УДК 65.011.56:519.23:616

ИНДУКТИВНАЯ МОДЕЛЬ ОПИСАНИЯ СИТУАЦИЙ ДЕЙСТВИТЕЛЬНОСТИ

Ю.А. Прокопчук
Украинский государственный химико-технологический университет,
Институт технической механики НАНУ и НКАУ
itk3@ukr.net

У роботі побудована індуктивна модель, що формалізує специфічні типи способів з'єднання або зв'язування матеріальних проявів і яка відповідає миру якісного досвіду.
Ключові слова: елементарні тести, графи доменів, події, ситуації дійсності, граничні узагальнення

In work the inductive model which corresponds to qualitative experience is constructed
Keywords: elementary tests, graph of domains, events, situation of reality, limiting generalizations

В работе построена индуктивная модель, формализующая специфические типы способов соединения или связывания материальных проявлений, которые соответствуют миру качественного опыта.
Ключевые слова: элементарные тесты, графы доменов, события, ситуации действительности, предельные обобщения

Введение

Одной из главных на сегодняшний день проблем при разработке систем искусственного интеллекта остается разработка единых методов, подходов к описанию объектов и процессов реального мира. Наличие хотя бы приблизительной, первоначальной модели сделало бы возможным создание когнитивной структуры, охватывающей все объекты и явления с единой фрактальной, многомерной способностью отображения информации [1 - 4].

В работе предпринимается попытка выявить и формализовать те довольно специфические типы способов соединения или связывания материальных проявлений, которые соответствуют миру качественного опыта. Предложенная модель очерчивает путь перехода через барьер между количеством и качеством или между физической и качественной модальностями материальных и информационных проявлений.

С практической точки зрения предложенный формализм служит методологической основой метода предельных обобщений [3].

1. Элементарные тесты. Ориентированные графы доменов

Пусть $\{\tau\}$ - множество элементарных тестов, с помощью которых описываются все факторы, обстоятельства и явления, имеющие отношение к изучаемой ситуации действительности. Элементарность теста означает, что его результат может быть представлен в виде: «тест = значение» или в эквивалентном виде «тест? значение». Мощность множества всех элементарных тестов совпадает с мощностью всех односложных вопросов, так как любой такой вопрос представим в виде «вопрос? ответ».

Конкретный результат теста τ будем обозначать через $\underline{\tau}$. Результаты тестов могут выбираться (формироваться) из разных доменов (множеств значений). Для фиксации того, что в качестве множества результатов теста τ используется домен T, будем использовать нотацию: τ/T. Примеры нотаций: a/A, $\{b/B\}$ – множество тестов, $\{c\}/C$ – множество однотипных тестов, $\{\underline{a}/A\}$ – множество значений группы тестов.

Используя разные домены, можно управлять общностью (масштабом) результата одного и того же теста []. Во многих случаях между доменами разного уровня общности могут быть заданы однозначные правила пересчета значений. Правила пересчета значений теста из одного домена в другой задают *ориентированный граф доменов* $G(\tau) \equiv \{T \to T'\}_{\tau}$, а N_{τ} – общее количество доменов в графе $G(\tau)$. Домен слева от стрелки в связке ($T \to T'$) будем называть *доменом – предком*, а справа от стрелки *доменом – наследником*. Соответственно любую связку ($T \to T'$) можно именовать «предок - наследник». Совокупная смысловая область элементов домена – наследника полностью совпадает с совокупной смысловой областью элементов домена – предка.

Правила $\{T \to T'\}_{\tau}$ задают, по – сути, *эмпирическую аксиоматику*, а также систему интерпретаций в рамках конкретной задачи (класса задач) предметной области.

Поскольку у любого теста может быть несколько графов доменов, то в некоторых случаях вместо нотации τ/T можно применять расширенную нотацию $\tau/G{:}T$.

Без ограничений общности положим, что домены графа $G(\tau)$ состоят из альтернативных элементов (точечных или атомарных элементов) и атомарный элемент любого домена однозначным образом преобразуется в единственные атомарные элементы доменов – наследников (если они существуют). Для любого атомарного элемента a из домена A автоматически определен элемент (и терм) «*не* a» ≡ «$\neg a$» $= A \setminus a$. То же справедливо и для любого подмножества $A' \subset A$: автоматически определены элементы – термы «*не* A'» ≡ «$\neg A'$» $= A \setminus A'$. Таким образом, операция отрицания действует в рамках конкретного домена.

Примем, что для любой связки $\{T \to T'\}$ выполняется соотношение: $|T| > |T'|$, где $|*|$ - мощность множества.

Ориентированный граф имеет одну *базовую вершину* – базовый домен со значениями (элементами) максимально высокого уровня точности. В базовую вершину не входит ни одна дуга графа. Любой домен графа проецируется на весь базовый домен. Ориентированный граф не имеет циклов. Если имеются разные пути перехода к какой-либо вершине, то эти пути должны приводить к одному и тому же результату (не должно быть конфликтов). *Терминальные вершины* или вершины, из которых не выходит ни одна дуга, задают домены максимальных уровней общности. В графе все вершины уникальны. Если базовый домен непрерывный, то любой домен наследник является дискретным.

