О.М. Аншаков

# ДСМ-метод и модификационные исчисления

**Аннотация.** Предлагается общее неформальное введение в новый тип исчислений, который позволяет применять как достоверные, так и правдоподобные правила вывода, допускает немонотонность и временную противоречивость. В связи с использованием некоторого известного понятийного аппарата, системы терминов и утверждений потребовалось объяснить, в каком конкретном смысле они применяются в статье.

**Ключевые слова:** многозначные логики, структуры, правила рассуждений, модификационные исчисления

## 1. Основные понятия

### 1.1. Факты и знания

Первая пара понятий, которая подлежит обсуждению, — факты и знания.

Вспомним, что понятие факта, как и понятие знания, можно определить по-разному. Факт — это:

(i) сведения (данные), которые получены в результате наблюдения или эксперимента,

(ii) конкретное элементарное утверждение без параметров (имеющее тот же формат, что экспериментальные данные). Факт можно представить в виде атомарной формулы без переменных.

Факт в смысле (i) является достоверной информацией (по крайней мере, не менее достоверной, чем сам эксперимент). Факт в смысле (ii) может быть получен не только в результате эксперимента, но и, например, в результате логического вывода, использующего эвристики (как это делается в экспертных системах). В этом случае, он будет, разумеется, только гипотезой.

Определяя знания, их обычно противопоставляют фактам. Можно выделить следующие признаки знаний:

(i) Знания невозможно получить непосредственно из результатов экперимента. Формирование знаний происходит посредством анализа экспериментальных данных. При анализе проводятся рассуждения, причем эти рассуждения могут быть не только дедуктивнвми, но и правдоподобными. Например, можно применять индуктивное обобщение. Процесс формирования знаний может потребовать дополнительных экспериментов для проверки гипотез, полученных с помошью рассуждений.

(ii) Знания выражают общие закономерности. Их можно представить в виде формул с кванторами общности или в виде правил (продукций), содержащих переменные.

И для фактов, и для знаний, в определении (i) перечисляются характеристики, связанные с происхождением, а в определении (ii) — особенности формата представления. Для компьютерных приложений, в том числе и в сфере искусственного интеллекта, более удобным являются, разумеется, определения через формат представления. Однако следует отметить, что формат представления может быть не единственным.

Например, что представляет собой утверждение о причинно-следственных отношениях? Ведь сведения о наличии (или отсутствии) причинно-следственных связей не могут содержаться *непосредственно* в экспериментальных данных. Сделать вывод о причинно-следственных связях можно только в результате анализа таких данных. Иногда такой анализ будет очень простым, и даже неосознанным, но он обязательно должен присутствовать/быть. Значит с точки зрения происхождения, сведения о причинно-следственных связях, являются знаниями.

С точки зрения формы представления, утверждение о причинно-следственной связи двух конкретных событий, например:

*электрический разряд в воздухе* (молния) является причиной *громкого звука* (грома)

является конкретным элементарным утверждением без параметров, т.е., *фактом*. С другой стороны, его можно интерпретировать и как универсальное утверждение (утверждение с квантором общности), описывающее закономерность:

*в каждом случае*, после электрического разряда в воздухе (молнии) слышен громкий звук (гром).

Тогда необходимо считать утверждение о причинно-следственной связи *элементом знания*.

Так как же будут интерпретироваться термины *факт* и *знание* в этой статье.

*Во-первых*, не будем исключать ни одного из возможных толкований этих терминов. Какое конкретно толкование будет выбрано, должно быть ясно из контекста или специально оговариваться в каждом случае употребления этих терминов.

*Во-вторых*, будем предпочитать прагматическую точку зрения. А именно, если для функционирования некоторой системы или применения некоторой методологии, *удобно* считать некоторое утверждение фактом, то оно и будет рассматриваться как факт, ели *удобно* считать его элементом знания, то мы и будем считать его элементом знания. Если одно и то же утверждение на одном этапе работы системы (или применения методологии) удобно рассматривать как факт, а на другом, как знание, то мы и будем считать это утверждение в одно случае фактом, а в другом — знанием.

Подводя итоги, все-таки необходимо сказать, как следует понимать термины *факт* и *знание* в данной статье в подавляющем большинстве случаев:

- ***факт*** — это предложение, *представленное* (в контексте функционирования некоторой системы или применения некоторой методологии) *в виде конкретного элементарного утверждения без параметров*,
- ***знание*** — это предложение, *смысл которого состоит в отражении некоторой общей закономерности*, независимо от формы представления этого предложения (в контексте функционирования рассматриваемой системы или применения рассматриваемой методологии).

Понятно, что согласно приведенным выше определениям, можно *одно и то же предложение рассматривать и как факт и как знание*. Это действительно можно делать.

В статье вводится некоторая общая схема, позволяющая анализировать *факты*, получать в результате этого анализа *закономерности*, и формировать *гипотезы*. Но с формальной точки зрения, и факты, и закономерности и гипотезы, которые подлежат рассмотрению в рамках предлагаемого подхода, являются *фактами*, т.е. конкретными элементарными утверждениями, которые можно выводить атомарными формулами без параметров. С другой стороны, нельзя рассматривать обнаруженные (в процессе применения нашей методологии) закономерности как что-то, отличное от *знаний*. Задача, рассматриваемая в данной работе, является задачей открытия знаний (knowledge discovery).

## 1.2. Истинностные значения: неформальное обсуждение

Язык, на котором излагается материал в большей части этой статьи — это язык математической логики. В статье широко используются понятия, относящиеся к этой области математики. Если говорить более конкретно, то используется аппарат *многозначных* логик. На неформальном уровне это означает, что могут существовать истинностные значения, отличные от классических «*истинно*» и «*ложно*».

Интуитивная интерпретация «*дополнительных*» истинностных значений может быть различной. Например, достаточно широко распространена интерпретация истинностных значений в трехзначной логике как «*истинно*», «*ложно*» и «неопределенно». Если множество истинностных значений многозначной логики упорядочено так, что «*ложно*» является наименьшим, а «*истинно*» — наибольшим элементом этого множества, то об истинностных значениях говорят как о «степенях истинности».

В данной статье все истинностные значения многозначной логики делятся на два класса: класс внешних истинностных значений и класс внутренних истинностных значений. Внешних истинностных значений *два*: это обычные классические «истинно» и «*ложно*». Внутренних истинностных значений не менее двух. Одно из них называется «*неопределенно*» и означает полное отсутствие информации по интересующему нас вопросу. Другие могут выражать *степень и характер* определенности, информированности, уверенности.

Степень и характер определенности могут быть связаны с количеством и значимостью/весомостью аргументом, которые можно привести за или против того суждения, которое характеризуется/оценивается внутренним истинностным значением. Приведем пример, в котором четыре внутренних истинностных значения, среди которых встречается и «*неопределенно*», объясняются через наличие или отсутствие подтверждающих или опровергающих примеров (аргументов за или аргументов против). Пусть φ — утверждение, которое может принимать внутренние истинностные значения. Тогда:

- утверждение φ получает истинностное значение «эмпирически истинно», если:
  - существуют аргументы за утверждение φ,
  - не существует аргументов против утверждения φ;
- утверждение φ получает истинностное значение «эмпирически ложно», если:
  - существуют аргументы против утверждения φ,
  - не существует аргументов за утверждение φ.
- утверждение φ получает истинностное значение «эмпирически противоречиво», если:
  - существуют аргументы за утверждение φ,
  - существует аргументы против утверждения φ;
- утверждение φ получает истинностное значение «неопределенно», если:
  - не существует аргументов за утверждение φ,
  - не существует аргументов против утверждения φ.

