_2)) \wedge \mathsf{At}(y_1,y_2)) \\ &\quad = \exists y_2(\exists y_3(\exists y_4((y_4 = v) \wedge \mathsf{At}(y_4,y_3)) \wedge \mathsf{Next}(y_3,y_2)) \wedge \mathsf{At}(y_1,y_2)). \end{aligned}$$

Упрощая формальную расшифровку и преобразовывая, получаем

$$\begin{aligned} &\mathrm{DC}(\mathsf{RShift}(\mathsf{At}^{-1}(\mathsf{Next}(\mathsf{At}(v))))) \\ &\quad = \bigcup y_1((\exists y_2 y_3(\mathsf{At}(v,y_3) \wedge \mathsf{Next}(y_3,y_2) \wedge \mathsf{At}(y_1,y_2)))? \cap \mathsf{RShift}(y_1)) \\ &\quad = \bigcup y_1 y_2 y_3(((\mathsf{At}(v,y_3) \wedge \mathsf{Next}(y_3,y_2) \wedge \mathsf{At}(y_1,y_2))? \cap \mathsf{RShift}(y_1)). \end{aligned}$$

Легко доказать

Утверждение 3 (Корректность функции расшифровки).

(1) *Для всех* $\phi \in \mathsf{L}(\Sigma)$, $\rho \in \mathsf{R}(\Sigma)$ *выполняется*

$$\mathrm{DC}(\phi) \Leftrightarrow \phi \text{ и } \mathrm{DC}(\rho) = \rho.$$

(2) *Для всех* $\phi(\mathbf{x}) \in \mathsf{L}_F(\Sigma), \rho(\mathbf{x}) \in \mathsf{R}_F(\Sigma)$ *и* $\mathbf{t} \in \mathsf{T}_F(\Sigma)^{|\mathbf{x}|}$ *выполняется*

$$\mathrm{DC}(\mathrm{DC}(\phi)(\mathbf{t})) \Leftrightarrow \mathrm{DC}(\phi(\mathbf{t})) \text{ и } \mathrm{DC}(\mathrm{DC}(\rho)(\mathbf{t})) = \mathrm{DC}(\rho(\mathbf{t})).$$

Далее расширенные формулы и операторы будут использоваться только в примерах.

§4. Свойства простых операторов

Многие свойства простых операторов реляционных преобразований можно охарактеризовать формулами FOL, используя отношение $\mu \models \phi$.

Для этого используем следующие понятия.

Определение 20. Позитивная p^+_ρ *и* негативная p^-_ρ характеристические формулы *языка* $\mathsf{L}(\Sigma)$ *оператора* $\rho \in \mathsf{R}^0(\Sigma)$ *относительно предиката* $p \in P$ *сигнатуры* Σ *определяются индуктивно следующим образом:*

(1)
$$p^+_{\{p(\mathbf{t})\}} = (y_1 = t_1) \wedge \ldots \wedge (y_n = t_n),$$
$$p^+_{\{q(\mathbf{t})\}} = false, \text{ если } p \neq q, \qquad p^-_{\{q(\mathbf{t})\}} = false;$$

(2)
$$p^+_{if\ \varphi\ then\ \rho\ else\ \xi\ fi} = (\varphi \wedge p^+_\rho) \vee (\neg\varphi \wedge p^+_\xi),$$
$$p^-_{if\ \varphi\ then\ \rho\ else\ \xi\ fi} = (\varphi \wedge p^-_\rho) \vee (\neg\varphi \wedge p^-_\xi);$$

(3) $$p^+_{\rho\cup\xi} = p^+_\rho \vee p^+_\xi, \qquad p^-_{\rho\cup\xi} = p^-_\rho \vee p^-_\xi;$$

(4) $$p^+_{\rho\cap\xi} = p^+_\rho \wedge p^+_\xi; \qquad p^-_{\rho\cap\xi} = p^-_\rho \wedge p^-_\xi;$$

(5) $$p^+_{\sim\rho} = \neg p^+_\rho, \qquad p^-_{\sim\rho} = \neg p^-_\rho;$$

(6) $$p^+_{-\rho} = p^-_\rho, \qquad p^-_{-\rho} = p^+_\rho;$$

(7) $$p^+_{\bigcup x\rho} = \exists x p^+_\rho, \qquad p^-_{\bigcup x\rho} = \exists x p^-_\rho;$$

(8) $$p^+_{\bigcap x\rho} = \forall x p^+_\rho, \qquad p^-_{\bigcap x\rho} = \forall x p^-_\rho.$$

Очевидно, что

Утверждение 4 (Характеризация простых операторов).

Для любого оператора $\rho(\mathbf{x}) \in \mathsf{R}^0(\Sigma)$*, для любой модели* $\mu \in \mathsf{M}(\Sigma)$*, для каждого предикатного символа* $p \in P$*, для всех* $\mathbf{a} \in C^{|\mathbf{x}|}, \mathbf{b} \in C^{|p|}$ *выполняется*

$$p(\mathbf{b}) \in \rho(\mathbf{a})(\mu) \quad \Longleftrightarrow \quad \mu \models p^+_{\rho(\mathbf{a})}(\mathbf{b})$$

и

$$\neg p(\mathbf{b}) \in \rho(\mathbf{a})(\mu) \quad \Longleftrightarrow \quad \mu \models p^-_{\rho(\mathbf{a})}(\mathbf{b}).$$

Это позволяет точно охарактеризовать многие свойства простых реляционных операторов сигнатуры Σ формулами FOL языка $\mathsf{L}(\Sigma)$.

В частности, на языке FOL описываются такие важнейшие свойства, как *пустота* и *совместность* спецификации оператора.

Утверждение 5 (Пустота спецификации оператора).

Для любого оператора $\rho(\mathbf{x}) \in \mathsf{R}(\Sigma)$*, для любой модели* $\mu \in \mathsf{M}(\Sigma)$*, для всех* $\mathbf{c} \in C^{|\mathbf{x}|}$ *выполняется*

$$\rho(\mathbf{c})(\mu) = \emptyset \quad \Longleftrightarrow \quad \mu \models empty_\rho(\mathbf{c}),$$

где

$$empty_\rho(\mathbf{x}) = \bigwedge_{p \in P} \forall \mathbf{y} \neg \big(p^+_{\rho(\mathbf{x})}(\mathbf{y}) \vee p^-_{\rho(\mathbf{x})}(\mathbf{y})\big)$$

означает формулу языка $\mathsf{L}(\Sigma)$*, выражающую* условие пустоты спецификации *оператора* $\rho(\mathbf{x})$.

Утверждение 6 (Совместность спецификации оператора).

Для любого оператора $\rho(\mathbf{x}) \in \mathsf{R}(\Sigma)$*, для любой модели* $\mu \in \mathsf{M}(\Sigma)$*, для всех* $\mathbf{c} \in C^{|\mathbf{x}|}$ *выполняется*

$$\rho(\mathbf{c})(\mu) \in \mathsf{CSpec}(\Sigma) \quad \Longleftrightarrow \quad \mu \models cons_\rho(\mathbf{c}),$$

где

$$cons_\rho(\mathbf{x}) = \bigwedge_{p \in P} \forall \mathbf{y} \neg \big(p^+_{\rho(\mathbf{x})}(\mathbf{y}) \wedge p^-_{\rho(\mathbf{x})}(\mathbf{y})\big)$$

есть формула языка $\mathsf{L}(\Sigma)$*, выражающая* условие совместности спецификации *оператора* $\rho(\mathbf{x})$.

