ИЗВЕСТИЯ РАН. ТЕОРИЯ И СИСТЕМЫ УПРАВЛЕНИЯ, 2020, № 6, с. 120–131

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

УДК 004.8 + 519.682

РЕКУРСИВНЫЕ ОПРЕДЕЛЕНИЯ ТАБЛИЧНЫХ ПРЕОБРАЗОВАНИЙ

© 2020 г. М. В. Кучуганов

Удмуртский государственный ун-т, Ижевск, Россия

e-mail: qmikle1@yandex.ru

Поступила в редакцию 05.06.2019 г.
После доработки 20.12.2019 г.
Принята к публикации 30.03.2020 г.

Рассматриваются основные конструкции и семантика языка описания действий (action description language), предназначенного для определения и вычисления преобразований отношений моделей ситуаций (реляционных преобразований). Основное отличие рассматриваемого 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, что существенно повышает выразительность языка. Приводится вариант рекурсивных определений *табличных* реляционных преобразований, сохраняющих конечность спецификаций изменяемых отношений модели. Описывается функция для вычисления табличных преобразований, доказываются ее частичная корректность и достаточное условие, гарантирующее завершение вычисления.

DOI: 10.31857/S0002338820040113

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*).

Языки описания действий используются в интеллектуальных системах управления и поддержки принятия решений для описания действий в моделируемой системе и автоматического синтеза планов действий.

Традиционные математические формализмы описания ситуаций и действий рассматриваются с примерами и обширной библиографией в [1], а методы планирования – в [2, 3].

Основное отличие языка KSL (knowledge specification language) от традиционных, таких, как STRIPS [4], ADL [5, 6], PDDL [7, 8] 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.

Применение оператора преобразования разделено на две стадии (фазы).

1. Метавычисления – чтение информации, анализ исходной модели (ситуации) и *расчет* (*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.

Такой подход 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 [9] описываются синтаксис и семантика простых (без рекурсии) операторов реляционных преобразований и рассматриваются их логические свойства.

Характеристическая нормальная форма простого реляционного оператора с точностью до обозначений совпадает с нормальной формой ADL-правила [10]. Это означает, что *простые* реляционные операторы с теоретико-множественными операциями, которые удобно использо-

вать для описания эффектов действий, преобразуются посредством нормализации (трансляции) в ADL-правила и могут быть легко включены в уже существующие системы поиска решений и планирования действий. В [11] описываются алгебраические свойства операции суперпозиции (updating) спецификаций, синтаксис и семантика рекурсивных определений реляционных преобразований общего вида и их вычисление.

В данной статье рассматривается вариант рекурсивных определений *табличных* реляционных преобразований, сохраняющих конечность описаний (спецификаций) изменяемых отношений модели. В разд. 1 приводятся основные семантические понятия: спецификация отношений объектов (knowledge specification) – это формальное описание информации о состоянии мира, об эффектах действий и т.п., операция суперпозиции спецификаций, понятия системы, реляционного преобразования и действия. В разд. 2 определяются синтаксис и семантика операторов и рекурсивных определений реляционных преобразований. Приводится пример системы, действия в которой (с побочными эффектами) удобно описывать с помощью рекурсии. В разд. 3 рассматриваются *синтаксические* ограничения на вид операторов реляционных преобразований, достаточные для элиминации кванторов из определений преобразований и ограничения, гарантирующие сохранение конечности спецификаций изменяемых отношений модели. В разд. 4 описывается функция для вычисления табличных преобразований, определенных с использованием рекурсии. Доказываются ее частичная корректность и достаточное условие, гарантирующее завершение вычисления.

1. Спецификации отношений и преобразований. Для описания состояний моделируемой системы и их преобразований используем классическую логику предикатов первого порядка (FOL). Определим основные понятия и обозначения.

Определение 1. *Сигнатура* (*алфавит нелогических символов*) Σ – это кортеж $\langle C, F, F_I, P, P_\Delta, A\rangle$, где C – *бесконечное* множество констант; F – множество функциональных символов; $F_I \subseteq F$ – множество символов *инъективных* функций; P – непустое множество предикатных символов; $P_\Delta \subseteq P$ – непустое конечное множество символов *изменяемых* (*fluent*) отношений; A – непустое конечное множество *символов преобразований*.

Количество аргументов (*арность*) символов $f \in F$, $p \in P$ и $H \in A$ будем обозначать через $|f|$, $|p|$ и $|H|$ соответственно.

Символы преобразований применяются только при описании преобразований (см. разд. 2). Термы и формулы языка логики первого порядка сигнатуры Σ строятся обычным образом с помощью предиката равенства =, связок $\wedge$, $\vee$, $\neg$ и квантора $\exists$. Множества термов и формул сигнатуры Σ будем обозначать через $T(\Sigma)$ и $L(\Sigma)$ соответственно, а количество элементов (*длину*) кортежа термов $\mathbf{t}$ – через $|\mathbf{t}|$.

Определение 2. *Спецификация* (*изменяемых*) *отношений* сигнатуры Σ – это множество формул FOL (литер) вида $p(\mathbf{a})$, $\neg p(\mathbf{a})$, где $p \in P_\Delta$, $\mathbf{a} \in C^{|p|}$.

Множество всех таких литер (*фактов*) обозначим через $F(\Sigma)$, а множество Σ – *спецификаций* $2^{F(\Sigma)}$ – через $SP(\Sigma)$.

На спецификациях как множествах определены обычные операции $\cap$ (пересечение), $\cup$ (объединение), $\setminus$ (разность), а также операции инверсии и суперпозиции.

Определение 3. *Инверсия* Σ-спецификации – это функция $- : SP(\Sigma) \to SP(\Sigma)$, такая, что для всех $\alpha \in SP(\Sigma)$

$$-\alpha = \{p(\mathbf{a}) \mid \neg p(\mathbf{a}) \in \alpha\} \cup \{\neg p(\mathbf{a}) \mid p(\mathbf{a}) \in \alpha\}.$$

Определение 4. *Суперпозиция* Σ-спецификаций – это функция $* : SP(\Sigma) \times SP(\Sigma) \to SP(\Sigma)$, такая, что для всех $\alpha, \beta \in SP(\Sigma)$

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

Семантика символов функций и неизменяемых отношений в описаниях моделируемой системы определяется следущей структурой.