Подчеркнем, что значение «эмпирически противоречиво» выражает характер определенности, оно является примером определенного внутреннего истинностного значения. По нашему соглашению, неопределенность означает полное отсутствие информации, а в случае эмпирического противоречия информация наличествует, хотя и является противоречивой.

Приведем еще один пример: пусть $p$ — количество аргументов «за», а $c$ — количество аргументов «против» (pro & contra). Тогда дробь

$$\frac{p}{p+c}$$

выражает «близость к эмпирической истине». Эта дробь равна 0 когда нет аргументов «за» и есть аргументы «против». Эта ситуация соответствует истинностному значению «эмпирически ложно». Эта дробь равна 1, если есть аргументы «за» и нет аргументов «против». Это соответствует истинностному значению «эмпирически истинно». Значению «неопределенно», соответствует неопределенность вида $\frac{0}{0}$.

Можно предложить и другое выражение, содержащее количества аргументов «за» и аргументов «против». Например, дробь

$$\frac{p-c}{p+c}$$

также можно использовать для количественной характеристики характера определенности. Эта дробь равна:

- +1, если есть аргументы «за» и нет аргументов против (эмпирически истинно),
- 0, если есть количество аргументов «за» равно количеству аргументов «против» (эмпирически противоречиво),
- –1, если есть аргументы «против», но нет аргументов «за» (эмпирически ложно).

Значению «неопределенно» также как и в предыдущем примере соответствует неопределенность вида $\frac{0}{0}$. В остальных случаях дробь $\frac{p-c}{p+c}$ характер определенности с точки зрения близости к значениям «эмпирически истинно» и «эмпирически ложно». Положительные числа соответствуют тем значениям, которые можно описать в терминах «скорее истинно, чем ложно», а отрицательные — «скорее ложно, чем истинно».

Каждому аргументу (как за, так и против) можно приписать свой весовой коэффициент. Тогда $p$ и $c$ следует интерпретировать как сумму весов аргументов «за» и аргументов «против», соответственно. При такой интерпретации $p$ и $c$ могут использовать прежние выражения для вычисления характера определенности.

Во всех приведенных выше примерах с помощью внутреннего истинностного значения выражается характер определенности. Чтобы одновременно с характером выразить и степень определенности можно истинностное значение представить с помощью упорядоченной пары. Например, в паре

$$\left\langle \frac{p-c}{p+c},\ p+c \right\rangle$$

первый элемент выражает характер (качество) определенности, а второй — степень (количество) определенности. Очевидно, что определенность тем выше, чем больше информации, т.е. чем больше как подтверждающих, так и опровергающих примеров.

В данной статье также рассматриваются внутренние истинностные значения, представленные в виде упорядоченных пар. Первый элемент пары — это характер определенности, который может быть вычислен одним из приведенных выше способов. Второй элемент пары — это номер стадии итеративного процесса, внутри которой было закончено вычисление характера определенности. Следует отметить, что тот итеративный процесс, который предлагается в статье, устроен таким образом, что на каждой следующей стадии получаются менее надежные (менее определенные) результаты, чем на предыдущей. В этом случае второй элемент пары следует понимать не как степень определенности, а как *степень неопределенности*.

Среди логик с двумя классами истинностных значений необходимо особо отметить логику, имеющую *всего два* внутренних истинностных значения: «*неопределенно*» (неизвестно) и «*определенно*» (известно). В этом минималистском случае принимаются во внимание только аргументы «за» и совершенно не учитываются аргументы «против». Тогда:

- утверждение получает истинностное значение «*неопределенно*», если нет аргументов «за» это утверждение;
- утверждение получает истинностное значение «*определенно*», если есть аргументы за это утверждение.

Завершим этот подраздел описанием интуитивной мотивации для деления истинностных значений на два класса: класс внешних и класс внутренних истинностных значений. Будем предполагать, что существуют два уровня знания:

- внешний, теоретический или конвенциональный,
- внутренний, эмпирический или индивидуальный.

На внешнем уровне находятся общие утверждения из устоявшейся общепринятой теории. Будем предполагать, что знания на внешнем уровне подчиняются законам классической логики.

На внутреннем уровне находятся факты (результаты экспериментов) и эмпирические закономерности, которые можно получить с помощью непосредственного анализа результатов экспериментов (например, с помощью каких либо статистических методов) без построения развитой теории. На внутреннем уровне можно использовать многозначную логику и формальные аналоги правил правдоподобных рассуждений, о которых будет сказано ниже.

## 1.3. Сорта и структуры

Логики, которые рассматриваются в данной статье не только многозначные, но и многосортные. *Сорт* является абстрактным аналогом атрибута (поля) таблицы реляционной базы данных. Вообще говоря, сорт можно понимать как *генерическое имя* класса объектов. Можно рассматривать сорт как аналог имени типа данных, но для нас будет более полезной аналогия с атрибутами.

Совокупность сортов делится на два класса:

- класс *предметных сортов* (individual sorts) и
- класс *логических сортов* (сортов истинностных значений) (truth-value sorts).

Предметные сорта — это аналоги нелогических атрибутов. Значениями предметных сортов могут быть числа, строки, события и т.п. Предметных сортов должно быть не менее одного.

Один предметный сорт имеет как раз односортная логика. Этот случай не исключается, но он не представляет для нас большого интереса.

Будем рассматривать логики с двумя сортами истинностных значений:

- сорт внешних истинностных значений (обозначается «Ext») и
- сорт внутренних истинностных значений (обозначается «Int»).

Каждому сорту (как предметному, так и логическому), также как и каждому атрибуту в реляционных базах данных, соответствует свой домен — множество всех возможных значений данного сорта. Например, домен сорта Ext содержит точно два элемента: *истинно* и *ложно*, т.е., Ext соответствует типу данных **Boolean**. Домен сорта Int содержит по крайней мере два элемента и среди них обязательно содержит истинностное значение *неопределенно*.

Домены могут быть как конечны, так и бесконечны. Но наиболее интересные результаты данной статьи касаются только конечных доменов. В дальнейшем в данной статье, если не оговорено противного, будем предполагать, что домены всех сортов конечны. Это еще более усиливает аналогию с реляционными базами данных.

В этой статье предполагается, что домены разных сортов различны и более того — не имеют общих элементов. Но это требование — техническое. Оно позволяет упростить некоторые определения и доказательства. Принципиального значения это ограничение не имеет и может быть отменено с сохранением всех результатов.

Главным семантическим понятием, без которого объяснить многие вещи было бы очень трудно, является понятие *структуры*. Чтобы определить структуру, необходимо (и достаточно) выполнить следующие действия:

- задать непустое множество предметных сортов (имен классов предметов),
- поставить в соответствие каждому сорту непустой домен,
- определить *операции* и *предикаты*.