С помощью характеристических формул, легко осуществляется преобразование простых реляционных операторов в более простую, «нормальную» форму.

Теорема 1 (Нормальная форма простого оператора). *Для любого простого реляционного оператора* $\tau(\mathbf{x}) \in \mathsf{R}^0(\Sigma)$

$$\tau(\mathbf{x}) = \bigcup_{p \in P} (\tau_p^+ \cup \tau_p^-),$$

где для каждого предикатного символа $p \in P$

$$\tau_p^+ = \bigcup \mathbf{y}(if\ p^+_{\tau(\mathbf{x})}(\mathbf{y})\ then\ \{p(\mathbf{y})\}\ fi),$$

$$\tau_p^- = \bigcup \mathbf{y}(if\ p^-_{\tau(\mathbf{x})}(\mathbf{y})\ then\ \{\neg p(\mathbf{y})\}\ fi).$$

Характеристическая нормальная форма простого реляционного оператора, с точностью до обозначений, совпадает с нормальной формой ADL-правила (см. [6, Theorem 1]).

Это означает, что *простые* реляционные операторы 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 поиска решений и планирования действий.

§5. Определения реляционных преобразований

Для (рекурсивного) определения реляционных преобразований будем использовать только позитивные реляционные операторы.

Пусть для любых реляционных преобразований $r, r' \in \mathsf{RT}(\Sigma)$

$$r \sqsubseteq_{\mathsf{RT}(\Sigma)} r' \quad \Longleftrightarrow \quad \forall \mathbf{c} \in C\ \forall \mu \in \mathsf{M}(\Sigma)\ r(\mathbf{c})(\mu) \subseteq r'(\mathbf{c})(\mu)$$

и для любых интерпретаций $J, J' \in \mathsf{RT}(\Sigma)^A$

$$J \sqsubseteq J' \quad \Longleftrightarrow \quad \forall F \in A\ J(F) \sqsubseteq_{\mathsf{RT}(\Sigma)} J'(F)).$$

Очевидно, что отношения $\sqsubseteq_{\mathsf{RT}(\Sigma)}$ и $\sqsubseteq$ являются *полными* частичными порядками на множествах $\mathsf{RT}(\Sigma)$ и $\mathsf{RT}(\Sigma)^A$ соответственно.

Легко доказать, что

Утверждение 7 (Монотонность и непрерывность позитивных операторов). *Любой позитивный реляционный оператор* $\rho \in \mathsf{R}^+(\Sigma)$ *является:*

(1) монотонным, *т.е. для любых* $J, J' \in \mathsf{RT}(\Sigma)^A$ *если* $J \sqsubseteq J'$, *то* $\rho(J) \sqsubseteq_{\mathsf{RT}(\Sigma)} \rho(J')$;

(2) непрерывным, *т.е. для любого* направленного $S \subseteq \mathsf{RT}(\Sigma)^A$ *множество* $\rho(S) = \{\rho(J)\}_{J \in S}$ *также направленное и выполняется* $\rho(\sup_{\sqsubseteq}(S)) = \sup_{\sqsubseteq_{\mathsf{RT}(\Sigma)}}(\rho(S))$.

Ограничение на использование операции универсального пересечения в определении позитивного оператора *необходимо*, чтобы гарантировать непрерывность. Например,

Пример 3. *Пусть сигнатура* $\Sigma = [N, \{>, \mathsf{Next}\}, \{F\}]$, *где* N — *множество натуральных чисел,* $>, \mathsf{Next}$ — *бинарные предикатные саимволы, а* F — *унарный символ преобразования.*

Оператор

$$\rho = \bigcap x(if\ x > 0\ then\ F(\mathsf{Next}^{-1}(x))\ else\ \top)$$

является монотонным, но не непрерывным, так как в «стандартной» модели μ_N, *где* $\mu_N \models x > y \iff N \models x > y$ *и* $\mu_N \models \mathsf{Next}(x, y) \iff N \models x + 1 = y$ *для направленного множества интерпретаций*

$$J_{\leq} = \{J_{\leq i}(F)(x) = \neg(i > x)?\}_{i \in N}$$

имеем

$$\sup_{\sqsubseteq}(J_{\leq}) = \top, \qquad \rho(\sup_{\sqsubseteq}(J_{\leq})) = \top,$$

но для всех $i \in N$

$$\rho(J_{\leq i}(F)) = \bot, \qquad \sup_{\sqsubseteq_{\mathsf{RT}(\Sigma)}}(\rho(J_{\leq i})) = \bot.$$

Опишем синтаксис и семантику определений реляционных преобразований.

Пусть каждому символу преобразования $F \in A$ сигнатуры Σ сопоставлен оператор $\rho_F(\mathbf{x}) \in \mathsf{R}(\Sigma)$. Результат замены в операторе $\rho \in \mathsf{R}(\Sigma)$ каждого вхождения применения преобразования вида $F(\mathbf{t})$,$F \in A$ на вхождение соответствующего оператора $\rho_F(\mathbf{t})$ *будем обозначать* $\rho[\forall_{F \in A}(F \leftarrow \rho_F)]$, а простые операторы $\rho[\forall_{F \in A}(F \leftarrow \bot)]$ и $\rho[\forall_{F \in A}(F \leftarrow \top)]$ — как $\rho[\bot]$ и $\rho[\top]$ соответственно.

Определение 21. Система (позитивных) определений реляционных преобразований *сигнатуры* Σ *есть отображение* $\mathfrak{D} : A \to \mathsf{R}^+(\Sigma)$ *такое, что для всех* $F \in A$ *количество свободных переменных оператора* $\mathfrak{D}(F)$ *не превышает арности символа преобразования* F.

Как обычно, система определений $\mathfrak{D}$ сигнатуры Σ описывается множеством равенств

$$\{F(\mathbf{x}) = \rho_F(\mathbf{x}) \mid F \in A, \rho_F = \mathfrak{D}(F)\}.$$

Далее $\mathsf{Def}^+(\Sigma)$ обозначает множество (позитивных) определений реляционных преобразований сигнатуры Σ.

Определение 22. Преобразование интерпретаций по системе определений $\mathfrak{D} \in \mathsf{Def}^+(\Sigma)$ *это такое отображение* $J_{\mathfrak{D}} : \mathsf{RT}(\Sigma)^A \to \mathsf{RT}(\Sigma)^A$*, для которого* $J_{\mathfrak{D}}(J)(F) = sp(J, \mathfrak{D}(F))$ *выполняется при всех* $F \in A$.

Семантика системы определений $\mathfrak{D} \in \mathsf{Def}^+(\Sigma)$ определяется через понятие наименьшей неподвижной точки (ННТ) преобразования $J_{\mathfrak{D}}$.

Определение 23. Неподвижная точка *преобразования интерпретаций* $J_{\mathfrak{D}}$ *по системе определений* $\mathfrak{D} \in \mathsf{Def}^+(\Sigma)$ *есть такая интерпретация* $J \in \mathsf{RT}(\Sigma)^A$*, что* $J_{\mathfrak{D}}(J) = J$.