Дискретный домен максимального уровня общности (терминальная вершина графа) называется *N-арным конструктом* теста, где N – число элементов домена. Примеры бинарных конструктов: {Истина; Ложь}, {Норма: Отклонение}, {Эффективно; Неэффективно}, {Жалобы есть; Жалоб нет}, {Благоприятный; Неблагоприятный} и т.д. Для одного и того же теста может быть задано потенциально сколь угодно много N-арных конструктов. Между базовым доменом теста и N-арным конструктом также может быть задано сколь угодно много промежуточных доменов. Необходимо стремиться к тому, чтобы в любом графе $G(\tau)$ все терминальные вершины являлись бинарными конструктами. Очевидно, граф может состоять из одной базовой вершины, например: Пол/ {М; Ж}. Графы всех тестов, имеющих дихотомические значения, состоят из одной вершины.

На алгоритм преобразования (обобщения) значений доменов могут влиять значения других тестов. Пусть $(T \rightarrow T') \in G(\tau)$, а внешние тесты $\{a/A\}$ влияют на алгоритм пересчета значений из домена T в домен T'. Для фиксации данного факта будем использовать нотацию: $T\ \ \{a/A\} \rightarrow T'$. Если результат какого либо теста из $\{a/A\}$ неизвестен, то будем считать, что и преобразование $T\,\{a/A\} \rightarrow T'$ не определено, следовательно, не определен весь подграф с базовой вершиной T'. Таким образом, граф $G(\tau\,|\{a/A\})$, где $\{a/A\}$ – множество всех используемых внешних тестов, имеет переменную структуру.

Правила вычислений элементов некоторых доменов на основе значений других тестов, в общем случае, имеют вид: f/μ: $\underline{\tau}/T$, $\{\underline{a}/A\} \rightarrow \underline{\tau}/T'$, где f/μ – некоторая однозначная функция (μ - механизм реализации функции). Данные правила будем относить к *онтологическим соглашениям*, так как сами графы принадлежат онтологии (задачи, предметной области). Заметим, что правила пересчета значений одних тестов в значения других тестов содержатся также в базе знаний.

Примеры доменов для теста τ = ‘Возраст’:
B1 = [0; 100]; B2 = {юный; молодой; средних лет; пожилой; старческий}; B3 = {молодой; средних лет; пожилой}.
Примеры результатов тестов:

Возраст/ B2 = «пожилой»; Возраст/ B1? 87; Возраст/ B3 = «пожилой».

Отметим, что (Возраст/ B2? пожилой) $\neq$ (Возраст/ B3? пожилой).

Зададим правила пересчета между доменами B1, B2 и B3 следующим образом:
B1.[0; 14] $\rightarrow$ B2.{юный}; B1.[15; 33] $\rightarrow$ B2.{молодой}; B1.[34; 55] $\rightarrow$ B2.{средних лет}; B1.[56; 70] $\rightarrow$ B2.{пожилой}; B1.[71; 100] $\rightarrow$ B2.{старческий}; B2.{юный; молодой} $\rightarrow$ B3.{молодой}; B2.{средних лет} $\rightarrow$ B3.{средних лет}; B2.{пожилой; старческий} $\rightarrow$ B3.{пожилой}.

Правила пересчета {B1 $\rightarrow$ B2; B2 $\rightarrow$ B3} задают ориентированный граф доменов для теста ‘Возраст’. Обозначим граф через G_1(Возраст).

Если задан граф доменов теста, то появляется возможность автоматического вычисления значений теста в рамках всех доменов по одному, например, самому точному значению. Примеры:

Возраст/ В1? 5 $\rightarrow$ Возраст/ В2? юный $\rightarrow$ Возраст/ В3? молодой;

Возраст/ В1? 76 $\rightarrow$ Возраст/ В2? старческий $\rightarrow$ Возраст/ В3? пожилой.

Преобразование множества значений:

Возраст/ В1? [13; 20] $\rightarrow$ Возраст/ В2? {юный; молодой} $\rightarrow$ Возраст/ В3? молодой.

Результат получен путем объединения преобразований атомарных значений.

Зададим другой набор доменов для того же теста τ = 'Возраст':

В1 = [0; 100]; В4 = {[0; 17]; [18; 30]; [31; 40] [41; 55]; [56; 60]; [61; 100]}; В5 = {молодой; средних лет; пожилой}; В6 = {трудоспособный; нетрудоспособный}. Правила пересчета значений из домена В1 в домен В4 очевидны. Зададим остальные правила (с учетом разного возраста ухода на пенсию мужчин и женщин: мужчины в 60 лет, женщины в 55):

В4.{[0; 17]; [18; 30]} $\rightarrow$ В5.{молодой}; В4.{[31; 40] [41; 55]} $\rightarrow$ В5.{средних лет}; В4.{[56; 60]; [61; 100]} $\rightarrow$ В5.{пожилой};

Пол? М & В4.{ [18; 30]; [31; 40] [41; 55]; [56; 60]} $\rightarrow$ В6.{трудоспособный};

Пол? Ж & В4.{ [18; 30]; [31; 40] [41; 55]} $\rightarrow$ В6.{трудоспособный};

Пол? М & В4.{[0; 18]; [61; 100]} $\rightarrow$ В6.{нетрудоспособный};

Пол? Ж & В4.{[0; 18]; [56; 60]; [61; 100]} $\rightarrow$ В6.{нетрудоспособный}.