Рассмотрим более подробно операции и предикаты. Как операции, так и предикаты являются функциями. Каждая операция (каждый предикат) имеет *арность* и *сорт*. Сорт операции (предиката) есть сорт ее значений. Отличие операции от предиката заключается в следующем:

- сорт операции является предметным сортом,
- сорт предиката является логическим сортом.

Предикаты сорта Ext называются *внешними*, а предикаты сорта Int — *внутренними*.

*Арность* операции (предиката) является строкой (string) сортов аргументов, взятых в определенном порядке. Операцию (предикат) можно наглядно представить в виде таблицы, заголовочная строка (heading row) которой содержит арность с приписанным к ней справа сортом. Каждая строка таблицы состоит из строки аргументов (argument row) и ячейки значения (value cell).

Ячейки таблицы удовлетворяют следующему условию/(подчиняются следующему ограничению):

- в каждом столбце таблицы,
  - каждая ячейка содержит элемент *того и только того* сорта, который записан в заголовочной ячейке этого столбца.

Левая часть таблицы (без столбца значений) содержит область определения (domain) операции (предиката). Эта область определения содержит все возможные строки (rows), которые удовлетворяют приведенному выше условию/(подчиняются приведенному выше ограничению).

Каждая ячейка столбца значений содержит значение функции от кортежа аргументов, записанных в той же строке, в которой находится данная ячейка значений. Общая схема таблицы для представления операций (предикатов) изображена в Табл. 1.

Табл. 1. Общая схема таблицы для представления операций и предикатов

| Arity<br>(argument fields) | Sort<br>(value field) |
|---|---|
| Domain | Value column |

Табличное представление *внутренних предикатов* очень полезно для объяснения/иллюстрации тех процессов, которые происходят при применения подхода, развиваемого в данной статье.

Приведем теперь пример структуры. Рассмотрим два сорта:

- сорт элементов (обозначается E),

- сорт множеств (обозначается S).

Пусть

- домен сорта E) содержит два элемента: **a** и **b**,
- домен сорта S содержит четыре множества: ∅, {**a**}, {**b**} и {**a**, **b**}.

Предположим, что наша структура содержит:

- две операции:
  - **insert** (включение элемента в множество),
  - **complement** (дополнение множества),
- один внешний предикат:
  - **belong** (элемент принадлежит множеству).

Будем обозначать внешние истинностные значения:

- «истинно» через **t**,
- «ложно» через **f**.

В Табл. 2 изображены упомянутые выше операции и предикат.

Табл. 2. Операции и предикат для примера структуры

**insert**

| Arity | | Sort |
|---|---|---|
| E | S | S |
| **a** | ∅ | {**a**} |
| **a** | {**a**} | {**a**} |
| **a** | {**b**} | {**a, b**} |
| **a** | {**a, b**} | {**a, b**} |
| **b** | ∅ | {**b**} |
| **b** | {**a**} | {**a, b**} |
| **b** | {**b**} | {**b**} |
| **b** | {**a, b**} | {**a, b**} |

**complement**

| Arity | Sort |
|---|---|
| S | S |
| ∅ | {**a, b**} |
| {**a**} | {**b**} |
| {**b**} | {**a**} |
| {**a, b**} | ∅ |

**belong**

| Arity | | Sort |
|---|---|---|
| E | S | Ext |
| **a** | ∅ | **f** |
| **a** | {**a**} | **t** |
| **a** | {**b**} | **f** |
| **a** | {**a, b**} | **t** |
| **b** | ∅ | **f** |
| **b** | {**a**} | **f** |
| **b** | {**b**} | **t** |
| **b** | {**a, b**} | **t** |

Рассмотрим теперь пример структуры, содержащей *внутренние предикаты*, т.е. предикаты сорта Int. Такие предикаты принимают внутренние истинностные значения. Пусть новая структура имеет дело с тремя предметными сортами:

- номер эксперимента (обозначается N),
- возможная причина (обозначается C),
- возможное следствие (обозначается E).

Пусть:

- домен сорта N содержит два номера: 1 и 2,
- домен сорта C содержит два события: **A** и **B**,
- домен сорта E содержит два события: **P** и **Q**.

Предположим, что наша структура содержит два внутренних предиката:

- **Exp** арности NCE — в эксперименте *n* наблюдается событие *c* и вслед за ним (через небольшой интервал времени) событие *e*,
- **PC** арности CE — событие *c* является возможной причиной события *e*.

Будем обозначать внутренние истинностные значения:

- «эмпирически истинно» через +1,
- «эмпирически ложно» через –1,
- «эмпирически противоречиво» через 0,
- «неопределенно» через τ.

Будем выбирать разные стратегии приписывания внутренних истинностных значений предикатам **Exp** и **PC**. Можно даже сказать, что для этих предикатов мы выбираем разные интуитивные интерпретации внутренних истинностных значений. Положим:

- $\mathbf{Exp}(n,c,e)=+1$, если в эксперименте *n*, происходит событие *c*, а вслед за ним происходит событие *e*,
- $\mathbf{Exp}(n,c,e)=-1$, если в эксперименте *n*, происходит событие *c*, а вслед за ним *не* происходит событие *e*,
- $\mathbf{Exp}(n,c,e)=\tau$, если в эксперименте *n*, нельзя однозначно опредилить происходит ли событие *e* вслед за событием *c* (например, если событие *c* вообще не происходит, или *e* происходит через такой большой промежуток времени после *c*, что связь между событиями *c* и *e* не очевидна).

При вычислении истинностных значений предиката **PC** используется таблица для предиката **Exp** как источник аргументов «за» и «против».

В Табл. 3 изображены упомянутые выше внутренние предикаты.

Табл. 3. Пример внутренних предикатов

**Exp**

| Arity | | | Sort |
|---|---|---|---|
| N | C | E | Int |
| 1 | **A** | **P** | +1 |
| 1 | **A** | **Q** | +1 |
| 1 | **B** | **P** | τ |
| 1 | **B** | **Q** | –1 |
| 2 | **A** | **P** | +1 |
| 2 | **A** | **Q** | –1 |
| 2 | **B** | **P** | τ |
| 2 | **B** | **Q** | –1 |

**PC**

| Arity | | Sort |
|---|---|---|
| C | E | Int |
| **A** | **P** | +1 |
| **A** | **Q** | 0 |
| **B** | **P** | τ |
| **B** | **Q** | –1 |

Поясним, каким образом вычисляются истинностные значения предиката **PC**:

- Как в эксперименте 1, так и в эксперименте 2, вслед за событием **A** наступает событие **P**. Значит есть аргументы за то, что **A** является причиной **P** и нет аргументов против этого утверждения. Тогда $\mathbf{PC}(\mathbf{A},\mathbf{P}) = +1$.
- В эксперименте 1 вслед за событием **A** наступает событие **Q**, а в эксперименте 2, вслед за событием **A** не наступает событие **Q**. Значит имеются как аргументы «за», так и аргументы «против» того, что **A** является причиной **Q**. Тогда $\mathbf{PC}(\mathbf{A},\mathbf{Q}) = 0$.
- Как в эксперименте 1, так и в эксперименте 2, нельзя определить, наступает ли за событием **B** событие **P**. То есть у нас отсутствуют как аргументы «за», так и аргументы «против» того, что **B** является причиной **P**. Тогда $\mathbf{PC}(\mathbf{B},\mathbf{P}) = \tau$.
- Как в эксперименте 1, так и в эксперименте 2, вслед за событием **B** не наступает событие **Q**. Значит есть аргументы против того, что **B** является причиной **Q** и нет аргументов за это утверждение. Тогда $\mathbf{PC}(\mathbf{B},\mathbf{Q}) = -1$.