Применяя теорему Клини о наименьшей неподвижной точке непрерывного функционала (см. подробное обсуждение в замечательной книге [7]) получаем, что

Теорема 2 (Существование наименьшей неподвижной точки преобразования $J_{\mathfrak{D}}$). *Пусть задана система позитивных определений* $\mathfrak{D} \in \mathsf{Def}^+(\Sigma)$ *и для всех* $F \in A$ *для соответствующей последовательности операторов* $(\tau^i_{F\uparrow})_{i \in N}$ *выполняется*

$$\tau^0_{F\uparrow} = \bot, \quad \tau^{i+1}_{F\uparrow} = \mathfrak{D}(F)[\forall_{F \in A}(F \leftarrow \tau^i_{F\uparrow})].$$

Тогда для всех $F \in A$, $i \in N$*, для любой модели* $\mu \in \mathsf{M}(\Sigma)$*, для всех* $\mathbf{c} \in C$ *выполняется*

$$\tau^i_{F\uparrow}(\mathbf{c})(\mu) \subseteq \tau^{i+1}_{F\uparrow}(\mathbf{c})(\mu)$$

и реляционное преобразование $fix(J_{\mathfrak{D}}) \in \mathsf{RT}(\Sigma)^A$ *такое, что*

$$fix(J_{\mathfrak{D}})(F)(\mathbf{c})(\mu) = \bigcup_{i \in N} \{\tau^i_{F\uparrow}(\mathbf{c})(\mu)\},$$

является наименьшей *(относительно $\sqsubseteq$) неподвижной точкой преобразования $J_{\mathfrak{D}}$.*

Для каждого $F \in A$ реляционное преобразование $fix(J_{\mathfrak{D}})(F) \in \mathsf{RT}(\Sigma)$ по системе определений $\mathfrak{D} \in \mathsf{Def}^{+}(\Sigma)$ будем обозначать через $F_{\mathfrak{D}}$.

Хотя для каждого символа преобразования $F \in A$ преобразование $F_{\mathfrak{D}}$ *всюду определено*, оно не всегда «вычислимо».

§6. Вычисление рекурсивно определенных реляционных преобразований

Опишем один из возможных способов вычисления реляционных преобразований, определенных рекурсивно.

Определение 24. Вычислительная последовательность *оператора $\rho \in C\mathsf{R}^{+}(\Sigma)$ по системе определений $\mathfrak{D} \in \mathsf{Def}^{+}(\Sigma)$ есть последовательность операторов $(\rho^{i})_{i \in N}$ такая, что*

$$\rho^{0} = \rho,\ \rho^{i+1} = (\rho^{i})',$$

где $\rho' = \rho[\forall_{F \in A}(F \leftarrow \mathfrak{D}(F))]$.

Очевидно, что

Утверждение 8 (Свойства вычислительной последовательности). *Пусть $(\rho^{i})_{i \in N}$ — вычислительная последовательность оператора $\rho \in C\mathsf{R}^{+}(\Sigma)$ по системе определений $\mathfrak{D} \in \mathsf{Def}^{+}(\Sigma)$. Тогда для любой модели $\mu \in \mathsf{M}(\Sigma)$, для всех $i \in N$ выполняется:*

(1) $\rho^{i}[\bot](\mu) \subseteq \rho^{i+1}[\bot](\mu),\ \rho^{i+1}[\top](\mu) \subseteq \rho^{i}[\top](\mu),;$

(2) $\bigcup\{\rho^{i}[\bot](\mu)\}_{i \in N} = \rho(J_{\mathfrak{D}})(\mu)$, *где $J_{\mathfrak{D}}$ — ННТ системы $\mathfrak{D} \in \mathsf{Def}^{+}(\Sigma)$;*

(3) *если $\rho^{i}[\top](\mu) \subseteq \rho^{i}[\bot](\mu)$, то $\rho^{i}[\bot](\mu) = \rho(J_{\mathfrak{D}})(\mu)$.*

К сожалению, последнего свойства недостаточно для вычисления значения многих, даже очень просто определенных операторов.

Например, не вычисляется оператор F, определенный системой $\mathfrak{D} = \{F = F\}$, хотя очевидно, что $F_{\mathfrak{D}} = \bot$.

Для доказательства корректности описываемого ниже алгоритма нам понадобятся следующие отношения.

Пусть для любой системы позитивных определений $\mathfrak{D} \in \mathsf{Def}^+(\Sigma)$, для любых операторов $\rho, \xi \in C\mathsf{R}^+(\Sigma)$, для любой модели $\mu \in \mathsf{M}(\Sigma)$

$$\xi \sqsubseteq_{\sup} \rho \iff \bigcup\{\xi^i[\bot](\mu)\}_{i\in N} \subseteq \bigcup\{\rho^i[\bot](\mu)\}_{i\in N},$$

$$\xi \sim_{\sup} \rho \iff \xi \sqsubseteq_{\sup} \rho \ \text{ и } \ \rho \sqsubseteq_{\sup} \xi.$$

Пусть для любых операторов $\rho(\mathbf{y}), \xi(\mathbf{z}) \in \mathsf{R}(\Sigma)$, для любой модели $\mu \in \mathsf{M}(\Sigma)$

$$\xi \sqsubseteq_\mu \rho \iff \forall J \in \mathsf{RT}(\Sigma)^A \ \bigcup \mathbf{z}\xi(\mathbf{z})(J)(\mu) \subseteq \bigcup \mathbf{y}\rho(\mathbf{y})(J)(\mu),$$

$$\xi \sim_\mu \rho \iff \xi \sqsubseteq_\mu \rho \ \text{ и } \ \rho \sqsubseteq_\mu \xi.$$

Очевидно, что отношения $\sqsubseteq_{\sup}, \sqsubseteq_\mu$ являются отношениями частичного порядка (*предельный* и *модельный* порядки соответственно), а $\sim_{\sup}, \sim_\mu$ — соответстствующие отношения эквивалентности.

Для «вычисления» модельного порядка в алгоритме будем использовать отношение *коньюктивного* (частичного) порядка, которое определяется следующим образом.

Пусть для любых операторов $\rho, \xi \in \mathsf{R}(\Sigma)$, для любой модели $\mu \in \mathsf{M}(\Sigma)$

$$\xi \sqsubseteq_K \rho \iff \text{существуют } \tau_\rho, \tau_\xi \in \mathsf{R}^0(\Sigma) \text{ и } \sigma_\rho, \sigma_\xi \in \mathsf{R}(\Sigma)$$

такие, что

$$\rho = \tau_\rho(\mathbf{y}) \cap \sigma_\rho(\mathbf{y}), \ \xi = \tau_\xi(\mathbf{y}, \mathbf{z}) \cap \sigma_\xi(\mathbf{y}, \mathbf{z}) \cap \sigma_\rho(\mathbf{z})$$

и

$$\mu \models \forall \mathbf{y} \exists \mathbf{z} \ \mathit{empty}(\tau_\xi(\mathbf{y}, \mathbf{z}) \setminus \tau_\rho(\mathbf{z})).$$

Лемма 1 (Свойства частичных порядков). *Для любой системы позитивных определений* $\mathfrak{D} \in \mathsf{Def}^+(\Sigma)$, *для любых* $\rho, \xi, \sigma \in \mathsf{R}(\Sigma)$, $\mu \in \mathsf{M}(\Sigma)$ *выполняется:*

(1) *если* $\xi \sqsubseteq_K \rho$, *то* $\xi \sqsubseteq_\mu \rho$;

(2) *если* $\rho' \sqsubseteq_\mu \xi \cup \sigma$ *и* $\sigma \sqsubseteq_\mu \rho$, *то* $\rho' \sqsubseteq_{\sup} \xi \cup \sigma[\bot]$.