Определение 5. *Система* (*структура*) $\mathfrak{S}$ сигнатуры Σ – это кортеж $\langle \mathcal{I}_F^{\mathfrak{S}}, \mathcal{I}_I^{\mathfrak{S}}, \mathcal{I}_R^{\mathfrak{S}}\rangle$, где $\mathcal{I}_F^{\mathfrak{S}}$ – отображение (*интерпретация функциональных символов*), сопоставляющее каждому символу $f \in F$ функцию $\mathcal{I}_F^{\mathfrak{S}}(f) : SP(\Sigma) \to (C^{|f|} \to C)$, такую, что для всех $f \in F_I$, $\alpha \in SP(\Sigma)$ функция $\mathcal{I}_F^{\mu}(f)(\alpha)$

инъективна; $\mathcal{I}_I^{\mathfrak{S}}$ – отображение (*обратная интерпретация символов инъективных функций*), сопоставляющее каждому символу $f \in \mathrm{F}_I$ функцию $\mathcal{I}_I^{\mathfrak{S}}(f) : \mathrm{SP}(\Sigma) \to (C \to C^{|f|} \cup \{\varnothing\})$, такую, что

$$\mathcal{I}_I^{\mathfrak{S}}(f)(\alpha)(a) = \mathbf{a} \Leftrightarrow \mathcal{I}_F^{\mathfrak{S}}(f)(\alpha)(\mathbf{a}) = a$$

для всех $f \in \mathrm{F}_I$, $\alpha \in \mathrm{SP}(\Sigma)$, $\alpha \in \mathrm{C}$, $\mathbf{a} \in \mathrm{C}^{|f|}$; $\mathcal{I}_R^{\mathfrak{S}}$ – отображение (*интерпретация символов нетабличных отношений*), сопоставляющее каждому символу $p \in \mathrm{P}\backslash\mathrm{P}_T$ функцию $\mathcal{I}_R^{\mathfrak{S}S}(p) : \mathrm{SP}(\Sigma) \to (\mathrm{C}^{|p|} \to \{0, 1\})$.

Множество систем сигнатуры Σ будем обозначать через $\mathfrak{S}(\Sigma)$, а спецификации $\mu \in \mathrm{SP}(\Sigma)$ будем называть (*возможными*) *состояниями* системы.

Дополнительный параметр $\alpha \in \mathrm{SP}(\Sigma)$ у интерпретаций системы $\mathfrak{S} \in \mathfrak{S}(\Sigma)$ позволяет определять функции и отношения, зависящие не только от явных аргументов, но и от состояния системы (см. ниже пример 1).

Далее, для удобства чтения, опускаем упоминание о рассматриваемой системе $\mathfrak{S} \in \mathfrak{S}(\Sigma)$, поскольку всегда будет ясно, о какой системе идет речь.

Определение 6. *Значение* $[\![t]\!]^{\mu} \in \mathrm{C}$ *основного* (не содержащего переменных) терма $t \in \mathrm{T}(\Sigma)$ в состоянии $\mu \in \mathrm{SP}(\Sigma)$ определяется индуктивно следующим образом:

1) $[\![a]\!]^{\mu} = a$, если $a \in \mathrm{C}$;

2) $[\![f(\mathbf{t})]\!]^{\mu} = \mathcal{I}_F^{\mathfrak{S}}(f)(\mu)([\![\mathbf{t}]\!]^{\mu})$, если $f \in \mathrm{F}$, $\mathbf{t} \in T(\Sigma)^{|f|}$.

Для описания (множеств) состояний системы будем использовать следующее отношение.

Определение 7. *Отношение истинности* (*выполнимости*) $\mu \models \phi$ *замкнутой* (не содержащей свободных переменных) формулы $\varphi \in L(\Sigma)$ в состоянии $\mu \in SP(\Sigma)$ определяется индуктивно следующим образом:

1) $\mu \models (t = t')$, если $[\![t]\!]^{\mu} = [\![t']\!]^{\mu}$;

2) $\mu \models p(\mathbf{t})$, если $p \in \mathrm{P}\backslash\mathrm{P}_{\Delta}$, $[\![\mathbf{t}]\!]^{\mu} \in \mathcal{I}_R^{\mathfrak{S}}(p)(\mu)$, $\mu \models_{\mathfrak{S}} p(\mathbf{t})$, если $p \in \mathrm{P}_{\Delta}$, $p([\![\mathbf{t}]\!]^{\mu}) \in \mu$;

3) $\mu \models (\phi \wedge \psi)$, если $\mu \models \phi$ и $\mu \models \psi$;

4) $\mu \models (\phi \vee \psi)$, если $\mu \models \phi$ или $\mu \models \psi$;

5) $\mu \models (\exists x \phi(x))$, если существует элемент $a \in \mathrm{C}$, такой, что $\mu \models \phi(a)$;

6) $\mu \models (\neg\phi)$, если неверно, что $\mu \models \phi$.

Преобразования состояний системы (действия) определим следующим образом.

Определение 8. *Реляционное преобразование* сигнатуры Σ арности k есть отображение

$$r : \mathrm{C}^k \to (\mathrm{SP}(\Sigma) \to \mathrm{SP}(\Sigma)).$$

Множество реляционных преобразований сигнатуры Σ будем обозначать через $\mathrm{RT}(\Sigma)$.

Реляционные преобразования не определяют действия явно, а “вычисляют” спецификации “эффектов” действий, изменения состояний.

Определение 9. *Действие* k-арного реляционного преобразования $r \in \mathrm{RT}(\Sigma)$ – это k-арное реляционное преобразование $\mathrm{App}_r \in \mathrm{RT}(\Sigma)$, такое, что для всех $\mathbf{a} \in \mathrm{C}^k$, для всех $\mu \in \mathrm{SP}(\Sigma)$

$$\mathrm{App}_r(\mathbf{a}, \mu) = \mu * r(\mathbf{a}, \mu).$$

Таким образом, каждое реляционное преобразование *однозначно* определяет действие – преобразование (спецификаций отношений) состояний системы.

2. Определения реляционных преобразований. Опишем синтаксис и семантику операторов и (рекурсивных) определений реляционных преобразований.

Определение 10. Множество $R(\Sigma)$ *операторов* (*правил*) *реляционных преобразований* сигнатуры Σ определяется индуктивно следующим образом:

1) если $p \in \mathrm{P}_{\Delta}$, $\mathbf{t} \in \mathrm{T}(\Sigma)^{|p|}$, то $\{p(\mathbf{t})\} \in \mathrm{R}(\Sigma)$ – (*позитивная*) *литера оператора*;

2) если $\phi \in \mathrm{L}(\Sigma)$ – *атомарная* формула, то $(\phi?) \in \mathrm{R}(\Sigma)$ – *проверка условия*, *тест*;

3) если $\rho \in \mathrm{R}(\Sigma)$ и $\xi \in \mathrm{R}(\Sigma)$, то $(\rho \cap \xi) \in \mathrm{R}(\Sigma)$ – *пересечение*;

4) если $\rho \in \mathrm{R}(\Sigma)$ и $\xi \in \mathrm{R}(\Sigma)$, то $(\rho \cup \xi) \in \mathrm{R}(\Sigma)$ – *объединение*;

5) если $\rho \in \mathrm{R}(\Sigma)$, x – переменная, то $(\cup x\rho) \in \mathrm{R}(\Sigma)$ – *универсальное объединение*, $\cup$ – *квантор объединения*;

6) если $\rho \in \mathrm{R}(\Sigma)$, то $(\sim\rho) \in \mathrm{R}(\Sigma)$ – *дополнение*;

7) если $\rho \in \mathrm{R}(\Sigma)$, то $(\neg\rho) \in \mathrm{R}(\Sigma)$ – *инверсия*;

8) если $H \in \mathrm{A}$, $\mathbf{t} \in \mathrm{T}(\Sigma)^{|H|}$, то $H(\mathbf{t}) \in \mathrm{R}(\Sigma)$ – *применение*, *вызов преобразования*.