Заметим, что точно такие же результаты будут получены при вычислении значения предиката **PC** как значения выражения

$$\frac{p-c}{p+c},$$

где $p$ — количество аргументов «*за*», $c$ — количество аргументов «*против*» (если $\tau$ будет обозначать неопределенность вида 0/0).

## 1.4. Рассуждения

В данной статье предлагается подход к формализации познавательной деятельности, центральным понятием которого является понятие рассуждения.

Можно очень долго обсуждать различные оттенки смысла данного понятия, приводить цитаты и анализировать мнения разных авторов, рассматривать эволюцию смысла этого понятия в историческом, культурном и идеологическом контексте. Можно обсуждать вопрос о том, следует ли противопоставлять рассуждение дедукции, или наоборот необходимо отождествлять рассуждение и дедукцию. Можно также рассматривать тонкие вопросы, касающиеся сходства и различия смыслов таких понятий как рассуждение, доказательство и вывод. Но в текущем подразделе ничего из перечисленного сделано не будет.

Не будет уточнено и формализовано понятие рассуждения. Будем надеяться, что читателя вполне удовлетворит наивное бытовое понимание этого термина. В том случае, когда будет определена некоторая формальная конструкция, соответствующая интуитивному понятию рассуждения, будем называть такую конструкцию *выводом*. Попытка формализовать само понятие рассуждения (а не его более формальный аналог — вывод), по мнению автора, неизбежно ведет к излишнему сужению объема этого понятия.

Например, можно сказать, что рассуждение — это последовательность утверждений, полученных из некоторых предпосылок (гипотез, постулатов) с помощью явно сформулированных правил. Правила в рассуждении можно применять только к ранее тем утверждениям, которые уже включены в рассуждение. Это уточнение понятия рассуждения кажется вполне приемлемым.

Однако можно сразу отметить, что рассуждение было названо последовательностью. Почему рассуждение обязательно является последовательностью? Во многих случаях рассуждение удобнее представлять в виде дерева.

Далее, насколько явным должно быть применение правил? В реальных человеческих рассуждениях всегда встречаются ситуация, когда некоторые положения используются в рассуждениях неявно. Каким образом описать применение положений, предполагаемых по умолчанию?

Еще один вопрос, который следует обсудить в связи с рассуждениями: возможная немонотонность рассуждений. Следует ли требовать, чтобы предпосылки (гипотезы, постулаты) не изменялись в процессе рассуждений? В реальных человеческих рассуждениях часто возникает такая ситуация, когда предпосылки подвергаются сомнению, опровергаются и заменяются на другие. Т.е., правила рассуждений должны позволять не только получать следствия, но и фальсифицировать и модифицировать посылки.

Все-таки сформулируем некоторое рабочее объяснение понятия рассуждения, стараясь наложить на это понятие минимум ограничений.

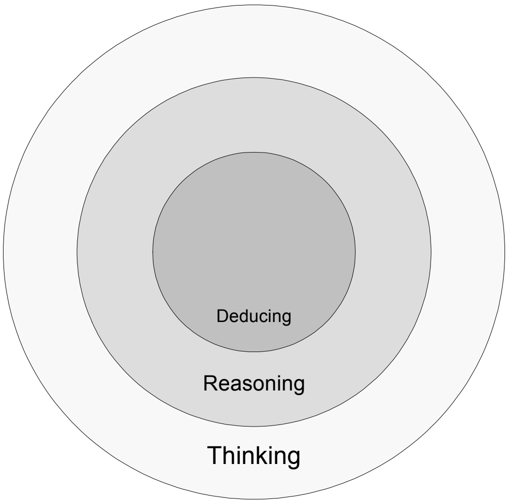


Рис. 1. Размышление, рассуждение, дедукция

Тем не менее, требуется ясно осознавать, что даже те минимальные ограничения могут оказаться излишними. Однако будем считаем, что рассуждения нельзя отождествлять с любой умственной деятельностью, reasoning ≠ thinking. Будем предполагать, что рассуждение — это процесс (и результат этого процесса), который в основном проходит под контролем сознания, тогда как в думании есть существенный компонент бессознателного/ (неосознаваемого). Также не следует отождествлять рассуждения с дедукцией (deducing), так как под дедукцией понимаются только достоверные рассуждения, где истинность посылок влечет истинность заключения. Будем считать рассуждения частным случаем думания, а дедукцию частным случаем рассуждения.

Соотношение между этими тремя понятиями представлено на Рис. 1.

Будем понимать рассуждение как процесс мыслительной деятельности, происходящей (в основном) под контролем сознания, результат которого может быть записан на естественном или формальном языке. Этот процесс начинается с принятия некоторых утверждения, называемых *предпосылками*, *гипотезами* или *постулатами*. Этот процесс включает применение *правил рассуждений*, записанных явно. Но при применении этих правил допускается неосознанное использование некоторых неявных положений, принимаемых по умолчанию. Применение правил приводит к изменению результата рассуждений *в целом*. Вариантами такого изменения могут быть: добавление следствий, удаление или модификация посылок.

Результат рассуждений, записанный на естественном или формальном языке, также называется рассуждением.

## 1.5. Правила рассуждений

Обсудим теперь правила, которые могут применяться в процессе рассуждений. Будем подразделять правила на дедуктивные и правдоподобные:

- Будем считать правило *дедуктивным*, если истинность посылок правила *обязательно* влечет истинность заключения.
- Будем относить правило к *правдоподобным*, если истинность посылок правила *не обязательно* влечет истинность заключения. Однако будем предполагать, что истинность посылок правдоподобного правила влечет заключение в *практически значимом количестве случаев*.

Подчеркнем, что требование правдоподобия рассуждений связано именно с возможностью практического использования их результатов. Давать заранее какие-то количественные ограничения, даже говорить о том, что заключение правдоподобного правила истинно в большинстве случаев, когда истинны его посылки, нельзя. Можно только сказать, что если истинны посылки правдоподобного правила, то его заключение тоже *иногда* может быть истинным. Но это нас устраивает, если помогает в практической деятельности.

Типичным дедуктивным правилом является правило *modus ponens*, формулируемое следующим образом.

$$\frac{\Phi, \quad \Phi \to \Psi}{\Psi}$$

Если истинно Φ и истинно, что из Φ следует Ψ, то Ψ также будет истинным, то есть это правило действительно является дедуктивным.

Аналогом *modus ponens* с кванторами является знаменитый *modus Barbara* Аристотеля.

$$\frac{\forall x\big(\Phi(x) \to \Psi(x)\big), \quad \Phi(a)}{\Psi(a)}$$

Самый известный пример рассуждений по модусу Barbara также принадлежит Аристотелю: «Каждый человек смертен. Сократ — человек. Следовательно, Сократ смертен» [1].