Доказательство.

(1) По утверждению 5, если $\mu \models \forall \mathbf{y} \exists \mathbf{z} \ \mathit{empty}(\tau_\xi(\mathbf{y}, \mathbf{z}) \setminus \tau_\rho(\mathbf{z}))$, то $\forall \mathbf{y} \exists \mathbf{z} (\tau_\xi(\mathbf{y}, \mathbf{z}) \sqsubseteq_\mu \tau_\rho(\mathbf{z}))$.

Тогда, если $\xi \sqsubseteq_K \rho$, то $\xi \sqsubseteq_\mu \rho$.

(2) Здесь $\rho, \xi, \sigma \in C\mathsf{R}^+(\Sigma)$, так как отношение $\sqsubseteq_{\mathrm{sup}}$ определено только для таких операторов. Докажем, что для всех $i \in N$ выполняется $\rho^{i+1} \sqsubseteq_\mu \xi^i \cup \sigma$.

Базис индукции: $\rho^1 \sqsubseteq_\mu \xi^0 \cup \sigma$ по условию.

Шаг индукции. Если $\rho^i \sqsubseteq_\mu \xi^{i-1} \cup \sigma$, то $\rho^{i+1} \sqsubseteq_\mu \xi^i \cup \sigma'$. Так как $\sigma \sqsubseteq_\mu \rho$, то и $\sigma' \sqsubseteq_\mu \rho'$, а значит $\xi^i \cup \sigma' \sqsubseteq_\mu \xi^i \cup \rho' \sqsubseteq_\mu \xi^i \cup (\xi \cup \sigma)$. Так как $\xi \sqsubseteq_\mu \xi'$, то $(\xi^i \cup \xi) \cup \sigma \sqsubseteq_\mu \xi^i \cup \sigma$, а значит $\rho^{i+1} \sqsubseteq_\mu \xi^i \cup \sigma$.

Таким образом, для всех $i \in N$ имеем $\rho^{i+1}[\bot] \sqsubseteq_\mu \xi^i[\bot] \cup \sigma[\bot]$, а значит $\rho' \sqsubseteq_{\mathrm{sup}} \xi \cup \sigma[\bot]$.

$\square$

Любое бинарное отношение $\mathsf{Rel} \subseteq \mathsf{R}(\Sigma) \times \mathsf{R}(\Sigma)$ однозначно определяет *аналогичное* отношение между подмножествами $\Gamma, \Delta \subseteq \mathsf{R}(\Sigma)$ $\mathsf{Rel} \subseteq 2^{\mathsf{R}(\Sigma)} \times 2^{\mathsf{R}(\Sigma)}$ такое, что

$$(\Gamma \ \mathsf{Rel}\ \Delta) \iff \forall \rho \in \Gamma \exists \xi \in \Delta\ (\rho \ \mathsf{Rel}\ \xi).$$

Далее $\mathcal{P}_{<\omega}(S)$ обозначает множество всех конечных подмножеств множества (операторов) S.

Пусть для любого $\Gamma \in \mathcal{P}_{<\omega}(\mathsf{R}(\Sigma))$ *(полное) объединение* множества операторов Γ есть оператор

$$\bigcup \Gamma \ = \ if\ \Gamma = \emptyset\ then\ \bot\ else \bigcup_{\rho(\mathbf{y}) \in \Gamma} (\bigcup \mathbf{y} \rho(\mathbf{y})).$$

Очевидно, что

Утверждение 9 (Дизъюнктивная нормальная форма). *Существует вычислимая функция* $\mathsf{DNF}(\rho) : C\mathsf{R}^+(\Sigma) \to \mathcal{P}_{<\omega}(K\mathsf{R}^+(\Sigma))$ *сопоставляющая* каждому *замкнутому оператору* $\rho \in C\mathsf{R}^+(\Sigma)$ *непустое конечное множество его конъюнктов* $\mathsf{DNF}(\rho) \subseteq K\mathsf{R}^+(\Sigma)$ *такое, что* $\rho = \bigcup \mathsf{DNF}(\rho)$.

Очевидно, что ограничение отношения $\sqsubseteq_K$ на множество (позитивных) конъюнктов $K\mathsf{R}^+(\Sigma)$ разрешимо, а значит

Утверждение 10 (Минимизация). *Если отношение* $\mu \models \phi$ *разрешимо, то существует вычислимая функция* $\mathsf{Min}(\Gamma) : \mathcal{P}_{<\omega}(K\mathsf{R}^+(\Sigma)) \to \mathcal{P}_{<\omega}(K\mathsf{R}^+(\Sigma))$ *сопоставляющая* каждому *конечному множеству конъюнктов* $\Gamma \subseteq K\mathsf{R}^+(\Sigma)$ *его* минимальное *(по отношению* $\subseteq$*) подмножество* $\mathsf{Min}(\Gamma) \subseteq \Gamma$ *такое, что* $\Gamma \sqsubseteq_K \mathsf{Min}(\Gamma)$.

Для вычисления определенных рекурсивно реляционных преобразований определим функцию $\mathsf{RT\text{-}Calc}$: $\mathcal{P}_{<\omega}(K\mathsf{R}^+(\Sigma)) \times \mathcal{P}_{<\omega}(\mathsf{R}^+(\Sigma)) \to C\mathsf{R}^0(\Sigma)$ Пусть

$$
\begin{aligned}
&\mathsf{RT\text{-}Calc}\ (\Gamma, \Delta) = \\
&\qquad if\ \ \mu \models empty(\textstyle\bigcup \Delta[\top] \setminus \bigcup(\Delta \cup \Gamma)[\bot])\ then\ \bigcup(\Delta \cup \Gamma)[\bot] \\
&\qquad else\ \mathsf{RT\text{-}Calc}\ (\Gamma \cup \Delta, \mathsf{Calc}\ (\Gamma, \Delta)),
\end{aligned}
$$

где

$$
\begin{aligned}
&\mathsf{Calc}\ (\Gamma, \Delta) = \mathsf{Cut}(\Gamma \cup \Delta, \mathsf{Min}(\mathsf{New}(\Delta))), \\
&\qquad \mathsf{New}(\Delta)\ = \textstyle\bigcup \{\mathsf{DNF}(\rho')\}_{\rho \in \Delta}, \\
&\qquad \mathsf{Cut}(\Gamma, \Delta) = \{\xi \in \Delta \mid \neg \exists \rho \in \Gamma\ \xi \sqsubseteq_K \rho\},
\end{aligned}
$$

DNF и Min — функции, определенные в утверждениях 9 и 10 соответственно.

Интуитивно, Γ есть множество (частично) вычисленных операторов, Δ — множество операторов, которые «еще имеет смысл» вычислять (делать подстановки).