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

О п р е д е л е н и е 11. *Интерпретация* символов преобразования сигнатуры Σ есть отображение $J : \mathrm{A} \to \mathrm{RT}(\Sigma)$, сопоставляющее каждому символу преобразования $H \in \mathrm{A}$ $|H|$-арное реляционное преобразование $r \in \mathrm{RT}(\Sigma)$.

Множество интерпретаций $\mathrm{RT}(\Sigma)^{\mathrm{A}}$ символов преобразования сигнатуры Σ будем обозначать через $\mathrm{J}(\Sigma)$. Каждая интерпретация $J \in \mathrm{J}(\Sigma)$ однозначно определяет интерпретацию операторов $\rho \in \mathrm{R}(\Sigma)$.

О п р е д е л е н и е 12. *Значение* $[\![\rho]\!]^{\mu}_{J} \in \mathrm{SP}(\Sigma)$ *замкнутого* (не содержащего свободных переменных) оператора $\rho \in \mathrm{R}(\Sigma)$ в состоянии $\mu \in \mathrm{SP}(\Sigma)$ при интерпретации $J \in \mathrm{J}(\Sigma)$ определяется индуктивно следующим образом:

1) $[\![\{p(\mathbf{t})\}]\!]^{\mu}_{J} = \{p([\![\mathbf{t}]\!]^{\mu})\}$;

2) $[\![\phi?]\!]^{\mu}_{J} = \mathit{if}\ \mu \vDash \phi\ \mathit{then}\ \mathrm{F}(\Sigma)\ \mathit{else}\ \varnothing$;

3) $[\![\rho \cap \xi]\!]^{\mu}_{J} = [\![\rho]\!]^{\mu}_{J} \cap [\![\xi]\!]^{\mu}_{J}$;

4) $[\![\rho \cup \xi]\!]^{\mu}_{J} = [\![\rho]\!]^{\mu}_{J} \cup [\![\xi]\!]^{\mu}_{J}$;

5) $\left[\!\left[\bigcup x\rho(x)\right]\!\right]^{\mu}_{J} = \bigcup_{a \in C} [\![\rho(a)]\!]^{\mu}_{J}$;

6) $[\![\sim\rho]\!]^{\mu}_{J} = \mathrm{F}(\Sigma) \setminus [\![\rho]\!]^{\mu}_{J}$;

7) $[\![\neg\rho]\!]^{\mu}_{J} = -[\![\rho]\!]^{\mu}_{J}$;

8) $H(\mathbf{t})]^{\mu}_{J} = J(H)([\![\mathbf{t}]\!]^{\mu}, \mu)$.

Если значение оператора не зависит от состояния системы или интерпретации, будем опускать соответствующие индексы в обозначении $[\![\rho]\!]^{\mu}_{J}$.

Для часто применяемых операторов удобно использовать следующие обозначения:

$\varnothing = \{p(\mathbf{a})\} \cap \sim \{p(\mathbf{a})\}$ – *пустое множество*, *bot-оператор*, *бот*;

$\top = \sim \varnothing$ – *top-оператор*, *тот*;

$(\phi \wedge \psi)? = \phi? \cap \psi?$ – *проверка конъюнкции*;

$(\phi \vee \psi)? = \phi? \cup \psi?$ – *проверка дизъюнкции*;

$(\exists x\phi)? = \bigcup x(\phi?)$ – *проверка существования*;

$(\neg\phi)? = \sim \phi?$ – *проверка отрицания*;

$\bigcap x\rho = \sim \bigcup x(\sim \rho)$ – *универсальное пересечение*, $\bigcap$ – квантор *пересечения*;

if ϕ *then* ρ *fi* $= \phi? \cap \rho$ – (*сокращенный*) *условный оператор*;

$\{\neg p(\mathbf{t})\} = -\{p(\mathbf{t})\}$ – *негативная литера оператора*;

$\{L_1, \ldots, L_k\} = \bigcup_{i \in 1..k} \{L_i\}$ – *множество литер*.

Семантические свойства операторов будем описывать с помощью следующих отношений. Пусть для любых замкнутых операторов $\rho, \xi \in \mathrm{R}(\Sigma)$, для любого состояния $\mu \in \mathrm{SP}(\Sigma)$

$$\rho \sqsubseteq_{\mu} \xi \Leftrightarrow \forall J \in J(\Sigma) [\![\rho]\!]^{\mu}_{J} \subseteq [\![\xi]\!]^{\mu}_{J},$$

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

Очевидно, что отношение $\sqsubseteq_\mu$ является отношением частичного порядка (*модельный* порядок), а $\sim_\mu$ – соответствующее отношение эквивалентности.

О п р е д е л е н и е 13. Операторы $\rho(\mathbf{x}), \xi(\mathbf{x}) \in \mathrm{R}(\Sigma)$ *равны* (обозначается $\rho(\mathbf{x}) = \xi(\mathbf{x})$), если для всех $\mathbf{a} \in \mathrm{C}^{|\mathbf{x}|}$, для всех $\mu \in \mathrm{SP}(\Sigma)$, $\mathfrak{S} \in \mathfrak{S}(\Sigma)$ $\rho(\mathbf{a}) \sim_\mu \xi(\mathbf{a})$.

Опишем синтаксис и семантику определений.

О п р е д е л е н и е 14. *Система определений реляционных преобразований* сигнатуры Σ – это отображение $\mathfrak{D} : \mathrm{A} \to \mathrm{R}(\Sigma)$, такое, что для всех $H \in \mathrm{A}$ количество свободных переменных оператора $\mathfrak{D}(H)$ не превосходит арности символа преобразования H.

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

$$\{H(\mathbf{x}) = \mathfrak{D}(H)(\mathbf{x}) \,|\, H \in \mathrm{A}\}.$$

Для определения реляционных преобразований будем использовать только *позитивные* реляционные операторы, т.е. такие, которые не содержат применения преобразований в области действия операций дополнения. Множество систем определений из таких операторов сигнатуры Σ будем обозначать через $\mathrm{Def}^+(\Sigma)$.

Приведем пример описания системы, в которой действия (с побочными эффектами) удобно описывать с помощью рекурсии.

П р и м е р 1 (Wagon World).

Система: на линейном (без разъездов и стрелок) железнодорожном пути, неограниченном в обе стороны, находятся вагоны ограниченной грузоподъемности, которые можно двигать, сцеплять или расцеплять соседние вагоны. Около участков пути могут находиться грузы различного веса. Грузы можно загружать в стоящий рядом вагон или выгружать из него. Количество вагонов и грузов конечно.