Типичным правдоподобным правилом является правило абдукции Ч.С.Пирса [4].

$$\frac{\Psi, \quad \Phi \to \Psi}{\Phi}$$

Это правило похоже на испорченный *modus ponens*. Очевидно, что это правило не является достоверным. Можно сформулировать и правило абдукции с кванторами (испорченный *modus Barbara*).

$$\frac{\forall x\left(\Phi(x) \to \Psi(x)\right), \quad \Psi(a)}{\Phi(a)}$$

Как образец рассуждений по абдукции приведем следующий пример: «Каждый спортсмен имеет развитую мускулатуру. Джон имеет развитую мускулатуру. Следовательно, Джон — спортсмен».

Действительно, это рассуждение недостоверное. Оно не соответствует правилам формальной логики. Джон не обязательно должен быть спортсменом. Он может быть, например, артистом балета. Однако в большинстве случаев, сделанное нами заключение, все-таки будет справедливо.

Данный пример основан на здравом смысле, который не всегда согласуется с формальной логикой. Ведь у того, что *Джон имеет развитую мускулатуру*, должна быть причина. И наиболее правдоподобная/очевидная причина это то, что *Джон — спортсмен*.

Приведенный пример показывает, что в правиле абдукции импликация часто (хотя и не всегда) понимается не как логическое следование, а как причинно-следственное отношение. Значит абдукцию можно использовать для поиска причин, например в технической диагностике.

В настоящее время логические формализмы, включающие абдукцию, активно развиваются. Например, абдукция используется в немонотонных логиках. Абдукция находит применение в интеллектуальных системах для решения задач обнаружения причинно-следственных связей [5].

Следующий пример популярного правила правдоподобного вывода — это правило наивной индукции. Его можно записать следующим образом.

$$\frac{\Phi(a_1),\ldots,\Phi(a_n)}{\forall x\, \Phi(x)}$$

или в форме с импликациями (условной форме),

$$\frac{\Phi(a_1) \to \Psi(a_1),\ldots,\Phi(a_n) \to \Psi(a_n)}{\forall x\left(\Phi(x) \to \Psi(x)\right)}.$$

В качестве примера рассуждений по индукции возьмем «рассуждения» *собаки Павлова*. В первый день, если загорается лампочка, то мне дают еду. Во второй день, если загорается лампочка, то мне дают еду. ... В *n*-й день, если загорается лампочка, то мне дают еду. Следовательно, каждый день, когда загорится лампочка, мне будут давать еду».

Этот пример приведен здесь не случайно. По мнению автора, индукция лежит в основе условного рефлекса, который является мощным механизмом адаптации животных к меняющимся условиям окружающей среды. Индукция позволяет обнаруживать *закономерности*, которые можно интерпретировать как *причинно-следственные отношения*. Т.е., как в случае абдукции, импликация в правиле индукции может интерпретироваться как причинно-следственное отношение.

Несмотря на свою очевидную полезность, индукция не является достоверным правилом, и заключения по индукции могут быть ложными[1].

Заметим, что в системах искусственного интеллекта форма правила индукции может быть самой разнообразной (иногда даже причудливой). Часто в индуктивных правилах присутствуют статистические компоненты. Различные правила, идуктивного характера, использующие статистические идеи, рассмотрены, например, в известной книге Гаека и Гавранека [2].

Разнообразные правила индукции, не содержащие статистических компонент, рассмотрены в работах по ДСМ-методу, предложенному В.К. Финном. Эти правила основаны на идеях Д.С. Милля [4]. Однако, заметим, что индукция в ДСМ-методе не является простой формализацией канонов Милля. Она адаптирована к решению задач поиска связи (интерпретируемой как причинно-следственная) между внутренней структурой объекта и тем фактом, что объект обладает некоторым внешним, практически важным (целевым) свойством.

[1] Всем известная математическая индукция, на самом деле индукцией не является. Она является разновидностью дедуктивной техники. Поэтому доказательства с помощью математической индукции являются достоверными.

Принципиальное отличие индукции В.К. Финна от канонов Д.С. Милля состоит в том, что Д.С. Миль неявно предполагает, что у исследуемого явления может быть *только одна причина*. Идеология ДСМ-метода допускает, что у целевого свойства может быть *много различных причин*. Кроме того, правила ДСМ-метода позволяют работать одновременно не с одним свойством, а с *множеством целевых свойств*.

Еще одним популярным правилом правдоподобных рассуждений является аналогия. Неформально, правило аналогии можно описать следующим образом: *a* похож на *b*; для *a* справедливо утверждение Ф; значит и для *b* справедливо Ф. Более формально правило аналогии можно записать так.

$$\frac{a \approx b, \quad \Phi(a)}{\Phi(b)}$$

Здесь $a \approx b$ означает «*a* похож на *b*».

Рассмотрим рассуждение по этому правилу: «Вася похож на Петю. Вася — умный. Следовательно, Петя тоже умный». Понять, насколько правдоподобно это рассуждение можно только зная, *в чем состоит сходство* Васи и Пети. Если и у Васи и у Пети оттопыренные уши — это одно, а если и Вася и Петя хорошо решают математические задачи — это совсем другое.

В.К. Финн, развивая ДСМ-метод, предложил уточнить правило аналогии. Это уточнение основано на понятии *сходства*. Сходство В.К. Финн трактует как (всё) то общее, что есть у двух объектов предметной области. Более конкретно, сходство двух объектов понимается как (наибольший) общий фрагмент их структуры.

В первом приближении, правило аналогии для ДСМ-метода выглядит следующим образом.

Пусть:
- *a* похож на *b*;
- *a* обладает свойством *p*;
- сходство *a* и *b* является возможной причиной *p*.

Тогда:
- *b* обладает свойством *p*.

Абстрагируясь от объектов, которые породили сходство, получим следующее правило.

Пусть:
- *s* является возможной причиной *p*;
- *s* содержится в *b*.

Тогда:
- *b* обладает свойством *p*.

На первый взгляд, это правило уже совсем не похоже на правило наивной аналогии. Однако, *s* в этом правиле всегда понимается как сходство каких-либо объектов, обладающих свойством *p*. Поэтому в работах по ДСМ-методу такое правило всегда называют правилом аналогии.

Итак, приведенные примеры показывают нам, что правила правдоподобного вывода могут иметь различную (иногда, экзотическую) форму. Общим для них является то, что они недостоверны, истинность посылок не гарантирует истинность заключения. Доверять результатам правдоподобных рассуждений — это всегда некоторый риск. Но на этот риск следует идти из прагматических соображений, когда заключения правил важны для принятия решения или для порождения гипотез в процессе научных исследований. Одна из главных задач разработчика когнитивных систем — это минимизация риска, связанного с недостоверностью правил[2].

## 1.6. Техника формализации

В этом подразделе будут введены такие понятия как формальный язык, теория и исчисление. Все эти понятия относятся к технике формализации рассуждений. Трудность нашей задачи обусловлена следующими обстоятельствами:

(i) В данной статье, за немногими исключениями, даются *неформальные* объяснения и приводятся наглядные примеры. Сохранить такой метод изложения материала при описании *формальных* конструкций, лежащих в основе *формализации* рассуждений, довольно трудно.

[2] Одним из способов минимизации риска являются различные процедуры фальсификации, которые также можно выразить в терминах правдопобных правил. Наличие таких процедур позволяет разрабатывать когнитивные системы, обладающие высокой степенью надежности. Например, системы, основанные на ДСМ-методе, обязательно включают процедуры фальсификации. За счет этого такие системы дают высокую точность предсказания.