В итоге получаем второй граф G_2(Возраст | {Пол}) $\equiv$ {В1 $\rightarrow$ В4; В4 $\rightarrow$ В5; В4 {Пол}$\rightarrow$ В6}. Обратим внимание, что, несмотря на внешнее совпадение термов доменов В3 и В5, это разные домены с отличными по смыслу термами. Следует также отметить, что если значение теста «Пол» не задано, то связка (В4 {Пол}$\rightarrow$ В6) не определена и граф G_2(Возраст) меняет структуру на {В1 $\rightarrow$ В4; В4 $\rightarrow$ В5}.

Примеры веерного вычисления значений теста на основе графа G_2(Возраст):

Возраст/ В1? 5 $\rightarrow$ Возраст/ В4? [0; 17] $\rightarrow$ Возраст/ В5? молодой;

Пол/{М; Ж}? М & Возраст/ В4? [0; 17] $\rightarrow$ Возраст/ В6? нетрудоспособный.

Граф G_1(Возраст) имеет линейную структуру с одной терминальной вершиной, а граф G_2(Возраст) – ветвящуюся, с двумя терминальными вершинами.

Определим *операцию объединения* '$\cup$' разных графов одного и того же теста, если у них совпадают как минимум базовые домены. В результирующем графе общим становится базовый домен. Обязательно объединяются и другие вершины - домены, если между элементами этих доменов имеется взаимно – однозначное соответствие и доказано, что отсутствуют конфликты (разные пути вычислений приводят к разным значениям в одном домене). Пусть $G_k(\tau\,|[\{a/A\}_k])$ – разные графы теста τ (k = 1,…, K) с совпадающим базовым доменом. В результате объединения получим граф $G(\tau)$:

$$G(\tau\,|\,\cup_{k=1,\dots K}\,[\{a/A\}_k]) = G_1(\tau\,|\,[\{a/A\}_1]) \cup\dots\cup G_K(\tau\,|\,[\{a/A\}_K]) \qquad (1)$$

В некоторых случаях может применяться *операция прицепления* одного графа к другому. Граф $G_2(\tau)$ может *прицепляться* базовой вершиной к графу $G_1(\tau)$ в вершине T с обязательным объединением других совпадающих вершин тогда и только тогда, когда совпадает состав элементов домена T и базовой вершины графа $G_2(\tau)$, совпадают их смысловые зоны, и нет конфликтов значений в результирующем графе. Граф, получаемый в результате прищепления $G_2(\tau)$ к $G_1(\tau)$, обозначим следующим образом: $G(\tau) = G_1(\tau)[T]G_2(\tau)$.

Пусть G(Возраст) = G_1(Возраст) $\cup$ G_2(Возраст | {Пол}). Необходимо стремиться к тому, чтобы терминальные вершины графа были бинарными конструктами. В этом смысле граф G(Возраст) не полный.

Введем *операцию наращивания вершины-домена* следующим образом: для любой дискретной вершины $D = \{a_1, a_2, \ldots, a_n\}$ такой, что $n > 2$, порождаются n терминальных вершин вида $\{a_j;$ *не* $a_j\} \equiv \{a_j; \neg\, a_j\} \equiv \{a_j; D \setminus a_j\}$, где $j = 1,\ldots,n$. В результирующем графе остаются только те вершины, которые отсутствовали в графе до выполнения операции наращивания. В рамках настоящего исследования будем предполагать, что реализован однозначный воспроизводимый алгоритм проверки совпадения (тождественности) вершин с учетом синонимии и разных областей отрицания (в разных доменах). Граф, получаемый в результате применения операции наращивания ко всем допустимым вершинам, назовем *структурно-завершенным*. Структурно-завершенный граф, построенный на основе графа $G(\tau)$, обозначим через $G^+(\tau)$.

Предложение 1. Для произвольного графа $G(\tau)$ его структурно-завершенный граф $G^+(\tau)$ определяется в общем случае неоднозначно, но состав вершин во всех графах $G^+(\tau)$ одинаковый.

Множество структурно-завершенных графов для графа $G(\tau)$ обозначим через $\{G^+(\tau)\}$.

Операция наращивания графа, в том виде как она приведена выше, не обеспечивает получение всех *истинных* структурно-завершенных графов. Построение *истинного множества структурно-завершенных графов* предполагает первоначальное выявление всех *блуждающих вершин* (вершин, которые образуются в результате наращивания и которые могут иметь разные домены-предки) с последующим построением всех комбинаций расположения блуждающих вершин. Предложение 1. справедливо и для истинного множества структурно-завершенных графов.

К какому типу непрерывному или дискретному отнести вершину В1? Если ее отнести к дискретному типу (каждый год – отдельный элемент), то при построении структурно-завершенного графа операция наращивания создаст 101 новую терминальную вершину. Для большинства приложений это ожжет оказаться многовато. Исключить вершину В1 из списка вершин, к которым будет применена операция наращивания, можно лишь одним способом – отнести ее к типу «*конструктивная непрерывность*», что эквивалентно фразе «очень большое число элементов». К вершинам графа двух типов - «непрерывная» и «конструктивно непрерывная» - операция наращивания не применяется.

Результатом применения операции наращивания к вершине B3 являются три новые вершины:

B7 = {молодой; средних лет $\vee$ пожилой} $\equiv$ {молодой; *не* молодой};

B8 = {средних лет; молодой $\vee$ пожилой} $\equiv$ {средних лет; *не* средних лет};

B9 = {пожилой; молодой $\vee$ средних лет} $\equiv$ {пожилой; *не* пожилой}.