Лемма 2 (Свойства функций в $\mathsf{RT\text{-}Calc}$). *Для любой системы позитивных определений $\mathfrak{D} \in \mathsf{Def}^+(\Sigma)$, для любых $\Gamma, \Delta \in \mathcal{P}_{<\omega}(K\mathsf{R}^+(\Sigma))$ и $\mu \in \mathsf{M}(\Sigma)$ выполняется:*

(1) $\mathsf{New}(\Gamma) = \Gamma'$, *где* $\Gamma' = \{\rho' \mid \rho \in \Gamma\}$*;*

(2) $\mathsf{Min}(\Gamma) \sim_\mu \Gamma$*;*

(3) $\mathsf{Calc}\ (\Gamma, \Delta) \sim_\mu \mathsf{Cut}(\Gamma, \Delta')$*;*

(4) $\mathsf{Cut}(\Gamma, \Delta) \subseteq \Delta,\ \Delta \setminus \mathsf{Cut}(\Gamma, \Delta) \sqsubseteq_\mu \Gamma$.

Доказательство.

(1) По определению функции New и утверждению 9;

(2) По определению функции Min и пункту 1 леммы 1;

(3) По определению функции Calc и пунктам 1, 2 леммы 2

$$\mathsf{Cut}(\Gamma \cup \Delta, \mathsf{Min}(\mathsf{New}(\Delta))) = \mathsf{Min}(\mathsf{Cut}(\Gamma \cup \Delta, \mathsf{New}(\Delta)),$$

$$\mathsf{Min}(\mathsf{Cut}(\Gamma, \mathsf{New}(\Delta)) \sim_\mu \mathsf{Cut}(\Gamma, \Delta');$$

(4) По определению функции Cut и пункту 1 леммы 1. □

Итак,

Теорема 3 (Корректность функции $\mathsf{RT\text{-}Calc}$). *Для любой модели $\mu \in \mathsf{M}(\Sigma)$, любой системы позитивных определений $\mathfrak{D} \in \mathsf{Def}^+(\Sigma)$, любого оператора $\rho \in C\mathsf{R}^+(\Sigma)$*

(1) *если отношение $\mu \models \phi$ разрешимо, то функция $\mathsf{RT\text{-}Calc}$ вычислима;*

(2) *если вычисление $\mathsf{RT\text{-}Calc}\ (\{\}, \{\rho\})$ заканчивается, то*

$$\mathsf{RT\text{-}Calc}\ (\{\}, \{\rho\})(\mu) = \rho(J_{\mathfrak{D}})(\mu).$$

Доказательство.

(1) По определению функции $\mathsf{RT\text{-}Calc}$ и утверждениям 9, 10;

(2) Вычислению $\mathsf{RT\text{-}Calc}\ (\{\}, \{\rho\})$ соответствует последовательность

$$\Gamma_0 = \{\},\ \Delta_0 = \{\rho\},\ \Gamma_{i+1} = \Gamma_i \cup \Delta_i,\ \Delta_{i+1} = \mathsf{Calc}\ (\Gamma_i, \Delta_i),$$

где Γ_i, Δ_i — это аргументы функции $\mathsf{RT\text{-}Calc}$ перед выполнением $i+1$ шага вычисления.

Для всех $i \in N$ имеем $(\Gamma_{i+1} \cup \Delta_{i+1}) \sqsubseteq_{\sup} (\Gamma_i \cup \Delta_i)'$, так как $\Gamma_{i+1} = \Gamma_i \cup \Delta_i \sqsubseteq_\mu (\Gamma_i \cup \Delta_i)'$ и $\Delta_{i+1} \subseteq \Delta_i'$.

По пункту 4 леммы 2, для всех $i \in N$ имеем

$$\Delta_i' \sqsubseteq_\mu \mathsf{Cut}\ (\Gamma_i \cup \Delta_i, \Delta_i') \cup \Gamma_i \cup \Delta_i,$$

а значит, по пункту 2 леммы 1

$$\Delta_i' \sqsubseteq_{\sup} \mathsf{Cut}\ (\Gamma_i \cup \Delta_i, \Delta_i') \cup \Gamma_i \cup \Delta_i[\bot].$$

Тогда для всех $i \in N$ имеем $(\Gamma_i \cup \Delta_i)' \sqsubseteq_{\sup} \Gamma_{i+1} \cup \Delta_{i+1}$, так как

$$\begin{aligned}
(\Gamma_i \cup \Delta_i)' =& (\bigcup_{j<i} \Delta_j \cup \Delta_i)' = (\bigcup_{j\leq i} \Delta_j)' = \bigcup_{j\leq i} \Delta_j' \\
\sqsubseteq_{\sup}& \bigcup_{j\leq i} (\mathsf{Cut}\ (\Gamma_j \cup \Delta_j, \Delta_j') \cup \Gamma_j \cup \Delta_j[\bot]) \\
\sim_\mu& \bigcup_{j\leq i} (\Delta_{j+1} \cup \Gamma_j \cup \Delta_j[\bot]) \\
=& (\bigcup_{0<j<i} \Delta_j \cup \Delta_i \cup \Delta_{i+1}) \cup \Gamma_i \cup \bigcup_{j\leq i} \Delta_j[\bot] \\
=& \Gamma_{i+1} \cup \Delta_{i+1} \cup \bigcup_{0<j<i} \Delta_j \cup \bigcup_{j\leq i} \Delta_j[\bot] = \Gamma_{i+1} \cup \Delta_{i+1}.
\end{aligned}$$

Таким образом, для всех $i \in N$ имеем $(\Gamma_i \cup \Delta_i)' \sim_{\sup} \Gamma_{i+1} \cup \Delta_{i+1}$, а значит $\Gamma_i \cup \Delta_i \sim_{\sup} \Gamma_{i+1} \cup \Delta_{i+1}$.

Тогда для любых $i, j \in N$ имеем $\Gamma_i \cup \Delta_i \sim_{\sup} \Gamma_j \cup \Delta_j$ и $\Gamma_i \cup \Delta_i \sim_{\sup} \Gamma_0 \cup \Delta_0 = \{\rho\}$.

Если на шаге $i+1$ выполняется условие завершения

$$\mu \models empty(\bigcup \Delta_i[\top] \setminus \bigcup(\Delta_i \cup \Gamma_i)[\bot]),$$

то для любого $j > i$ будет выполнятся

$$\bigcup(\Delta_j \cup \Gamma_j)[\bot])(\mu) \subseteq \bigcup(\Delta_i \cup \Gamma_i)[\bot])(\mu),$$

а значит

$$\bigcup(\Delta_i \cup \Gamma_i) \sim_{\sup} \bigcup(\Delta_i \cup \Gamma_i)[\bot]).$$

Тогда вычисление можно закончить, так как

$$\rho \sim_{\sup} \bigcup(\Delta_i \cup \Gamma_i) \sim_{\sup} \bigcup(\Delta_i \cup \Gamma_i)[\bot])$$

и

$$\rho(J_{\mathfrak{D}})(\mu) = \bigcup(\Delta_i \cup \Gamma_i)(J_{\mathfrak{D}})(\mu) = \bigcup(\Delta_i \cup \Gamma_i)[\bot])(\mu).$$

$\square$

Покажем процесс вычисления реляционного оператора на простом примере.

Пример 4. *Пусть Σ — сигнатура, содержащая 0-арные символы преобразований F_1, F_2 и*

$$\mathfrak{D} = \{F_1 = A_1F_1 \cup B_1F_2 \cup C_1, F_2 = A_2F_1 \cup B_2F_2 \cup C_2\},$$

где $A_1, B_1, C_1, A_2, B_2, C_2 \in C\mathsf{R}^0(\Sigma)$.