(ii) Техника формализации рассуждений постоянно развивается и попытки уточнить понятия языка, теории и исчисления, ввести их в формальные рамки, может привести нас к излишнему сужению объема этих понятий. Например, ниже вводится весьма общее понятие теории. Однако для того класса теорий, которые рассматриваются в данной статье (а именно, для класса *модификационных теорий*), оно окажется слишком узким. То же самое можно сказать и о вводимых ниже понятиях (формального) языка и исчисления.

Возможным решением второй из перечисленных выше проблем является введение настолько общих версий конструкций, используемых для формализации рассуждений, которые подходили бы и для модификационных исчислений. Но такой подход приводит к слишком громоздким объяснениям. Поэтому ниже поступим следующим образом:

- сначала обсудим более традиционный/стандартный подход к понятиям языка, теории и исчисления,
- затем введем модификационные версии этих понятий.

Формальный язык включает следующие компоненты:

- **синтаксис языка:** алфавит и способы определения конструкций языка, среди которых особую роль играют *формулы* — конструкции предназначенные для представления утверждений о предметной области),
- **семантика языка:** класс таких структур, для которых *формулы можно интерпретировать как утверждения* о некоторых соотношениях в этих структурах, каждую такую структуру можно рассматривать как упрощенную модель предметной области;
- **отношение интерпретации** между структурами и формулами: если структура и формула находятся в этом отношении, то говорят, что *структура является моделью формулы* или *формула является истинной в структуре*.

Выражение $\mathsf{M} \models \varphi$ используется для того, чтобы записать тот факт, что структура M является моделью формулы φ (формула φ истинна в структуре M).

Пусть *A* — множество формул.

Тогда мы говорим, что

*структура* M *является моделью множества формул A* (записывается $\mathsf{M} \models A$),

если M является моделью каждой формулы из *A*.

Обсудим парные/двойственные понятия *класса моделей множества формул* и *теории класса структур*.

(i) Класс всех моделей множества формул *A* будем обозначать через Mod(A). Класс моделей множества формул может быть пустым. Например, он является пустым для *противоречивого* множества формул.

(ii) Если K — класс структур, то множество формул, истинных в каждой структуре из K называется *теорией класса структур* K. Теория класса структур K обозначается через Th(K).

Таким образом, теория вводится как семантическое понятие через класс моделей. Обычно, когда задается теория, описывается множество ее *аксиом*. Способ задания теории через множество аксиом вполне согласуется с приведенным выше семантическим определением.

Пусть L — формальный язык, *A* — множество формул языка L.

Мы говорим, что *T* является *теорией над языком/(в языке)* L *с множеством аксиом A*, если

$$T = \mathrm{Th}\left(\mathrm{Mod}\left(A\right)\right),$$

т.е., если *T* является теорией класса моделей множества формул *A*.

Для краткости вместо $\mathrm{Th}\left(\mathrm{Mod}\left(A\right)\right)$ будем писать просто $\mathrm{Th}\left(A\right)$.

$\mathrm{Th}\left(A\right)$ — это теория с множеством аксиом *A*. Очевидно, что $A \subseteq \mathrm{Th}\left(A\right)$.

Если $\varphi \in \mathrm{Th}\left(A\right)$, то будем говорить, что φ *семантически следует из A* (записывается $A \models \varphi$).

Элементы теории будем называть *теоремами*.

Очевидно, что справедливо следующее утверждение:

- формула φ семантически следует из *A* тогда и только тогда, когда φ является теоремой теории с множеством аксиом *A*.

Если ограничиться введенными (главным образом семантическими) средствами, то для того, чтобы установить, что формула является теоремой некоторой теории, необходимо проверить ее истинность во всех структурах некоторого класса. Понятно, что выполнять такую проверку не всегда легко. Во многих случаях это просто невозможно.

Хотелось бы сделать задачу поиска или порождения теорем решаемой более конструктивными средствами. Для этого необходимо расширить возможности нашего синтаксиса — включить в него средства для порождения теорем. Эти средства должны привести нас к определению отношения, аналогичного отношению семантического следования, только определяемого синтаксическими средствами. Это отношение впоследствии назовем отношением *выводимости*. Чтобы определить это отношение, предварительно введем понятие *исчисления*.

Точному описанию понятия исчисления предпошлем неформальное обсуждение требований, которые следует предъявить к исчислениям.

(i) Исчисление должно позволять нам определять понятие выводимости — синтаксический аналог семантического следования.

(ii) Исчисление должно предоставлять нам средства для формального представления рассуждений, в частности:

- исчисление должно содержать формальные аналоги правил рассуждений (эти формальные аналоги называются *правилами вывода*),
- исчисление должно предоставлять средства для определения формального аналога истории рассуждений (формальный аналог истории рассуждений называется *выводом* в исчислении).

Весьма общее определение исчисления можно сформулировать следующим образом.

Пусть L — формальный язык, F — множество формул языка L, S — семейство конечных подмножеств множества F.

*Исчислением над языком/(в языке)* L будем называть произвольное бинарное отношение между элементами множеств S и F.

Пусть **C** — исчисление. Предположим, что конечное множество формул *A* и формула φ находятся в отношении **C**. Как обычно в случае бинарных отношений, этот факт можно записать как

$$A\,\mathbf{C}\varphi\,.$$

В этом случае будем говорить, что φ *непосредственно выводится из* A в исчислении **C**.

В большинстве практически важных случаев, отношение непосредственной выводимости определяется *конструктивно*. Задается конечная совокупность общих схем, шаблонов, если *A* и φ соответствуют одному из таких шаблонов, то говорят, что φ непосредственно выводится из *A*. Каждая такая общая схема называется *правилом вывода исчисления* **C**.

Правила вывода, как уже было сказано выше, являются формальными аналогами правил рассуждений.

Отношение выводимости можно определить через отношение непосредственной выводимости разными способами. Некоторые из этих способов предполагают определение промежуточной/вспомогательной конструкции, которая называется *выводом в исчислении*.

Обычно, вывод — это либо конечная последовательность формул, либо конечное дерево, узлам которого ставятся в соответствие формулы. Из определения вывода можно извлечь *способ его построения*. Правила вывода в этом случае можно интерпретировать как *правила построения вывода*.

Если отношение выводимости определяется через вывод, то оно обычно определяется следующим образом.

Формула φ *называется выводимой из множества формул A в исчислении* **C**, если существует вывод в исчислении **C**, который содержит формулу φ.

Тот факт, что формула $\varphi$ выводима из множества формул $A$ в исчислении **С**, обозначается через

$$A \vdash_{\mathrm{C}} \varphi.$$

Если из контекста ясно, о каком исчислении идет речь, будем вместо $A \vdash_{\mathrm{C}} \varphi$ писать просто $A \vdash \varphi$.