Результатом применения операции наращивания к вершине B5 являются три аналогичные вершины. Так как вершина B6 является дихотомической, то к ней операция наращивания не применяется. После наращивания всех вершин, включая B2 и B4, получим структурно – завершенный граф G^+(Возраст | {Пол}), в котором 23 вершины.

Построим структурно-завершенный граф для теста «Жалобы на боли в области живота», исходный граф которого базируется на двух доменах:

Жалобы на боли в области живота ^ЖБОЖ:

D2 {Болей в животе нет (норма) ^N; Жалобы на боли в животе ^b}

D1 {Болей в животе нет (норма) ^N; Постоянные ноющие боли в правом подреберье ^b; Схваткообразные боли в правом подреберье ^b; Опоясывающие боли в эпигастрии ^b; Боли в левом подреберье ^b}

Граф *G*(^ЖБОЖ) = {D1 $\rightarrow$ D2}.

Правила преобразования: D1.^b $\rightarrow$ D2.^b; D1.^N $\rightarrow$ D2.^N.

Формально вершина - домен D1 в результате наращивания порождает 5 терминальных вершин (по числу элементов), однако вершина D2 входит в число данных вершин, поэтому структурно-завершенный граф G^+(^ЖБОЖ) содержит 6 вершин – одну базовую и 5 терминальных (рис. 1).

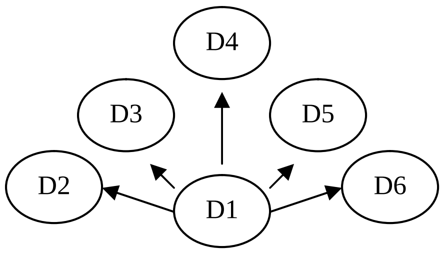

Рис. 1. Структурно-завершенный граф теста «Жалобы на боли»

Автоматически порождаемые в результате операции наращивания терминальные вершины, а также вершины, которые можно отнести к такому типу, образно можно назвать «*листьями*». Листьев в графе может быть достаточно много и хранить их в памяти весьма неэкономно. В памяти достаточно хранить *ствол графа* – граф без листьев, а листья должны порождаться автоматизмами вычислительной среды в моменты использования графа. Для произвольного графа $G(\tau)$ его ствол обозначим через $G^{\sim}(\tau)$.

Не все дихотомические вершины являются листьями, например, вершина B6 в графе G(Возраст) не является листком.

Предложение 2. Ствол произвольного графа $G(\tau)$ определяется единственным образом, состав вершин всех графов из $\{(G^{\sim}(\tau))^{+}\}$ одинаковый и, кроме того, выполняются следующие соотношения:

(i) $\forall\ G(\tau),\ \{G^{+}(\tau)\} \subseteq \{(G^{\sim}(\tau))^{+}\}$.

(ii) $\forall\ G^{+}(\tau)\ \in \{G^{+}(\tau)\},\ (G^{+}(\tau))^{\sim} = G^{\sim}(\tau)$.

Примеры. Ствол графа G(Возраст) совпадает с самим графом. Ствол графа G(^ЖБОЖ) состоит из одной вершины D1.

Любой из элементов домена B4 можно разбить на несколько частей, например, первый элемент [0; 17] можно представить как обобщение элементов {[0; 5]; [6; 10]; [11; 17]}. Новые элементы также можно разбить на несколько частей. При любом подобном разбиении получается новый – более точный домен. *Операцию деления* произвольного домена, состоящего из интервалов, можно формализовать разными схемами. Одни схемы являются чистыми разбиениями (например, каждый элемент домена делится на N равных частей), другие схемы предполагают просеивание, т.е. отбрасывание некоторых участков (фрактальный способ деления). Если базовый домен непрерывный, то операция деления домена наследника может выполняться бесконечное число раз, символизируя отсутствие пределов делимости материи и информации. *Схема деления* Σ_τ домена наследника для базового непрерывного домена является неотъемлемой частью графа $G(\tau)$. В общем случае, схема деления задается совокупностью параметров $\{s/S\}$, т.е. $\Sigma_\tau = \{\underline{s}/S\}_\tau$.

При решении локальной задачи из графа $G(\tau)$ может быть выделен произвольный подграф $G'(\tau)$ с одной базовой вершиной. Препятствием для выделения могут быть лишь онтологические соглашения на маршруте до базовой вершины подграфа. В случае выделения подграфа при наличии таких соглашений может нарушиться логика вычислений.

Как известно время выхода на пенсию будут постепенно повышать, следовательно, будет меняться и верхняя граница трудоспособного возраста (55 лет, 60 лет и т.д.). В общем случае, любая граница, разделяющая элементы домена, может динамически меняться в зависимости от контекста ситуации. Следовательно, в графе $G(\tau)$ там, где это целесообразно вместо чисел нужно использовать параметры $\{\underline{p}/P\}$, которые также являются тестами (со своими графами). Итоговая спецификация графа имеет вид:

$$G(\tau|[\{a/A\}]\ |\ [\{p/P\}]\ |\ [\Sigma_\tau]) = \{T \to T'\}, \tag{2}$$

где $\{a/A\}$ – внешние тесты для расчета значений; $\{p/P\}$ – структурные параметры; Σ_τ - схема деления домена - наследника для базового домена. Параметры могут быть взаимозависимы, что определяется онтологическими соглашениями. Структурные параметры должны быть заданы перед началом использования графа.