В записи конъюнктов в уравнениях и далее опущены очевидные, в данном примере, применения операции $\cap$.

Вычислим F_1. Итак, $\Gamma_0 = \{\}$, $\Delta_0 = \{F_1\}$.

ШАГ 1. Вычисляем

$$\begin{aligned} \Gamma_1 =& \Gamma_0 \cup \Delta_0 = \{F_1\}; \\ \cup\Delta_0[\top] =& F_1[\top] = \top; \\ \cup(\Delta_0 \cup \Gamma_0)[\bot] =& \cup\,\Gamma_1[\bot] = F_1[\bot] = \bot. \end{aligned}$$

Так как $\cup\Delta_0[\top](\mu) \nsubseteq \cup(\Delta_0 \cup \Gamma_0)[\bot](\mu)$, *продолжаем вычисление.*

$$\begin{aligned}\mathsf{New}(\Delta_0) =& \{A_1F_1, B_1F_2, C_1\};\\ \Delta_1 =& \mathsf{Cut}(\Gamma_1, \mathsf{Min}(\mathsf{New}(\Delta_0)))\\ =& \mathsf{Cut}(\{F_1\}, \{A_1F_1, B_1F_2, C_1\}) = \{B_1F_2, C_1\}.\end{aligned}$$

Здесь функцией Cut *сокращен конъюнкт* A_1F_1 *из* $\mathsf{New}(\Delta_0)$, *который уже можно не вычислять, так как* $A_1F_1 \sqsubseteq_K F_1$ *– он поглощается уже вычисляемым конъюнктом* F_1 *из* Γ_1.

ШАГ 2. Вычисляем

$$\begin{aligned}\Gamma_2 =& \Gamma_1 \cup \Delta_1 = \{F_1, B_1F_2, C_1\};\\ \cup\Delta_1[\top] =& \cup\{B_1, C_1\};\\ \cup(\Delta_1 \cup \Gamma_1)[\bot] =& \cup\Gamma_2[\bot] = C_1.\end{aligned}$$

Так как $\cup\Delta_1[\top](\mu) \nsubseteq \cup(\Delta_1 \cup \Gamma_1)[\bot](\mu)$, *продолжаем вычисление.*

$$\begin{aligned}\mathsf{New}(\Delta_1) =& \mathsf{New}(\{B_1F_2\}) \cup \mathsf{New}(\{C_1\})\\ =& \{B_1A_2F_1, B_1B_2F_2, B_1C_2\} \cup \{C_1\};\\ \Delta_2 =& \mathsf{Cut}(\Gamma_2, \mathsf{Min}(\mathsf{New}(\Delta_1)))\\ =& \mathsf{Cut}(\{F_1, B_1F_2, C_1\}, \{B_1A_2F_1, B_1B_2F_2, B_1C_2, C_1\})\\ =& \{B_1C_2\}\end{aligned}$$

Здесь функцией Cut *сокращены конъюнкты из* $\mathsf{New}(\Delta_1)$, *которые уже можно не вычислять, так как они поглощаются уже вычисляемыми или вычисленными конъюнктами из* Γ_2 :

$$B_1A_2F_1 \sqsubseteq_K F_1,\ B_1B_2F_2 \sqsubseteq_K B_1F_2,\ C_1 \sqsubseteq_K C_1.$$

ШАГ 3. Вычисляем

$$\begin{aligned}\Gamma_3 =& \Gamma_2 \cup \Delta_2 = \{F_1, B_1F_2, C_1, B_1C_2\}.\\ \cup\Delta_2[\top] =& B_1C_2;\\ \cup(\Delta_2 \cup \Gamma_2)[\bot] =& \cup\Gamma_3[\bot] = \cup\{C_1, B_1C_2\}.\end{aligned}$$

Так как $\cup\Delta_2[\top](\mu) \subseteq \cup(\Delta_2 \cup \Gamma_2)[\bot](\mu)$, *заканчиваем вычисление с результатом* $C_1 \cup B_1C_2$.

Более содержательный пример рассматривается далее.

§7. Пример: Wagon World

Рассмотрим пример системы, действия с побочными эффектами в которой удобно описывать с помощью рекурсии.

Пример 5 (Wagon World).

Система: *на линейном (без разъездов и стрелок) железнодорожном пути, неограниченном в обе стороны, находятся n вагонов, которые можно двигать, сцеплять и расцеплять.*

Объекты:

$s, t, \dots \in \mathbb{Z}$ *[участки железнодорожного пути],*

$v, w, \dots \in \{1, \dots, n\}$ *[вагоны].*

Отношения:

$\mathsf{Next}(s, t)$ *[участок t следует за участком s],*

$\mathsf{At}(v, s)$ *[вагон v находится на участке s],*

$\mathsf{Linked}(v, w)$ *[вагон v сцеплен с вагоном w].*

Действия:

$\mathsf{Link}(v, w) = if\ \mathsf{Next}(v, w) \vee \mathsf{Next}(w, v)\ then\ \{\mathsf{Linked}(v, w), \mathsf{Linked}(w, v)\}$

$\mathsf{UnLink}(v, w) = -\mathsf{Link}(v, w)$ *[сцепление и расцепление вагонов];*

$$\begin{aligned}
\mathsf{RShift}(v) =& \{\neg\mathsf{At}(v, \mathsf{At}(v)), \mathsf{At}(v, \mathsf{Next}(\mathsf{At}(v)))\} \quad \textit{[вагон вправо]}\\
&\cup \mathsf{RShift}(\mathsf{At}^{-1}(\mathsf{Next}(\mathsf{At}(v)))) \quad \textit{[толкает (рекурсия!)]}\\
&\cup \mathsf{RShift}(\mathsf{Linked}(v)) \quad \textit{[тянет прицепленные (рекурсия!)]},\\
\mathsf{LShift}(v) =& \{\neg\mathsf{At}(v, \mathsf{At}(v)), \mathsf{At}(v, \mathsf{Next}^{-1}(\mathsf{At}(v)))\} \quad \textit{[вагон влево]}\\
&\cup \mathsf{LShift}(\mathsf{At}^{-1}(\mathsf{Next}^{-1}(\mathsf{At}(v)))) \quad \textit{[толкает (рекурсия!)]}\\
&\cup \mathsf{LShift}(\mathsf{Linked}(v)) \quad \textit{[тянет прицепленные (рекурсия!)]}.
\end{aligned}$$

Вычислим $\mathsf{RShift}(3)$ в модели

$$\mu = \mathsf{cwa}(\mu_{\mathsf{At}} \cup \mu_{\mathsf{Linked}} \cup \mu_{\mathsf{Next}}),$$

где

$$\begin{aligned}
\mu_{\mathsf{At}} =& \{\mathsf{At}(1,1), \mathsf{At}(2,2), \mathsf{At}(3,3), \mathsf{At}(4,4)\},\\
\mu_{\mathsf{Linked}} =& \{\mathsf{Linked}(1,2), \mathsf{Linked}(2,1), \mathsf{Linked}(2,3), \mathsf{Linked}(3,2)\},\\
\mu_{\mathsf{Next}} =& \{\mathsf{Next}(x, x+1) \mid x \in Z\}.
\end{aligned}$$

Итак, $\Gamma_0 = \{\}$, $\Delta_0 = \{\mathsf{RShift}(3)\}$.