Сигнатура и ее интерпретация:

$\mathrm{C} = \mathbb{Z}$ – номера участков железнодорожного пути, номера и веса вагонов и грузов;

$\mathrm{P} = \{\leq, At, Near, Linked, Loaded\}$, где

$x \leq y$ – обычный частичный порядок на множестве целых чисел,

$At(x, y)$ – вагон x находится на участке y,

$Near(x, y)$ – груз x находится на участке y,

$Linked(x, y)$ – вагон x сцеплен с вагоном y,

$Loaded(x, y)$ – груз y находится в вагоне x;

$\mathrm{P}_\Delta = \{At, Near, Linked, Loaded\}$;

$\mathrm{F} = \{+, w, w_{\max}, w_{cur}\}$, где

$x + y$ – сложение целых чисел,

$w(x)$ – вес груза x,

$w_{\max}(x)$ – грузоподъемность вагона x,

$w_{cur}(x) = \sum_{Loaded(x,y)\in\mu} w(y)$ – вес грузов в вагоне x, если μ – конечно, иначе -1;

$\mathrm{F}_I = \{\}$;

$$\mathrm{A} = \{Link, UnLink, Load, UnLoad, RShift, LShift\}.$$

Определения действий:

1) сцепление вагонов:

$$\mathrm{Link}(x, y) = \mathit{if}\ \exists z(At(x, z) \wedge (At(y, z-1) \vee At(y, z+1)))\ \mathit{then}\ \{Linked(x, y), Linked(y, x)\}\mathit{fi};$$

2) расцепление вагонов:

$$\mathrm{UnLink}(x, y) = \{\neg Linked(x, y), \neg Linked(y, x)\};$$

3) загрузка вагона x:

$$\mathrm{Load}(x, y) = \mathit{if}\ \exists z(At(x, z) \wedge Near(y, z)) \wedge (w_{cur}(x) + w(y) \leq w_{\max}(x))\ \mathit{then}\ \{Loaded(x, y)\}\mathit{fi};$$

4) разгрузка вагона x:

$$\mathrm{UnLoad}(x,y) = \{\neg Loaded(x,y)\};$$

5) сдвиг вагона вправо:

$$\mathrm{RShift}(x) = \bigcup y(if\ \ At(x,y)\ then\ \{\neg At(x,y), At(x,y+1)\} \quad \text{[вагон вправо]},$$

$$\cup\bigcup z\ if\ \ Loaded(x,z)\ then\ \{\neg Near(z,y), Near(z,y+1)\}\ fi \quad \text{[грузы вправо]},$$

$$\cup\bigcup z\ if\ \ At(z,y+1)\ then\ RShift(z)\ fi \quad \text{[толкает, рекурсия!]},$$

$$\cup\bigcup z\ if\ \ At(z,y-1)\wedge Linked(x,z)\ then\ \mathrm{RShift}(z)\ fi\ fi) \quad \text{[тянет, рекурсия!]};$$

6) сдвиг вагона влево:

$$\mathrm{LShift}(x) = \bigcup y(if\ \ At(x,y)\ then\ \{\neg At(x,y), At(x,y-1)\} \quad \text{[вагон влево]},$$

$$\cup\bigcup z\ if\ \ Loaded(x,z)\ then\ \{\neg Near(z,y), Near(z,y-1)\}\ fi \quad \text{[грузы влево]},$$

$$\cup\bigcup z\ if\ \ At(z,y-1)\ then\ \mathrm{LShift}(z)\ fi \quad \text{[толкает, рекурсия!]},$$

$$\cup\bigcup z\ if\ \ At(z,y+1)\wedge Linked(x,z)\ then\ \mathrm{LShift}(z)\ fi\ fi) \quad \text{[тянет, рекурсия!]}.$$

Семантика системы определений $\mathfrak{D}\in \mathrm{Def}^+(\Sigma)$ определяется через понятие наименьшей неподвижной точки преобразования интерпретаций, соответствующего системе $\mathfrak{D}$.

О п р е д е л е н и е 15. *Преобразование интерпретаций по системе определений* $\mathfrak{D}\in \mathrm{Def}^+(\Sigma)$ есть отображение $\mathrm{JT}_{\mathfrak{D}}: \mathrm{J}(\Sigma)\to \mathrm{J}(\Sigma)$, такое, что

$$\forall J\in J(\Sigma)\ \forall H\in \mathrm{A}\ \forall \mathbf{a}\in C^{|H|}\ \forall \mu\in \mathrm{SP}(\Sigma)[\![H(\mathbf{a})]\!]^{\mu}_{JT_{\mathfrak{D}}(J)} = [\![\mathfrak{D}(H(\mathbf{a}))]\!]^{\mu}_{J}.$$

Неподвижная точка преобразования $\mathrm{JT}_{\mathfrak{D}}$ есть такая интерпретация $J\in \mathrm{J}(\Sigma)$, что $\mathrm{JT}_{\mathfrak{D}}(J)=J$.

Пусть для любых интерпретаций $J, J'\in J(\Sigma)$

$$J\sqsubseteq J' \Leftrightarrow \forall H\in \mathrm{A}\forall \mathbf{a}\in \mathrm{C}^{|H|}\forall \mu\in \mathrm{SP}(\Sigma) J(H)(\mathbf{a},\mu)\subseteq J'(H)(\mathbf{a},\mu).$$

Если каждому символу преобразования $H\in \mathrm{A}$ сигнатуры Σ сопоставлен оператор $\rho_H(\mathbf{x})\in \mathrm{R}(\Sigma)$, то запись $\rho[\forall_{H\in \mathrm{A}}(H\leftarrow \rho_H)]$ будет обозначать оператор – результат замены в операторе $\rho\in \mathrm{R}(\Sigma)$ каждого вхождения применения преобразования вида $H(\mathbf{t})$, $H\in \mathrm{A}$ на вхождение соответствующего оператора $\rho_H(\mathbf{t})$, а запись $\rho[\varnothing]$ – операторы $\rho[\forall_{H\in \mathrm{A}}(H\leftarrow \varnothing)]$.

Применяя теорему Клини о наименьшей неподвижной точке непрерывного функционала (см. подробное обсуждение в [12]), легко доказать [11] следующую теорему.

Т е о р е м а 1 (существование наименьшей неподвижной точки преобразования $\mathrm{JT}_{\mathfrak{D}}$). Пусть задана система позитивных определений $\mathfrak{D}\in \mathrm{Def}^+(\Sigma)$ и для всех $H\in \mathrm{A}$, $i\in \mathbb{N}$

$$F_0=\varnothing, \quad H_{i+1}=\mathfrak{D}(H)[\forall_{H\in \mathrm{A}}(H\leftarrow F_i)].$$

Тогда для всех $H\in \mathrm{A}$, $i\in \mathbb{N}$, для всех $\mathbf{a}\in \mathrm{C}^{|H|}$, для любого состояния $\mu\in \mathrm{SP}(\Sigma)$

$$[\![H_i(\mathbf{a})]\!]^{\mu}\subseteq [\![H_{i+1}(\mathbf{a})]\!]^{\mu}$$

и реляционное преобразование $J_{\mathfrak{D}}\in J(\Sigma)$, такое, что

$$J_{\mathfrak{D}}(H)(\mathbf{a},\mu)=\bigcup_{i\in N}[\![H_i(\mathbf{a})]\!]^{\mu},$$

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

Реляционное преобразование $J_{\mathfrak{D}}(H)\in \mathrm{RT}(\Sigma)$ по системе определений $\mathfrak{D}\in \mathrm{Def}^+(\Sigma)$ будем обозначать через $H_{\mathfrak{D}}$. Хотя для каждого символа преобразования $H\in \mathrm{A}$ преобразование $H_{\mathfrak{D}}$ *всюду определено*, оно не всегда вычислимо.