2. Замыкание, верхний и нижний пределы множества данных

Введем в рассмотрение функцию *Test*, которая произвольному множеству значений $\{\underline{\tau}/T\}$ ставит в соответствие множество используемых тестов $\{\tau\}$. Можем записать: $Test(\{\underline{\tau}/T\}) = \{\tau\}$.

Примеры:

Test({ФИО/C? Петров И.И.; Пол/C? М; Возраст/N? 34; АДс/N? 145; АДд/N? 87}) = {ФИО; Пол; Возраст; АДс; АДд};

Test({Возраст/ B1? 33; Возраст/ B4? [31; 40]}) = {Возраст}.

Введем операцию $\{\underline{\tau}/T\}\nabla\{G(\tau)\}$ означающую, что если $\tau \in Test(\{\underline{\tau}/T\})$, то $G(\tau) \in \{G(\tau)\}$.

Произвольное множество $\{\underline{\tau}/T\}$ назовем *изначально неконфликтным*, если у любого теста в рамках одного домена присутствует только одно значение.

Произвольную совокупность графов $\{G(\tau)\}$ назовем *онтологически согласованной*, если при любых реальных изначально неконфликтных исходных данных $\{\underline{\tau}/T\}\nabla\{G(\tau)\}$, описанных в терминах базовых доменов, в результате применения всех онтологических соглашений, не выходящих за рамки $\{G(\tau)\}$, не возникает конфликта значений тестов (не появляются разные значения одного теста в рамках одного домена).

Предложение 3. Если $\{G(\tau)\}$ онтологически согласовано, то любое множество структурно-завершенных графов $\{(G^{\sim}(\tau))^{+} \mid G(\tau) \in \{G(\tau)\}\}$ также онтологически согласовано.

Замыканием произвольного множества значений $\{\underline{\tau}/T\}$ над графами $\{G(\tau) \mid \tau \in Test(\{\underline{\tau}/T\})\}$ назовем множество $\{\underline{\tau}/T\}^{+}$, содержащее все вычисляемые значения тестов на основе графов и исходного множества $\{\underline{\tau}/T\}$.

Множество $\{\underline{\tau}/T\}$ назовем *неконфликтным* над графами $\{G(\tau) \mid \tau \in Test(\{\underline{\tau}/T\})\}$, если замыкание $\{\underline{\tau}/T\}^{+}$ над данными графами не содержит конфликтов значений тестов.

Неконфликтное множество данных не может содержать более одного значения теста в рамках любого домена. Однако это не означает, что нельзя строить замыкания для произвольных множеств $\{\underline{\tau}/T\}$.

Изначально неконфликтное множество $\{\underline{\tau}/T\}$ не всегда является неконфликтным. Изначально конфликтное множество остается конфликтным всегда. Пример конфликтного множества: $\{\underline{\tau}_1;\ \underline{\tau}_2\}/T$. Изменение состава множества графов $\{G(\tau)\}$ может изменить характеристику конфликтности $\{\underline{\tau}/T\}$ на противоположную.

Верхним пределом множества $\{\underline{\tau}/T\}$ над графами $\{G(\tau) \mid \tau \in Test(\{\underline{\tau}/T\})\}$ назовем множество $\{\underline{\tau}/T\}^{\wedge}$, являющееся подмножеством замыкания $\{\underline{\tau}/T\}^{+}$ и состоящее из результатов тестов максимального уровня общности (значения принадлежат доменам, образующим терминальные вершины).

Нижним пределом множества $\{\underline{\tau}/T\}$ над графами $\{G(\tau) \mid \tau \in Test(\{\underline{\tau}/T\})\}$

назовем множество $\{\underline{\tau}/T\}^?$, являющееся минимальным подмножеством $\{\underline{\tau}/T\}^+$, замыкание которого содержит $\{\underline{\tau}/T\}$, т.е. $\{\underline{\tau}/T\} \subset (\{\underline{\tau}/T\}^?)^+$.

Согласно определениям $\{\underline{\tau}/T\}$, $\{\underline{\tau}/T\}^?$, $\{\underline{\tau}/T\}^\wedge \subset \{\underline{\tau}/T\}^+$. Очевидно, выполняются соотношения: $\{\underline{\tau}/T\}^+ = (\{\underline{\tau}/T\}^?)^+$ и $\{\underline{\tau}/T\}^? = (\{\underline{\tau}/T\}^+)^?$ над графами $\{G(\tau) \mid \tau \in Test(\{\underline{\tau}/T\})\}$. В идеальном случае замыкание, верхний и нижний пределы нужно искать над графами $\{(G^{\sim}(\tau))^+ \mid \tau \in Test(\{\underline{\tau}/T\})\}$.

Подчеркнем, что введенные выше операции замыкания, получения верхнего и нижнего пределов целиком опираются на онтологические описания вида (2), т.е. не используют возможности базы знаний [4]. Операции вычисления замыкания, нижних и верхних пределов следует отнести к автоматизмам вычислительной среды.

Во многих случаях для принятия решения достаточно информации $\{\underline{\tau}/T\}^\wedge$, т.е. информации максимального уровня общности. Напротив, при хранении прецедентов в базе данных достаточно хранить нижние пределы множеств результатов тестов. При извлечении прецедента - ситуации из памяти с помощью автоматизмов среды возможно генерирование полного описания ситуации (замыкания множества всех имеющихся данных).

Справедливо следующее утверждение.

Предложение 4. Пусть задано $\{\underline{\tau}/T\}$ тогда замыкание и верхний предел над графами $\{G(\tau) \mid \tau \in Test(\{\underline{\tau}/T\})\}$ определяются единственным образом, а нижний предел, в общем случае, определяется не единственным образом.