Для исчисления над формальным языком L имеет смысл ставить вопрос о его *корректности* и *полноте*. Пусть **C** — исчисление над формальным языком L. Тогда

(i) Исчисление **C** называется *корректным* (относительно L), если для любого множества формул $A$ языка L и любой формулы $\varphi$ этого языка,

$$A \vdash_{\mathrm{C}} \varphi \qquad \text{влечет} \qquad A \models \varphi.$$

(ii) Исчисление **C** называется *полным* (относительно L), если для любого множества формул $A$ языка L и любой формулы $\varphi$ этого языка,

$$A \models \varphi \qquad \text{влечет} \qquad A \vdash_{\mathrm{C}} \varphi.$$

Пусть исчисление **C** является корректным и полным относительно формального языка L, $A$ — множество формул языка L, $\varphi$ — формула языка L. Тогда фомула $\varphi$ является теоремой теории с множеством аксиом $A$ , если и только если $\varphi$ выводима из $A$ в исчислении **C**.

Таким образом, в случае корректного и полного исчисления можно свести семантическое понятие теоремы теории (определяемое через истинность в моделях) через синтаксическое понятие выводимости в исчислении.

Подводя итоги, заметим, что введенные нами сейчас понятия формального языка, теории и исчисления являются весьма общими. Они подходят для широкого класса формальных языков, применяющих для определения истинности в моделях аппарат различных, в том числе неклассических логик.

Однако эти понятия не являются универсальными. Они подразумевают *монотонность* отношения выводимости, т.е. справедливость утверждения:

$$A \vdash_{\mathrm{C}} \varphi \text{ и } A \subseteq B \qquad \text{влечет} \qquad B \vdash_{\mathrm{C}} \varphi,$$

для любого исчисления **C** над языком L, любой формулы $\varphi$ этого языка и любых множеств формулы $A$ и $B$ языка L.

Желательно представить в исчислении немонотонность, которая имеет место в реальных рассуждениях. Для имитации немонотонности поступим следующим образом.

Будем снабжать формулы языка некоторыми координатами, которые можно интерпретировать как момент времени.

В качестве моделей будем рассматривать последовательности структур, индексированные упомянутыми выше координатами, т.е., каждому моменту времени будет соответствовать своя структура.

Будем говорить не просто о том, что формула $\varphi$ истинна в модели M, а о том, что формула $\varphi$ истинна в модели M *в момент времени t*.[3] Это будет означать, что формула $\varphi$ истинна в структуре с индексом $t$, содержащейся в последовательности M.

Отношение выводимости в исчислении определяется через понятие вывода, который является конечной последовательностью записей, каждая из которых включает следующие поля:

формулу,

основание, на котором формула была включена в вывод (обычно, ссылку на правило вывода),

временные координаты (момент времени),

признак активности данной записи.

Вывод определяется таким образом, что, добавляя текущую запись к выводу, можно одновременно делать некоторые из предыдущих записей неактивными. Это позволяет, фактически, заменять в выводе одни формулы на другие.

Также как в случае истинности в модели, не будем говорить, что формула $\varphi$ выводима в исчислении $\mathrm{m}$ из множества формул $A$, а будем говорить, что формула $\varphi$ выводима в исчислении $\mathrm{m}$ из множества формул $A$ *в момент времени t*.

Теорию будем определять синтаксически как упорядоченную пару $\langle A, \mathrm{m} \rangle$, где $A$ — множество формул, $\mathrm{m}$ — исчисление.

[3] Из того, что формула $\varphi$ истинна в модели M в момент времени $t$ не следует ее истинность в этой модели в более поздние моменты времени.

Языки, исчисления и теории, которые удовлетворяют перечисленным выше условиям и рассматриваются в данной статье, в этой статье называются *модификационными*. Это название связано с тем фактом, что правила вывода не просто продолжают/расширяют вывод, но и изменяют (модифицируют) те записи, которые уже были в выводе.

Модификационные исчисления и модификационные теории оказались весьма удобными для представления динамики когнитивного процесса. Более точно, модификационными теориями представляется часть когнитивного процесса, а именно, процесс рассуждений.

Напомним, что операции, которые можно проделывать над выводом в модификационном исчислении, можно интерпретировать как замену в выводе одних формул на другие. Эта замена может касаться и аксиом модификационной теории. Поэтому вывод можно рассматривать как *историю развития модификационной теории*.

Модификационные исчисления являются корректными и полными относительно своих языков в некотором специальном смысле. Соответствующие теоремы о корректности и о полноте доказываются в последующих статьях.

Для модификационных исчислений справедливы некоторые варианты *теорем о корректности и о полноте*, которые доказываются в последующих статьях.

## 2. Схема когнитивных рассуждений

В этом разделе будет более подробно рассмотрена структура процесса рассуждений и средства формализации рассуждений модификационные теории и исчисления.

В этом разделе обсуждается класс логических алгебр, служащих фундаментом для построения модификационных исчислений. Эти алгебры имеют два сорта истинностных значений (внутренние и внешние). Интуитивный смысл внешних и внутренних истинностных значений подробно рассматривался в подразделе “Истинностные значения: Неформальное обсуждение”.

Будет описан также подход к когнитивным рассуждениям как к движению от незнания к знанию. Динамика процесса когнитивных рассуждений, поддерживаемых модификационными теориями, как процесса уменьшения количества неопределенности в данных и знаниях.

Будет неформально рассматриваться построение модификационных теорий, модификационных исчислений, вывода в модификационных исчислениях. Будет рассмотрена связь между выводом в модификационном исчислении и алгоритмом работы познающего субъекта.

### 2.1. Логические основания

Определение истинности формулы формального языка в структуре использует операции в логической алгебры, а сами структуры включают предикаты, которые являются функциями, принимающими истинностные значения.

Те логические алгебры, которые подходят для того, чтобы служить основаниями для модификационных языков, исчислений и теорий являются двусортными алгебрами. Они имеют два сорта истинностных значений. Это сорт внешних истинностных значений (Ext) и сорт внутренних истинностных значений (Int). Множество внешних истинностных значений будем обозначать через $V^{\mathrm{Ext}}$, множество внутренних истинностных значений — через $V^{\mathrm{Int}}$.

Множество внешних истинностных значений имеет два элемента **t** (true) и **f** (false).

Множество внутренних истинностных значений конечно, содержит по крайней мере два элемента, один из которых интерпретируется как uncertain и обозначается через $\tau$. Что касается других истинностных значений, то будем считать, что они выражают различные оттенки определенности.

На множестве внешних истинностных значений $V^{\mathrm{Ext}}=\{\mathbf{t},\mathbf{f}\}$ будут определены обычные логические операции (логические связки). Для определенности будем считать, что на этом множестве определено минимальное функционально полное множество логических операций $\{\rightarrow,\neg\}$ (импликация и отрицание).

На множестве внутренних истинностных значений определены J-операторы из множества

$$\mathrm{J}\left(V^{\mathrm{Int}}\right)=\left\{\mathrm{J}_\alpha \mid \alpha\in V^{\mathrm{Int}}\right\}.$$

Для каждого $\alpha\in V^{\mathrm{Int}}$, $\mathrm{J}_\alpha$ есть отображение множества $V^{\mathrm{Int}}$ в множество $V^{\mathrm{Ext}}$ такое, что для любого $\beta\in V^{\mathrm{Int}}$,

$$J_{\alpha}(\beta)=\begin{cases}\mathbf{t}, & \text{if } \beta=\alpha,\\ \mathbf{f}, & \text{if } \beta\neq\alpha.\end{cases}$$

J-оператор — это характеристическая функция внутреннего истинностного значения.