3. Табличные преобразования. Особый интерес представляют преобразования, сохраняющие конечность описаний состояний системы – спецификаций изменяемых отношений модели. Такие преобразования будем называть *табличными*, так как конечные спецификации отношений могут быть (в принципе) описаны таблицами. Определим класс операторов, в которых значения переменных неявно ограничены табличными отношениями. Множество конечных (*табличных*) Σ-спецификаций будем обозначать через $\mathrm{SP}_T(\Sigma)$, а множество *инъективных* (содержащих только инъективные функциональные символы) термов – через $\mathrm{T}_I(\Sigma)$.

О п р е д е л е н и е 16. Отношения "переменная x *позитивно* (*негативно*) *ограничена* в операторе $\rho \in \mathrm{R}(\Sigma)$" определяются индуктивно следующим образом:

1) любая переменная позитивно ограничена в операторе $\varnothing$;

2) если $p \in P_\Delta$, переменная x имеет вхождение в $t \in \mathrm{T}_I(\Sigma)$ и $t \in \mathbf{t}$, то переменная x позитивно ограничена в операторе $p(\mathbf{t})?$;

3) если переменная x имеет вхождение в $t \in \mathrm{T}_I(\Sigma)$ и все переменные в $t' \in T(\Sigma)$ позитивно ограничены в операторе ρ, то переменная x позитивно ограничена в операторах $(t = t')? \cap \rho$, $(t' = t)? \cap \rho$, $\rho \cap (t = t')?$, $\rho \cap (t' = t)?$;

4) если переменная x позитивно (негативно) ограничена в операторе ρ, то она негативно (позитивно) ограничена в операторе $\sim\rho$;

5) если переменная x позитивно (негативно) ограничена в операторах ρ и ξ, y – другая переменная и $\chi \in \mathrm{R}(\Sigma)$, то переменная x позитивно ограничена в операторах $-\rho$, $\bigcup y\rho$, $\rho \cup \xi$, $\rho \cap \chi$, $\chi \cap \rho$ (негативно ограничена в операторах $-\rho$, $\bigcup y\rho$, $\rho \cap \xi$, $\rho \cup \chi$, $\chi \cup \rho$).

Вхождения переменной в (под)оператор $\rho \in \mathrm{R}(\Sigma)$, в котором она (позитивно или негативно) ограничена, будем называть *ограниченными*.

Очевидно, что эти отношения разрешимы. Для определения и вычисления табличных преобразований важны следующие их свойства.

Пусть для любого оператора $\rho(x, \mathbf{y}) \in \mathrm{R}(\Sigma)$ для любого состояния $\mu \in \mathrm{SP}(\Sigma)$

$$range_x(\rho(x, \mathbf{y}), \mu) \doteq \left\{a \in \mathrm{C} \mid \exists J \in \mathrm{J}(\Sigma) \left[\!\left[\bigcup \mathbf{y}\rho(a, \mathbf{y})\right]\!\right]_J^\mu \neq \varnothing\right\}$$

– μ-*диапазон* переменной x в операторе $\rho(x, \mathbf{y})$.

У т в е р ж д е н и е 1 (операторы с ограниченными переменными). Для любого оператора $\rho(x, \mathbf{y}) \in \mathrm{R}(\Sigma)$, любых $\mathbf{a} \in \mathrm{C}^{|\mathbf{y}|}$, для любого состояния $\mu \in \mathrm{SP}_T(\Sigma)$ выполняется следующее:

1) если переменная x позитивно (негативно) ограничена в операторе $\rho(x, \mathbf{y})$, то множество $range_x(\rho(x, \mathbf{y}), \mu)$ ($range_x(\sim \rho(x, \mathbf{y}), \mu)$) конечно;

2) если переменная x позитивно ограничена в операторе $\rho(x, \mathbf{y})$, то для любого конечного множества $C' \subseteq C$, такого, что $range_x(\rho(x, \mathbf{y}), \mu) \subseteq C'$,

$$\bigcup x\rho(x, \mathbf{a}) \sim_\mu \bigcup_{a \in C'} \rho(a, \mathbf{a});$$

3) если переменная x негативно ограничена в операторе $\rho(x, \mathbf{y})$, то

$$\bigcup x\rho(x, \mathbf{a}) \sim_\mu \top.$$

Д о к а з а т е л ь с т в о. 1. Базис и шаги индукции очевидны. Например, если множество $range_x(\sim\rho(x, \mathbf{y}), \mu)$ конечно, то и множество $range_x(\sim \bigcup y\rho(x, \mathbf{y}), \mu)$ конечно, так как

$$range_x\left(\sim\bigcup y\rho(x, \mathbf{y}), \mu\right) = range_x\left(\bigcap y \sim \rho(x, \mathbf{y}), \mu\right) \subseteq range_x(\sim\rho(x, \mathbf{y}), \mu).$$

2. Следует из п. 1.

3. Следует из п. 1 и того, что множество констант C – бесконечное, а значит, существуют $a \in (\mathrm{C} \backslash range_x(\rho(x, \mathbf{y}), \mu))$, такие, что $\left[\!\left[\bigcup \mathbf{y}\rho(a, \mathbf{y})\right]\!\right]_J^\mu = \varnothing$.

Операторы $\rho \in \mathrm{R}(\Sigma)$, в которых во всех подоператорах вида $\bigcup x\xi(x, \mathbf{y})$ переменная x (позитивно или негативно) ограничена в операторе $\xi(x, \mathbf{a})$ для произвольного $\mathbf{a} \in \mathrm{C}^{|\mathbf{y}|}$, будем называть опе-

раторами *с* (*таблично*) *ограниченными кванторами*, а те из них, в которых все вхождения переменных в применения преобразований оператора ограничены, – операторами с *ограниченными применениями преобразований* (*вызовами*).

Используя утверждение 1, легко доказать следующее утверждение.

У т в е р ж д е н и е 2 (элиминация кванторов). Пусть $\rho \in \mathrm{R}(\Sigma)$ – замкнутый оператор с ограниченными кванторами. Тогда для любого состояния $\mu \in \mathrm{SP}_T(\Sigma)$ существует бескванторный оператор $\mathrm{ELIM}(\rho,\mu) \in \mathrm{R}(\Sigma)$, такой, что $\rho \sim_\mu \mathrm{ELIM}(\rho,\mu)$.

Значение оператора может быть бесконечным из-за наличия в нем операций проверки условия, дополнения, универсального объединения и рекурсии. Для определения табличных преобразований будем использовать операторы, в которых эти операции и рекурсия ограничены следующим образом.