Под *Банком тестов* будем понимать произвольное онтологически согласованное множество графов $\{G(\tau)\}$. Банк тестов является частью онтологии ПрО. Пусть фиксирован произвольный банк тестов $\{G(\tau)\}$.

Замыканием произвольного множества значений $\{\underline{\tau}/T\}$ над банком тестов $\{G(\tau)\}$ назовем множество $\{\underline{\tau}/T\}^+$, содержащее все вычисляемые значения тестов на основе графов банка тестов и исходного множества $\{\underline{\tau}/T\}$.

Множество $\{\underline{\tau}/T\}$ назовем *неконфликтным* над банком тестов $\{G(\tau)\}$, если замыкание $\{\underline{\tau}/T\}^+$ над данными графами не содержит конфликтов значений тестов.

Верхним пределом множества $\{\underline{\tau}/T\}$ над банком тестов $\{G(\tau)\}$ назовем множество $\{\underline{\tau}/T\}^\wedge$, являющееся подмножеством замыкания $\{\underline{\tau}/T\}^+$ и состоящее из результатов тестов максимального уровня общности (значения принадлежат доменам, образующим терминальные вершины графов из $\{G(\tau)\}$).

Нижним пределом множества $\{\underline{\tau}/T\}$ над банком тестов $\{G(\tau)\}$ назовем множество $\{\underline{\tau}/T\}^?$, являющееся минимальным подмножеством $\{\underline{\tau}/T\}^+$, замыкание которого содержит $\{\underline{\tau}/T\}$, т.е. $\{\underline{\tau}/T\} \subset (\{\underline{\tau}/T\}^?)^+$.

В приложениях используются как первые, так и вторые определения замыкания, верхнего и нижнего пределов.

3. Тест «Время». События. Описание ситуации

Пусть t/Λ - выделенный тест, означающий время (абсолютное). Примеры доменов Λ: $\Lambda 1$= ‘Дата: время’; $\Lambda 2$= ‘Дата: {утро, день, вечер, ночь}’; $\Lambda 3$= ‘Дата’; $\Lambda 4$= ‘Месяц, Год’; $\Lambda 5$= ‘Год’.

Возможны и другие домены. Конкретное значение времени согласно принятой нотации будем обозначать: $\underline{t}/\Lambda$. Примеры конкретных значений времени: $\underline{t}/\Lambda 1$ = ’10.01.09: 08.30’, $\underline{t}/\Lambda 2$ = ’10.01.09: утро’, $\underline{t}/\Lambda 3$ = ’10.01.09’, $\underline{t}/\Lambda 4$ = ’январь, 2009’, $\underline{t}/\Lambda 5$ = ’2009’.

Вполне очевиден способ задания пересчета из одного домена в другой (с большим номером), т.е. определена вычислительная цепочка (ориентированный граф доменов): $\Lambda 1 \to \Lambda 2 \to \Lambda 3 \to \Lambda 4 \to \Lambda 5$.

Уточнения требует лишь связка $\Lambda 1 \to \Lambda 2$, которую зададим следующими правилами: ‘00.00 – 6.00’ $\to$ ‘ночь’; ‘6.00 – 10.00’ $\to$ ‘утро’; ‘10.00 – 17.00’ $\to$ ‘день’; ‘17.00 – 24.00’ $\to$ ‘вечер’.

Определим два *элементарных события*:

$$e_1 = <\underline{\tau}/T, \underline{t}/\Lambda> \quad \text{и} \quad e_2 = <\neg\underline{\tau}/T, \underline{t}/\Lambda>. \tag{3}$$

Событие e_1 означает, что в момент времени $\underline{t}/\Lambda$ тест τ/T имел значение $\underline{\tau}/T$. Событие e_2 означает, что в момент времени $\underline{t}/\Lambda$ тест τ/T *не* имел значение $\underline{\tau}/T$. Положим: $e_1 = \neg e_2$, $e_2 = \neg e_1$. Примеры элементарных событий:
<Кашель/{Имеется; Отсутствует}? Имеется, $\underline{t}/\Lambda 1$? 23.02.08: 12.45>,
<¬’Le крови’/{Снижено; Норма; Повышено}? Норма, $\underline{t}/\Lambda 3$? 23.02.08>;
<Ток солнечных батарей/{Низкий; Рабочий}? Низкий, $\underline{t}/\Lambda 1$? 13.05.09: 09.15>;
<Риск инвестиций /{Высокий; Средний; Низкий}? Высокий, $\underline{t}/\Lambda 5$? 2008>.

Любое событие может быть поименованным. Если предметная область «Медицина», то все элементарные события фиксируются в медицинской карте (бумажной или электронной).

Шаблоном события назовем кортеж $<\tau/T, t/\Lambda>$. Шаблоны событий задаются, как правило, перед началом мониторинга ситуации действительности.

Событие $<\underline{\tau}/T, \underline{t}/\Lambda>$ можно представить как результат теста: «Значение τ/T в момент $\underline{t}/\Lambda = \underline{\tau}$». Подобная трактовка позволяет упростить многие определения.