ШАГ 1. Вычисляем

$$\begin{aligned}\Gamma_1 =& \Gamma_0 \cup \Delta_0 = \{\mathsf{RShift}(3)\};\\ \cup\Delta_0[\top] =& \mathsf{RShift}(3)[\top] = \top;\\ \cup(\Delta_0 \cup \Gamma_0)[\bot] =& \cup\,\Gamma_1[\bot] = \mathsf{RShift}(3)[\bot] = \bot.\end{aligned}$$

Так как $\cup\Delta_0[\top](\mu) \not\subseteq \cup(\Delta_0 \cup \Gamma_0)[\bot](\mu)$, то продолжаем вычисление.

$$\begin{aligned}\mathsf{New}(\Delta_0) =& \mathsf{New}(\{\mathsf{RShift}(3)\})\\ =& \{\mathsf{RShift}(\mathsf{At}^{-1}(\mathsf{Next}(\mathsf{At}(3))))\} \cup \{\mathsf{RShift}(\mathsf{Linked}(3))\} \cup K_3\\ =& \{\mathsf{RShift}(4)\} \cup \{\mathsf{RShift}(2)\} \cup K_3,\end{aligned}$$

где

$$\begin{aligned}K_3 =& \{\neg\mathsf{At}(3, \mathsf{At}(3)), \mathsf{At}(3, \mathsf{Next}(\mathsf{At}(3)))\}\\ =& \{\neg\mathsf{At}(3,3), \mathsf{At}(3,4)\};\\ \Delta_1 =& \mathsf{Cut}(\Gamma_1, \mathsf{Min}(\mathsf{New}(\Delta_0))) = \mathsf{New}(\Delta_0).\end{aligned}$$

ШАГ 2. Вычисляем

$$\begin{aligned}\Gamma_2 = \Gamma_1 \cup \Delta_1 =& K_3 \cup \{\mathsf{RShift}(2), \mathsf{RShift}(3), \mathsf{RShift}(4)\};\\ \cup\Delta_1[\top] =& \top;\\ \cup(\Delta_1 \cup \Gamma_1)[\bot] =& \cup\,\Gamma_2[\bot] = K_3.\end{aligned}$$

Так как $\cup\Delta_1[\top](\mu) \not\subseteq \cup(\Delta_1 \cup \Gamma_1)[\bot](\mu)$, то продолжаем вычисление.

$$\mathsf{New}(\Delta_1) = K_3 \cup \mathsf{New}(\{\mathsf{RShift}(2)\}) \cup \mathsf{New}(\{\mathsf{RShift}(4)\}),$$

где

$$\begin{aligned}\mathsf{New}(\{\mathsf{RShift}(2)\}) =& \{\mathsf{RShift}(\mathsf{At}^{-1}(\mathsf{Next}(\mathsf{At}(2))))\} \cup \{\mathsf{RShift}(\mathsf{Linked}(2))\}\\ & \cup K_2\\ =& \{\mathsf{RShift}(3)\} \cup \{\mathsf{RShift}(1)\} \cup K_2,\end{aligned}$$

где

$$\begin{aligned}K_2 =& \{\neg\mathsf{At}(2, \mathsf{At}(2)),\\ \mathsf{At}(2, \mathsf{Next}(\mathsf{At}(2)))\} =& \{\neg\mathsf{At}(2,2), \mathsf{At}(2,3)\};\end{aligned}$$

$$\begin{aligned}\mathsf{New}(\{\mathsf{RShift}(4)\}) =& \{\mathsf{RShift}(\mathsf{At}^{-1}(\mathsf{Next}(\mathsf{At}(4))))\} \cup \{\mathsf{RShift}(\mathsf{Linked}(4))\} \\ & \cup K_4 \\ =& \{\} \cup \{\} \cup K_4 = K_4,\end{aligned}$$

где

$$\begin{aligned}K_4 =& \{\neg\mathsf{At}(4, \mathsf{At}(4)), \\ \mathsf{At}(4, \mathsf{Next}(\mathsf{At}(4)))\} =& \{\neg\mathsf{At}(4,4), \mathsf{At}(4,5)\};\end{aligned}$$

Итак, $\mathsf{New}(\Delta_1) = K_2 \cup K_3 \cup K_4 \cup \{\mathsf{RShift}(1), \mathsf{RShift}(3)\}$. Поэтому $\Delta_2 = \mathsf{Cut}(\Gamma_2, \mathsf{Min}(\mathsf{New}(\Delta_1))) = K_2 \cup K_4 \cup \{\mathsf{RShift}(1)\}$. Здесь функцией Cut сокращены конъюнкт $\mathsf{RShift}(3)$ и конъюнкты из K_3, так как они уже есть в Γ_2.

ШАГ 3. Вычисляем

$$\begin{aligned}\Gamma_3 = \Gamma_2 \cup \Delta_2 =& K_2 \cup K_3 \cup K_4 \\ & \cup \{\mathsf{RShift}(1), \mathsf{RShift}(2), \mathsf{RShift}(3), \mathsf{RShift}(4)\}; \\ \cup\Delta_2[\top] =& \top; \\ \cup(\Delta_2 \cup \Gamma_2)[\bot] =& \cup\,\Gamma_3[\bot] = K_2 \cup K_3 \cup K_4.\end{aligned}$$

Так как $\cup\Delta_2[\top](\mu) \nsubseteq \cup(\Delta_2 \cup \Gamma_2)[\bot](\mu)$, то продолжаем вычисление.

$$\mathsf{New}(\Delta_2) = K_2 \cup K_4 \cup \mathsf{New}(\{\mathsf{RShift}(1)\}),$$

где

$$\begin{aligned}\mathsf{New}(\{\mathsf{RShift}(1)\}) =& \{\mathsf{RShift}(\mathsf{At}^{-1}(\mathsf{Next}(\mathsf{At}(1))))\} \cup \{\mathsf{RShift}(\mathsf{Linked}(1))\} \\ & \cup K_1 \\ =& \{\mathsf{RShift}(2)\} \cup \{\} \cup K_1,\end{aligned}$$

где

$$\begin{aligned}K_1 =& \{\neg\mathsf{At}(1, \mathsf{At}(1)), \mathsf{At}(1, \mathsf{Next}(\mathsf{At}(1)))\} \\ =& \{\neg\mathsf{At}(1,1), \mathsf{At}(1,2)\};\end{aligned}$$

Итак,

$$\begin{aligned}\mathsf{New}(\Delta_2) =& K_1 \cup K_2 \cup K_4 \cup \{\mathsf{RShift}(2)\}. \\ \Delta_3 =& \mathsf{Cut}(\Gamma_3, \mathsf{Min}(\mathsf{New}(\Delta_2))) = K_1.\end{aligned}$$

Здесь функцией Cut сокращены конъюнкт $\mathsf{RShift}(2)$ и конъюнкты из K_2, K_4, так как они уже есть в Γ_3.