О п р е д е л е н и е 17. Отношение "оператор $\rho \in \mathrm{R}(\Sigma)$ является *табличным* оператором в (при) системе определений $\mathfrak{D} \in \mathrm{Def}_T^+(\Sigma)$" определяется индуктивно следующим образом:

1) оператор $\varnothing$ является табличным;

2) операторы $\{p(\mathbf{t})\} \in \mathrm{R}(\Sigma)$ являются табличными;

3) если операторы ρ и ξ – табличные и $\chi \in \mathrm{R}(\Sigma)$, то операторы $-\rho$, $\rho \cup \xi$, $\rho \cap \chi$, $\chi \cap \rho$ являются табличными;

4) если оператор ρ – табличный и переменная x позитивно ограничена в операторе ρ, то оператор $\bigcup x\rho$ является табличным;

5) если $H \in \mathrm{A}$ и оператор $\mathfrak{D}(H)$ – табличный, то операторы $H(\mathbf{t}) \in \mathrm{R}(\Sigma)$ являются табличными.

Очевидно, что это отношение также разрешимо.

Все операторы системы определений, описанной в примере 1, являются табличными операторами с ограниченными вызовами.

Используя утверждение 1 и теорему 1, легко доказать следующее утверждение.

У т в е р ж д е н и е 3 (табличные преобразования). Пусть задана система определений $\mathfrak{D} \in \mathrm{Def}^+(\Sigma)$, $H \in \mathrm{A}$ и оператор $\mathfrak{D}(H)$ является табличным в системе $\mathfrak{D}$. Тогда для любых $\mathbf{a} \in \mathrm{C}^{|H|}$, для любого состояния $\mu \in \mathrm{SP}_T(\Sigma)$

$$H_{\mathfrak{D}}(\mathbf{a},\mu) \in \mathrm{SP}_T(\Sigma).$$

Итак, синтаксические, легко проверяемые ограничения на вид операторов гарантируют, что преобразования, определяемые табличными операторами, являются табличными.

4. Вычисление табличных преобразований. Опишем один из возможных способов вычисления табличных преобразований, определенных рекурсивно.

Для вычисления будем использовать (*частично вычисленные*) *конъюнкты* – это операторы $\rho \in \mathrm{R}(\Sigma)$ вида τ или $\tau \cap \pi_1 \cap \ldots \cap \pi_k$, где

τ – *коэффициент* конъюнкта, оператор, построенный из литер вида $\{P(\mathbf{a})\}$, где $\mathbf{a} \in \mathrm{C}^{|P|}$, с помощью операций $\cap, \cup, \sim, -$;

π_i – *варианты применений преобразований* конъюнкта, попарно различные операторы вида $H(\mathbf{a})$, $-H(\mathbf{a})$, где $\mathbf{a} \in \mathrm{C}^{|H|}$.

Множество всех таких конъюнктов будем обозначать через $\mathrm{K}(\Sigma)$, а коэффициент конъюнкта $\kappa \in \mathrm{K}(\Sigma)$ и *множество* его вариантов применений преобразований – через $\tau(\kappa)$ и $\pi(\kappa)$ соответственно.

Операторы вида $\kappa_1 \cup \ldots \cup \kappa_n$, где $\kappa_i \in \mathrm{K}(\Sigma)$, будем называть *дизъюнктивными*.

Применяя тождества алгебры множеств, легко доказать следующее утверждение.

У т в е р ж д е н и е 4 (дизъюнктивная нормальная форма). Пусть $\rho \in \mathrm{R}(\Sigma)$ – замкнутый позитивный бескванторный оператор. Тогда для любого состояния $\mu \in \mathrm{SP}_T(\Sigma)$ существует дизъюнктивный оператор $\mathrm{DNF}(\rho,\mu) \in \mathrm{R}(\Sigma)$, такой, что $\rho \sim_\mu \mathrm{DNF}(\rho,\mu)$.

Будем говорить, что дизъюнктивный оператор $\kappa_1 \cup \ldots \cup \kappa_n \in \mathrm{R}(\Sigma)$ является *минимальным*, если для всех $i, j \in 1..n$ выполняются следующие условия:

1) если $[\![\tau(\kappa_i)]\!] = \varnothing$, то $\pi(\kappa_i) = \varnothing$;

2) если $[\![\tau(\kappa_i)]\!] \subseteq [\![\tau(\kappa_j)]\!]$ и $\pi(\kappa_j) \subseteq \pi(\kappa_i)$, то $i = j$;

3) если $\pi(\kappa_i) = \pi(\kappa_j)$, то $i = j$.

Утверждение 5 (минимизация). Пусть $\rho \in R(\Sigma)$ – дизъюнктивный оператор. Тогда существует минимальный дизъюнктивный оператор $MIN(\rho) \in R(\Sigma)$, такой, что $\rho = MIN(\rho)$.

Для представления дизъюнктивных операторов и операций с ними удобно использовать множества конъюнктов. Множество всех *конечных* подмножеств множества $K(\Sigma)$ будем обозначать через $\mathcal{P}_{K(\Sigma)}$.

Каждому $\Gamma \in \mathcal{P}_{K(\Sigma)}$ соответствует единственный (с точностью до порядка конъюктов, который не влияет на значение) дизъюнктивный оператор

$$\cup\Gamma = \mathit{if}\ \Gamma = \varnothing\ \mathit{then}\ \varnothing\ \mathit{else} \bigcup_{\kappa\in\Gamma} \kappa.$$

Если $\Gamma, \Delta \in \mathcal{P}_{K(\Sigma)}$, то вместо $\cup\Gamma \sqsubseteq_\mu \cup\Delta$ будем писать $\Gamma \sqsubseteq_\mu \Delta$ и аналогично для отношений $\sim_\mu$ и $=$.

Функция TT–CALC, определяемая ниже, является модификацией функции RT–CALC [11], проще и не требует разрешимости отношения $\mu \models \phi$ для вычисления.

Пусть для любой системы определений $\mathfrak{D} \in \mathrm{Def}^+_T(\Sigma)$, для любых $\Gamma, \Delta \in \mathcal{P}_{K(\Sigma)}$ и любого состояния $\mu \in SP_T(\Sigma)$

$$\text{TT–CALC}(\Gamma, \Delta) = \mathit{if}\ \Delta = \varnothing\ \mathit{then} \cup\Gamma[\varnothing]\ \mathit{else}\text{TT–CALC}(MIN(\Gamma \cup \Delta),$$
$$CUT(MIN(\Gamma \cup \Delta), NEW(\Delta)),$$

где

$NEW(\Delta) = \bigcup_{\kappa\in\Delta'} DNF(ELIM(\kappa))$ – вычисление новых конъюнктов;

$\Delta = \{\kappa[\forall_{H\in A}(H \leftarrow \mathfrak{D}(H))] \mid \kappa \in \Delta\}$ – подстановка определений в конъюнкты;

$CUT(\Gamma, \Delta) = \{\kappa \in \Delta \mid \neg\exists\kappa' \in \Gamma\, \kappa \sqsubseteq_\mu \kappa'\}$ – сокращение “бесполезных” конъюнктов;

ELIM, DNF, MIN – функции, описанные в утверждениях 2, 4 и 5 соответственно.

Интуитивно, Γ есть множество (частично) вычисленных конъюнктов, а Δ – множество конъюнктов, которые “еще имеет смысл” вычислять (делать подстановки).

При доказательстве корректности функции TT–CALC будем использовать следующие очевидные свойства.