Если для теста τ определен ориентированный граф доменов $G(\tau)$, а для времени граф $G(t)$, то событие $<\underline{\tau}/T, \underline{t}/\Lambda>$, описанное с помощью базовых доменов, автоматически порождает множество событий $\{<\underline{\tau}/T', \underline{t}/\Lambda'>\}$ общей численностью $N_\tau \times N_t$, где N_τ – общее количество доменов в графе $G(\tau)$, а N_t – общее количество доменов в графе $G(t)$.

На основе элементарных событий можно определить *составные события*, следующим образом: $e = <\{\underline{\tau}/T\}, \underline{t}/\Lambda>$ или $e' = <\{[\neg]\ \underline{\tau}/T\}, \underline{t}/\Lambda>$.

Для составных событий эквивалентны записи: $e = \{e_1, e_2,..., e_n\} = \{<\underline{\tau}/T, \underline{t}/\Lambda>\} = <\{\underline{\tau}/T\}, \underline{t}/\Lambda>$. Однако обратная операция – разделение составных событий на независимые элементарные события – не всегда возможна, так как составные события часто объединяют *батарею тестов*, т.е. те тесты, которые неразрывно связаны между собой.

Примеры составных событий: жалобы (с указанием времени фиксации жалоб), осмотр (с указанием времени осмотра), анализ крови, параметры ЭКГ, АД, пульс и т.д. Так для АД и пульса можем записать:

АД1= <АДс/N? 145, АДд/N? 93, $\underline{t}/\Lambda 1$? 13.05.09: 12.45>;

Пульс2 = <Частота/N? 70, Ритмичность/D1? ритмичный, $\underline{t}/\Lambda 2$? 23.02.08: утро>;

Исследование пульса на лучевых артериях ($\underline{t}/\Lambda 2$? 23.02.08: утро): {

ЧАСТОТА/N: 86;

РИТМИЧНОСТЬ/D1={ритмичный; аритмичный}: ритмичный;

НАПРЯЖЕНИЕ/{среднего напряжения; твердый; мягкий}: твердый;

НАПОЛНЕНИЕ/{среднего наполнения; полный; пустой}: полный;

ФОРМА/{быстрый; скачущий; медленный}: быстрый;

ДЕФИЦИТ/{имеется; отсутствует}: отсутствует}

Приведенные примеры объединяют батареи тестов, которые всегда выполняются пакетом. Заметим, что «Исследование пульса на лучевых артериях» следует трактовать как название составного события (батареи тестов). Каждый тест батареи тестов, как и любой тест, задается своим графом. В приведенных выше примерах графы тестов состоят из одного домена.

Если для разных времен $\underline{t}/\Lambda$ повторяется состав тестов $\{\underline{\tau}/T\}$, то составной тест $<\{\underline{\tau}/T\},\ \underline{t}/\Lambda>$ можно рассматривать как *кадр*. Примером является последовательность изображений (видеоряд, видеопоток), где каждое изображение - это составной тест.

Для многих тестов, кроме их результатов, важной дополнительной (служебной) информацией являются, например, такие тесты: место проведения теста; кем проводился тест; стоимость теста; достоверность результата теста (оператор - тест *J*) и т.д. В общем случае, у каждого теста или события может быть свой набор уточняющих тестов. Данный набор прописывается в рамках онтологии предметной области.

Если через $\{p/P\}$ обозначить дополнительные параметры – уточняющие тесты, то расширенная трактовка элементарных событий примет вид составных событий: $e_1 = <\underline{\tau}/T,\ \{\underline{p}/P\},\ \underline{t}/\Lambda>$ и $e_2 = <\neg\underline{\tau}/T,\ \{\underline{p}/P\},\ \underline{t}/\Lambda>$. Пример расширенного описания составного события:

АД1 = <АДс/D2? [140; 150]; АДд/D2? [85; 90]; МестоПровТеста/C1? Дом; КемПровТест/C2? Врач СМП; Метрология/M3? Выполнена; $\underline{t}/\Lambda 1$? 13.05.09: 12.45>

Очень часто необходимость проведения одних тестов зависит от результатов других тестов. Приведем пример. Если «ГОЛОВНАЯ БОЛЬ/{отсутствует; имеется}= отсутствует», то больше никаких уточняющих тестов проводить не нужно. Однако, если «ГОЛОВНАЯ БОЛЬ/{отсутствует; имеется}= имеется», то необходимо уточнить локализацию боли, ее характер, периодичность и т.д., которые также являются тестами. Возникает проблема процедурного представления взаимосвязанной совокупности доменов тестов (графов доменов тестов).

В самом простом случае, если граф доменов теста «ТЕСТ» состоит из одного базового домена, то при наличии уточняющих тестов на процедурном уровне его можно представить следующим образом:

$$\textit{ТЕСТ}: \textit{элем}1\ [\{\textit{Уточн. тесты}\}];\ldots;\ \textit{элемK}\ [\{\textit{Уточн. тесты}\}], \tag{4}$$

где ‘*элем_k*’ – к-й элемент базового домена; [{*Уточн. тесты*}] – необязательное множество уточняющих тестов каждого элемента домена. Каждый уточняющий тест, в простейшем случае, имеет ту же структуру.