ШАГ 4. Вычисляем

$$\begin{aligned}\Gamma_4 =& \Gamma_3 \cup \Delta_3 = K_1 \cup K_2 \cup K_3 \cup K_4 \\ & \qquad \cup \{\mathsf{RShift}(1), \mathsf{RShift}(2), \mathsf{RShift}(3), \mathsf{RShift}(4)\}; \\ \cup\Delta_3[\top] =& K_1; \\ \cup(\Delta_3 \cup \Gamma_3)[\bot] =& \cup\Gamma_4[\bot] = K_1 \cup K_2 \cup K_3 \cup K_4.\end{aligned}$$

Так как $\cup\Delta_3[\top](\mu) \subseteq \cup(\Delta_3 \cup \Gamma_3)[\bot](\mu)$, заканчиваем вычисление с результатом $K_1 \cup K_2 \cup K_3 \cup K_4$.

Итак, $\mathsf{RShift}(3)(\mu) = \{\neg\mathsf{At}(3,3), \mathsf{At}(3,4)\} \cup \{\neg\mathsf{At}(4,4), \mathsf{At}(4,5)\}$
$\cup\{\neg\mathsf{At}(2,2), \mathsf{At}(2,3)\} \cup \{\neg\mathsf{At}(1,1), \mathsf{At}(1,2)\}$.

Результатом применения действия $\mathsf{RShift}(3)(\mu)$ к модели μ являются суперпозиция модели и множества вычисленных эффектов действия

$$\mathsf{App}_{\mathsf{RShift}}(3,\mu) = \mu * \mathsf{RShift}(3)(\mu) = \mathsf{cwa}(\mu_{\mathsf{At}} \cup \mu_{\mathsf{Linked}} \cup \mu_{\mathsf{Next}}),$$

где $\mu_{\mathsf{At}} = \{\mathsf{At}(1,2), \mathsf{At}(2,3), \mathsf{At}(3,4), \mathsf{At}(4,5)\}$, а $\mu_{\mathsf{Linked}}, \mu_{\mathsf{Next}}$ такие же, как до применения действия.

Заключение

Итак, полученные результаты позволяют описывать системы действий, как системы рекурсивно определенных реляционных преобразований и эффективно вычислять эффекты действий, определенных с использованием рекурсии.

Приведенный пример демонстрирует выразительные возможности рекурсивных определений реляционных преобразований.

Список литературы

[1] F. Harmelen, V. Lifschitz, B. Porter (eds.), *Handbook of Knowledge Representation*, Foundations of Artificial Intelligence, 1st edition, Elsevier, 2008, 1034 p. ↑53

[2] R. Fikes, N. Nilsson. "STRIPS: A new Approach to the Application of Theorem Proving to Problem Solving", *Artificial Intelligence*, **2**:3–4 (1971), pp. 189–208, doi ↑53

[3] E. P. D. Pednault. "ADL: Exploring the middle ground between STRIPS and the Situation Calculus", *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, KR'89 (Royal York Hotel, Toronto, Ontario, Canada, May 15–19, 1989), Morgan Kaufmann Publishers Inc., San Francisco, 1989, pp. 324–332, ↑53

[4] E. P. D. Pednault. "ADL and the State-Transition Model of Action", *J. Log. Comput.*, **4**:5 (1994), pp. 467–512, ↑53

[5] М. В. Кучуганов. «Системы реляционных преобразований: правила и критерий реализуемости», *Вестник Удмуртского университета. Математика. Механика. Компьютерные науки*, **25**:1 (2015), с. 117–125, URL ↑54

[6] J. Claßen, G. Lakemeyer. "A Semantics for ADL as Progression in the Situation Calculus", *Proceedings of the 11th International Workshop on Non-Monotonic Reasoning*, NMR'06 (Lake District, UK, 30 May–1 June, 2006), Institut für Informatik, Technische Universität Clausthal, pp. 334–341, URL ↑67

[7] V. Stoltenberg-Hansen, I. Lindstrom, E. R. Griffor, *Mathematical Theory of Domains*, Cambridge Tracts in Theoretical Computer Science, vol. **22**, 1st edition, Cambridge University Press, 2008, 364 p., ↑69

Рекомендовал к публикации *д.ф.-м.н. Н.Н. Непейвода*

Пример ссылки на эту публикацию:

М. В. Кучуганов. «Рекурсивные определения реляционных преобразований». *Программные системы: теория и приложения*, 2018, **9**:1(36), с. 53–83.

10.25209/2079-3316-2018-9-1-53-83

URL http://psta.psiras.ru//read/psta2018_1_53-83.pdf

Об авторе:

Михаил Валерьевич Кучуганов

Преподавал на математическом факультете Удмуртского государственного университета. Автор нескольких статей по конструктивным логикам, синтезу схем программ и баз данных, планированию действий.

0000-0002-9550-2753

e-mail: qmikle1@yandex.ru

UDC 519.682

Mikhail Kuchuganov. *A recursive definitions of relational transformations.*

ABSTRACT. In the paper we describe and investigate a basic constructions and semantics of a new action description language KSL (Knowledge Specification Language) which is based on the notion of relational transformation.

The main difference the described language from traditional ones (STRIPS, ADL, etc.) is in extension traditional (STRIPS-like) rules by means of its set-theoretic compositions and recursion - this greatly increases the expressiveness of the language.

Also we define a function to calculate a recursively defined relational transformations and prove its partial correctness. (*in Russian*).

Key words and phrases: action description languages, STRIPS, ADL, situation calculus, relational transformations.

References

[1] F. Harmelen, V. Lifschitz, B. Porter (eds.), *Handbook of Knowledge Representation*, Foundations of Artificial Intelligence, 1st edition, Elsevier, 2008, 1034 p., ISBN : 0444522115/9780444522115

[2] R.~Fikes, N.~Nilsson. "STRIPS: A new Approach to the Application of Theorem Proving to Problem Solving", *Artificial Intelligence*, **2**:3–4 (1971), pp. 189–208,

[3] E. P. D.~Pednault. "ADL: Exploring the middle ground between STRIPS and the Situation Calculus", *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, KR'89 (Royal York Hotel, Toronto, Ontario, Canada, May 15–19, 1989), Morgan Kaufmann Publishers Inc., San Francisco, 1989, pp. 324–332,

[4] E. P. D.~Pednault. "ADL and the State-Transition Model of Action", *J. Log. Comput.*, **4**:5 (1994), pp. 467–512,

[5] M. V.~Kuchuganov. "Systems of Relational Transformations: Rules and Realizability Criterion", *Vestnik Udmurtskogo universiteta. Matematika. Mekhanika. Komp'yuternyye nauki*, **25**:1 (2015), pp. 117–125,

[6] J.~Claßen, G.~Lakemeyer. "A Semantics for ADL as Progression in the Situation Calculus", *Proceedings of the 11th International Workshop on Non-Monotonic Reasoning*, NMR'06 (Lake District, UK, 30~May–1~June, 2006), Institut für Informatik, Technische Universität Clausthal, pp. 334–341,

[7] V.~Stoltenberg-Hansen, I.~Lindstrom, E. R.~Griffor, *Mathematical Theory of Domains*, Cambridge Tracts in Theoretical Computer Science, vol. **22**, 1st edition, Cambridge University Press, 2008, 364 p.,

10.25209/2079-3316-2018-9-1-53-83

Sample citation of this publication:

Mikhail Kuchuganov. “A recursive definitions of relational transformations”. *Program Systems: Theory and Applications*, 2018, **9**:1(36), pp. 53–83. (*In Russian*). 10.25209/2079-3316-2018-9-1-53-83

http://psta.psiras.ru//read/psta2018_1_53-83.pdf