Утверждение 6 (свойства функций в TT–CALC). Для любой системы определений $\mathfrak{D} \in \mathrm{Def}^+(\Sigma)$, для любых $\Gamma, \Delta \in \mathcal{P}_{K(\Sigma)}$ и любого состояния $\mu \in SP_T(\Sigma)$ выполняется:

1) $\Gamma' = \Gamma \cup \Gamma'$;

2) $\Gamma' \sim_\mu NEW(\Gamma)$;

3) если $\Gamma \sim_\mu \Delta$, то $\Gamma' \sim_\mu \Delta'$;

4) $\Gamma \cup \Delta = \Gamma \cup CUT(\Gamma, \Delta)$.

Для вычисления функций в TT–CALC достаточно, чтобы система $\mathfrak{S} \in \mathfrak{S}(\Sigma)$ была определена *эффективно*, т.е. чтобы для всех $f \in F$, $g \in F_I$, $p \in P \backslash P_\Delta$ функции $\mathcal{I}^{\mathfrak{S}}_F(f)$, $\mathcal{I}^{\mathfrak{S}}_I(g)$, $\mathcal{I}^{\mathfrak{S}}_R(p)$ были вычислимы и определены на множестве $SP_T(\Sigma)$.

Теорема 2 (корректность функции TT–CALC). Пусть $\mathfrak{D} \in \mathrm{Def}^+(\Sigma)$ – система определений из операторов с ограниченными кванторами. Тогда для любых $H \in A$, $\mathbf{a} \in C^{|H|}$ и любого состояния $\mu \in SP_T(\Sigma)$ выполняется:

1) если вычисление TT–CALC$(\{\}, \{H(\mathbf{a})\})$ заканчивается, то

$$H_{\mathfrak{D}}(\mathbf{a}, \mu) = [\![\text{TT–CALC}(\{\}, \{H(\mathbf{a})\})]\!];$$

2) если $\mathfrak{D}$ – система из операторов с ограниченными применениями преобразований, то вычисление TT–CALC$(\{\}, \{H(\mathbf{a})\})$ заканчивается.

Доказательство. 1. Вычислению TT–CALC($\{\},\{H(\mathbf{a})\}$) соответствуют последовательности Γ_i, Δ_i, $i \in \mathbb{N}$, такие, что

$$\Gamma_0 = \{\}, \quad \Delta_0 = \{H(\mathbf{a})\}, \quad \Gamma_{i+1} = \mathrm{MIN}(\Gamma_i \cup \Delta_i), \quad \Delta_{i+1} = \mathrm{CUT}(\Gamma_{i+1}, \mathrm{NEW}(\Delta_i)),$$

где Γ_i, Δ_i – аргументы функции TT–CALC перед выполнением $i+1$ шага вычисления.

Докажем, что для всех $i > 0$ $\Gamma_i' \sim_\mu \Gamma_{i+1}$.

Базис индукции:

$$\Gamma_1' = \Gamma_1 \cup \Gamma_1 = \Gamma_1 \cup \Delta_0' \sim_\mu \Gamma_1 \cup \mathrm{NEW}(\Delta_0) =$$
$$= \Gamma_1 \cup \mathrm{CUT}(\Gamma_1, \mathrm{NEW}(\Delta_0)) = \Gamma_1 \cup \Delta_1 = \mathrm{MIN}(\Gamma_1 \cup \Delta_1) = \Gamma_2.$$

Шаг индукции:

$$\Gamma_{i+1}' = \mathrm{MIN}(\Gamma_i \cup \Delta_i)' = (\Gamma_i \cup \Delta_i)' = \Gamma_i' \cup \Delta_i' \sim_\mu (\Gamma_i \cup \Delta_i) \cup \Delta_i' \sim$$
$$\sim_\mu \mathrm{MIN}(\Gamma_i \cup \Delta_i) \cup \mathrm{NEW}(\Delta_i) = \Gamma_{i+1} \cup \mathrm{NEW}(\Delta_i) =$$
$$= \Gamma_{i+1} \cup \mathrm{CUT}(\Gamma_{i+1}, \mathrm{NEW}(\Delta_i)) = \Gamma_{i+1} \cup \Delta_{i+1} = \mathrm{MIN}(\Gamma_{i+1} \cup \Delta_{i+1}) = \Gamma_{i+2}.$$

Из теоремы 1 следует, что

$$H_{\mathfrak{D}}(\mathbf{a}, \mu) = \bigcup_{i \in N} [\![\cup \Gamma_i[\varnothing]]\!].$$

Если $\Delta_i = \varnothing$ для некоторого $i \in \mathbb{N}$, то $H_{\mathfrak{D}}(\mathbf{a}, \mu) = [\![\cup \Gamma_i[\varnothing]]\!]$ и вычисление можно закончить.

2. Допустим, что $\Delta_i \neq \varnothing$ для всех $i \in \mathbb{N}$. Пусть $\Delta = \bigcup_{i \in N} \Delta_i$, а *calls*, *lits* и *vals* – множества всех применений преобразований, литер и значений коэффициентов в конъюнктах $\kappa \in \Delta$ соответственно.

Эти множества являются конечными, *calls* – так как все операторы системы являются операторами с ограниченными вызовами, *lits* – так как для каждого $H(\mathbf{a}) \in$ *calls* множество литер в операторе $\mathrm{ELIM}(\mathfrak{D}(H)(\mathbf{a}))$ конечно, *vals* – так как коэффициенты конъюнктов являются булевой комбинацией литер из *lits*.

Поскольку у любого конъюнкта $\kappa \in \Delta$ $[\![\tau(\kappa)]\!] \in$ *vals* и $\pi(\kappa) \in 2^{calls}$, т.е. являются элементами конечных множеств, то найдутся такие $i, j \in \mathbb{N}$, $\kappa_i \in \Delta_i$, $\kappa_j \in \Delta_j$, что $i < j$ и $\kappa_i \sim_\mu \kappa_j$.

Это противоречит определению, согласно которому $\kappa_i \sqsubseteq_\mu \Gamma_j$ и $\kappa_j \not\sqsubseteq_\mu \Gamma_j$, а значит, существует наименьшее $i \in \mathbb{N}$, такое, что $\Delta_i = \varnothing$ и вычисление заканчивается на шаге $i+1$.

Покажем процесс вычисления реляционного оператора в системе из примера 1.

Пример 2 (Wagon World, вычисление оператора). Вычислим $\mathrm{RShift}(2)(\mu)$ в системе, описанной в примере 1, в состоянии

$$\mu = \{At(1,-1), At(2,0), At(3,1), Linked(1,2), Linked(2,1), Loaded(1,1), Near(1,-1),$$
$$Loaded(3,2), Near(2,1)\}.$$

Итак, $\Gamma_0 = \{\}$, $\Delta_0 = \{\mathrm{RShift}(2)\}$.