Для корректного отражения логики уточняющих тестов во множестве результирующих данных можно использовать следующую нотацию:

$$\{\underline{\tau}/T\} = \{\underline{\tau}_1/T_1\ [\{\underline{\tau}/T\}_1]\ ;\ldots;\ \underline{\tau}_n/T_n\ [\{\underline{\tau}/T\}_n]\}, \tag{5}$$

где $\{\underline{\tau}/T\}_j$ – множество результатов уточняющих тестов к j-му тесту $\underline{\tau}_j/T_j$. Уровней уточнения может быть сколь угодно много. Пример:
$\{\underline{\tau}/T\}$ = {Головная боль? имеется {Локализация? лобная область; Характер? тупая;…}; ЧСС? 77; Онемение? имеется {Тип? тактильное; Интенсивность? умеренная;…}; Снижение зрения? имеется {Глаз? правый {Характер? вдаль;…}; Глаз? левый {Характер? вблизи;…}}}

Протяженным событием назовем событие $e = <\{\underline{\tau}/T\}, \underline{\delta}/\Lambda>$, где $\underline{\delta}$ - произвольный временной интервал (возможно не односвязный). Для интервалов определена операция объединения: $e = \&_i <\{\underline{\tau}/T\}, \underline{\delta}_i/\Lambda> = <\{\underline{\tau}/T\}, \underline{\delta}/\Lambda>$, где $\underline{\delta} = \&_i\ \underline{\delta}_i$. Пример протяженного события:

<Креатинин / {мг / 100 мл}? 1.3, [12.02.09; 14.02.09]/Λ3>,

где δ = [12.02.09; 14.02.09] (в терминах домена Λ3).

Если для любого момента времени t в рассматриваемом периоде наблюдения значение некоторого теста $\underline{\tau}/T$ не меняется, то такой результат назовем *фактом*. Так, например, перенесенные в прошлом операции или травмы являются фактами, заболевания родителей (наследственность) являются фактами и т.д.

Жизненным циклом произвольного теста не являющегося фактом назовем множество $\{<\underline{\tau}/T, \underline{t}/\Lambda> \mid \tau/T$ фиксировано, Λ фиксировано$\}$.

Автоматизмами вычислительной среды из любого события $<\underline{\tau}/T, \underline{t}/\Lambda>$ формируется связанное событие-факт: «*Имело место* $\underline{\tau}/T$». Событие-факт позволяет устранить привязку значения теста к конкретному времени, что упрощает во многих случаях установление формальной эквивалентности между событиями.

Пусть J - оператор оценки истинности/возможности той или иной информации. Если информация x имеет истинность β, то будем использовать нотацию: $J_\beta x$ или $J_\beta\, \underline{\tau}/T$. С возрастанием уровня общности результата теста, как правило, меняется и значение истинности результата.

Оценка истинности J также как любой тест может выражаться с помощью доменов разного уровня общности – от числовой шкалы до лингвистической шкалы, например: “абсолютно достоверно” (полная определенность); “скорее всего” (крайне высокая степень определенности); “весьма вероятно” (высокая степень определенности) и т.д.

Произвольную ситуацию действительности обозначим через $\alpha = \alpha(\{<J_\tau\ \underline{\tau}/T, J_t\ \underline{t}/\Lambda>\})$, а множество ситуаций действительности обозначим через $\Omega = \{\alpha_1, \alpha_2, \ldots, \alpha_\mathrm{n}\}$. Множество Ω является основой для системной реконструкции «образа мира» [4].

Полную модель описания произвольной ситуации действительности на основе онтологии можно представить следующим образом:

$$\begin{aligned} &\alpha = \alpha(\{<J_\tau\ \underline{\tau}/T, \{\ J_p\ \underline{p}/P\}, J_t\ \underline{t}/\Lambda>\}); \\ &\forall\ \tau, G(\tau\,|[\{a/A\}]\,|\,[\{p/P\}]\,|\,[\Sigma_\tau]) = \{T \to T'\}; \\ &\forall\ \tau, G(\tau) \to \{G^+(\tau)\}, G^{\tilde{}}(\tau), \{(G^{\tilde{}}(\tau))^+\}; \\ &\forall\ \{\underline{\tau}/T\} \to \{\underline{\tau}/T\}^?, \{\underline{\tau}/T\}^\wedge, \{\underline{\tau}/T\}^+. \end{aligned} \tag{6}$$

Последняя строка описывает, по-сути, автоматизмы вычислительной среды, которые функционируют постоянно в процессе решения любой задачи анализа и управления. В дальнейшем данная модель будет дополнена моделью знаний [4].

4. Выводы

Графы доменов содержат экспертные знания. Они являются удобным и математически корректным способом кодирования разноуровневых элементов эмпирических систем. Каждый граф является, по сути, моделью индуктивного обобщения результатов теста. Для числовых тестов ключевым моментом является переход от непрерывного интервала к дискретному разбиению (фазовый переход от бесконечности к конечности). Подобный переход может быть выполнен разными способами, что, безусловно, отражается на результатах моделирования. Важно отметить, что графы доменов стирают границу между непрерывным и дискретным – любой непрерывный тест всегда имеет множественное дискретное представление (интервальное, символическое).

Литература

1. Саати Т. Принятие решений. Метод анализа иерархий. Пер. с англ. – М.: «Радио и связь», 1993. – 320 с.
2. Прокопчук Ю.А. Интеллектуальные медицинские системы: формально-логический уровень. – Днепропетровск: ИТМ НАНУ и НКАУ, 2007.- 259 с.
3. Прокопчук Ю.А. Метод предельных обобщений для решения слабо формализованных задач // Управляющие системы и машины. - 2009. - №1. – С.31 – 39.
4. Прокопчук Ю.А. Информационная структура теории естественной предметной области // Вестник ХНТУ, 2010. - №2(38). – С. 11 – 19.