Шаг 1. Так как $\Delta_0 \neq \varnothing$, вычисляем

$$\Gamma_1 = \mathrm{MIN}(\Gamma_0 \cup \Delta_0) = \{\mathrm{RShift}(2)\};$$
$$\mathrm{NEW}(\Delta_0) = \{\{\neg At(2,0), At(2,1)\}, \mathrm{RShift}(3), \mathrm{RShift}(1)\};$$
$$\Delta_1 = \mathrm{CUT}(\Gamma_1, \mathrm{NEW}(\Delta_0)) = \mathrm{NEW}(\Delta_0).$$

Шаг 2. Так как $\Delta_1 \neq \varnothing$, вычисляем

$$\Gamma_2 = \{\mathrm{RShift}(2), \mathrm{RShift}(3), \mathrm{RShift}(1), \{\neg At(2,0), At(2,1)\}\};$$
$$\mathrm{NEW}(\Delta_1) = \{\{\neg At(2,0), At(2,1)\}, \{\neg At(3,1), At(3,2)\}, \{\neg Near(2,1), Near(2,2)\},$$
$$\{\neg At(1,-1), At(1,0)\}, \{\neg Near(1,-1), Near(1,0)\}, \mathrm{RShift}(2)\};$$
$$\Delta_2 = \{\{\neg At(3,1), At(3,2)\}, \{\neg Near(2,1), Near(2,2)\}, \{\neg At(1,-1), At(1,0)\},$$

$$\{\neg Near(1,-1), Near(1,0)\}\}.$$

Здесь в $\mathrm{NEW}(\Delta_1)$ функцией CUT сокращены конъюнкты RShift(2) и $\{\neg At(2,0), At(2,1)\}$.

Ш а г 3. Так как $\Delta_2 \neq \varnothing$, вычисляем

$$\Gamma_3 = \{\mathrm{RShift}(2), \mathrm{RShift}(3), \mathrm{RShift}(1), \neg At(2,0), At(2,1), \neg At(3,1), At(3,2),$$
$$\neg Near(2,1), Near(2,2), \neg At(1,-1), At(1,0), \neg Near(1,-1), Near(1,0)\};$$

$\mathrm{NEW}(\Delta_2) = \Delta_2$, так как $\Delta_2' = \Delta_2$;

$$\Delta_3 = \varnothing.$$

Здесь в $\mathrm{NEW}(\Delta_2)$ функцией CUT сокращены все конъюнкты.

Ш а г 4. Так как $\Delta_3 = \varnothing$, заканчиваем вычисление.

Результатом вычисления является спецификация эффектов действия:

$$\mathrm{RShift}(2)(\mu) = \{\neg At(2,0), At(2,1), \neg At(3,1), At(3,2), \neg Near(2,1), Near(2,2)\},$$
$$\neg At(1,-1), At(1,0), \neg Near(1,-1), Near(1,0)\},$$

а результатом соответствующего действия $\mathrm{App}_{\mathrm{RShift}}(2,\mu)$ – состояние

$$\mu' = \{At(1,0), At(2,1), At(3,2), Linked(1,2), Linked(2,1), Loaded(1,1), Near(1,0),$$
$$Loaded(3,2), Near(3,2), \neg At(1,-1), \neg At(2,0), \neg At(3,1), \neg Near(1,-1), \neg Near(2,1)\}.$$

Заключение. Итак, полученные результаты позволяют описывать системы действий как системы рекурсивно определенных реляционных преобразований и вычислять эффекты действий с помощью теоретико-множественных операций.

Применение рекурсии позволяет учитывать причинно-следственные связи, существующие в моделируемой системе, и описывать вместе с самим действием его побочные эффекты – реакцию системы на действие. Это делает язык KSL не менее выразительным, чем логические языки описания действий (см., например, [13, 14]), сохраняя при этом возможности планирования, присущие STRIPS-like языкам.

Чисто синтаксические, легко проверяемые ограничения на вид операторов в системах табличных преобразований, рассмотренные в статье, гарантируют, что определяемые ими преобразования сохраняют конечность спецификаций изменяемых отношений модели, а предложенный алгоритм их вычисления всегда завершает работу.

Приведенный пример демонстрирует выразительные возможности рекурсивных определений табличных реляционных преобразований и показывает, на наш взгляд, перспективность данного подхода к описанию действий в моделируемых системах.

СПИСОК ЛИТЕРАТУРЫ

1. *Harmelen F., Lifschitz V., Porter B.* (eds.). Handbook of Knowledge Representation. Elsevier, 2008. P. 1034.
2. *Ghallab M., Nau D., Traverso P.* Automated Planning: Theory & Practice. Elsevier, 2004. P. 635.
3. *Ghallab M., Nau D., Traverso P.* Automated Planning and Acting. Cambridge University Press, 2016. P. 451. http://projects.laas.fr/planning/.
4. *Fikes R., Nilsson N.* STRIPS: A new Approach to the Application of Theorem Proving to Problem Solving // Artificial Intelligence. 1971. V. 2. № 3–4. P. 189–208. https://doi.org/10.1016/0004-3702(71)90010-5
5. *Pednault E.P. D.* ADL: Exploring the Middle Ground Between STRIPS and the Situation Calculus // Proc. First Intern. Conf. on Principles of Knowledge Representation and Reasoning. San Francisco. 1989. P. 324–332.
6. *Pednault E.P. D.* ADL and the State-Transition Model of Action // J. Log. Comput. 1994. V. 4. № 5. P. 467–512. https://doi.org/10.1093/logcom/4.5.467
7. *Ghallab M., Howe A., Knoblock C., McDermott D., Ram A., Veloso M., Weld D., Wilkins D.* PDDL – the Planning Domain Definition Language. Technical Report CVC TR98003/DCS TR1165. New Haven, CT: Yale Center for Computational Vision and Control. 1998. P. 26. ftp://ftp.cs.yale.edu/pub/mcdermott/software/pddl.tar.gz
8. *Pellier D., Fiorino H.* PDDL4J: a Planning Domain Description Library for Java // J. of Experimental and Theoretical Artificial Intelligence. 2018. V. 30 (1). P. 143–176. https://hal.archives-ouvertes.fr/hal-01731542.

9. *Кучуганов М.В.* Системы реляционных преобразований: правила и критерий реализуемости // Вестн. Удмуртского ун-та. Математика. Механика. Компьютерные науки. 2015. Т. 25. № 1. С. 117–125. http://vst.ics.org.ru/journal/article/2255.
10. *Claßen J., Lakemeyer G.* A Semantics for ADL as Progression in the Situation Calculus // Proc. 11th Intern. Workshop on Non-Monotonic Reasoning, Clausthal-Zellerfeld, 2006. P. 334–341. http://www.in.tu-clausthal.de/uploads/media/NMR_Proc_TR4.pdf.
11. *Кучуганов М.В.* Рекурсивные определения реляционных преобразований // Программные системы: теория и приложения. 2018. Т. 9. № 1. С. 53–83. https://doi.org/10.25209/2079-3316-2018-9-1-53-83
12. *Stoltenberg-Hansen V., Lindstrom I., Griffor E.R.* Mathematical Theory of Domains. Cambridge University Press. 2008. P. 364. https://doi.org/10.1017/CBO9781139166386.
13. *Gelfond M., Lifschitz V.* Action Languages // Electronic Transactions on Artificial Intelligence. 1998. V. 3. P. 195–210.
14. *Inclezan D., Gelfond M.* Modular Action Language ALM // Theory and Practice of Logic Programming. 2016. V. 16 (2). P. 189–235. https://doi.org/10.1017/S1471068415000095