Алгебру

$$\mathsf{S}=\left\langle V^{\mathrm{Ext}},V^{\mathrm{Int}};\{\rightarrow,\neg\}\cup J\left(V^{\mathrm{Int}}\right)\right\rangle$$

будем называть *алгеброй логики состояния*.

Название логика состояния связано с тем, что с помощью этой логики предполагается описывать стационарное *состояние знания* в период спокойного течения процесса рассуждений. Также как в когнитивном процессе в целом, в процессе рассуждений будут чередоваться две фазы: медленная и быстрая. Но об этом подробно будем говорить в следующих подразделах этого раздела.

Обычно в качестве примера будем брать алгебру логики состояния с четырехэлементным множеством внутренних истинностных значений, обозначаемых через +1, –1, 0 и $\tau$.

В этой статье не будет подробно рассматриваться формальный язык модификационных теорий. Скажем только, что каждый такой язык предназначен для исчерпывающего описания некоторой конечной многосортной и многозначной структуры, аналогичной рассмотренным в подразделе «Сорта и структуры».

Такую структуру будем называть начальной/исходной структурой/прототипом. На структуру-прототип накладываются следующие ограничения:

- она должна быть конечной, т.е. множество объектов всех доступных сортов этой структуры конечно,
- множество внутренних предикатов структуры-прототипа конечно.

Приступим к описанию языка модификационной теории. Сначала сформулируем систему требований к алфавиту этого языка.

(i) Алфавит языка модификационной теории включает имена сущностей (различных типов), относящихся к структуре-прототипу. Каждая сущность (объект, операция, предикат), относящаяся к структуре-прототипу имеет свое имя, разные сущности имеют разные имена.

(ii) Имена объектов называются *константами*, имена операций — *функциональными символами*, имена предикатов — *предикатными символами*.

(iii) Предикатные символы делятся на два класса: *внешних и внутренних*, в зависимости от того, именем какого предиката (внешнего или внутреннего) является символ.

(iv) Для каждого сорта объектов алфавит содержит бесконечное множество переменных.

(v) Алфавит содержит символы логических связок $\rightarrow$, $\neg$, $J_{\alpha}$ $\left(\alpha\in V^{\mathrm{Int}}\right)$, квантора общности $\forall$. (Подразумевается, что все прочие логические связки и квантор существования $\exists$ могут быть введены по определению.)

(vi) Алфавит содержит вспомогательные символы: скобки "(" и ")" и запятую ",".

Расскажем теперь об основных конструкциях языка модификационной теории.

(i) Как обычно определяются *термы* и *атомарные формулы* языка модификационной теории. Атомарные формулы имеют вид $p(t_1,...,t_n)$, где p — предикатный символ, $t_1,...,t_n$ — термы или $t_1=t_2$, где $t_1$ и $t_2$ — термы одного сорта.

(ii) Атомарные формулы делятся на два класса — *внешние и внутренние* в соответствии с тем, какой предикатный символ (внешний или внутренний) входит в атомарную формулу. Атомарные формулы вида $t_1=t_2$ считаются внешними.

(iii) Выражения вида $J_{\alpha}\varphi$, где $\varphi$ — внутренняя атомарная формула называются J-*атомарными формулами*.

(iv) Из внешних атомарных формул и J-атомарных формул с помощью логических связок и кванторов строятся *внешние формулы*.

Именно внешние формулы будут играть в модификационных теориях ту роль, которую обычно играют формулы. Только внешние формулы могут:

- быть истинными в моделях,
- содержаться в выводе,
- быть теоремами модификационных теорий.

В дальнейшем, если в контексте модификационных теорий или модификационных исчислений будет употребляться слова «формула», будем всегда подразумевать (если не оговорено противного), что это внешняя формула.

Внешние формулы могут принимать только внешние истинностные значения **t** и **f**.

Наибольший интерес для нас будут представлять J-атомарные формулы. С помощью подходящего набора J-атомарные формул можно получить исчерпывающее описание внутреннего предиката. Именно на работе с описаниями внутренних предикатов будет основан наиболее важный компонент вывода в модификационных исчислениях.

## 2.2. От незнания к знанию

Предлагаемый подход предполагает интерпретацию процесса рассуждений, как движения от неопределенности а определенности (от незнания к знанию).

Процесс является дискретным, он разбит на стадии (шаги). С каждой новой стадией область неопределенности сужается. Будем предполагать, что область неопределенности состоит из некоторых утверждений, описывающих факты или закономерности, формат которых заранее известен (Рис. 2).

Если ограничиться конечными структурами, то когнитивный процесс, рассматриваемый как движение от незнания к знанию, обязательно должен завершиться. Признаком завершения является так называемая *стабилизация* или *насыщение* — такая ситуация, когда уже невозможно добиться сужения области неопределенности. В конечной структуре и область неопределенности конечна, и она не может бесконечно сужаться.

Наглядное изображение когнитивного процесса со стабилизацией дано на Рис. 1. Область неопределенности изображена в виде белого пятна на карте. В результате когнитивного процесса площадь белых пятен сокращается. Неопределенность заменяется не определенность какого либо характера (определенно да, определенно нет и т.п.). Этому на рисунке соответствует закрашивание областей карты в цвета, соответствующие характеру неопределенности.

## 2.3. Правила модификации

Инструментом для реализации подхода к рассуждениям как к движению «от незнания к знанию» являются правила модификации. Правила модификации можно понимать как своеобразную интерпретацию формализованных правил правдоподобных рассуждений.

Объясним преобразование правила рассуждения в правило модификации с помощью примера. Рассмотрим следующее правило.

Пусть
- существует хотя бы один эксперимент, где вслед за событием *c* наблюдается событие *e*,
- не существует ни одного эксперимента, где вслед за событием *c* не наблюдается событие *e*.

Тогда
- событие *c* является возможной причиной события *e*.

Запишем это правило формально, используя имена предикатов структуры из подраздела Sorts and structures.

$$\frac{\exists n\left(\mathbf{Exp}(n,c,e)\right),\quad \neg\exists m\left(\mathbf{Exp}(m,c,e)\right)}{\mathbf{PC}(c,e)}$$

Заметим, что наши предикаты внутренние. А статус формул у нас имеют только внешние формулы. Поэтому в правиле внутренние атомарные формулы должны быть заменены на J-атомарные, а именно:

- утверждение «в эксперименте *x* вслед за событием *c* наблюдается событие *e*» должно быть записано как $\mathrm{J}_{+1}\,\mathbf{Exp}(x,c,e)$,
- утверждение «в эксперименте *x* вслед за событием *c* наблюдается событие *e*» должно быть записано как $\mathrm{J}_{-1}\,\mathbf{Exp}(x,c,e)$,
- утверждение «*c* является возможной причиной (наступления) события *e*» должно быть записано как $\mathrm{J}_{+1}\,\mathbf{PC}(c,e)$.

Вспомним, что требуется сформулировать правило модификации (т.е. изменения истинностного значения), причем реализовать подход «движение от неопределенности к определенности (от незнания к знанию)», т.е. до применения правила значение предиката $\mathbf{PC}$ должно равняться $\tau$, а после применения правила оно станет равным +1. Чтобы отличать правила модификации от обычных правил вывода, будем отделять в правилах модификации посылки

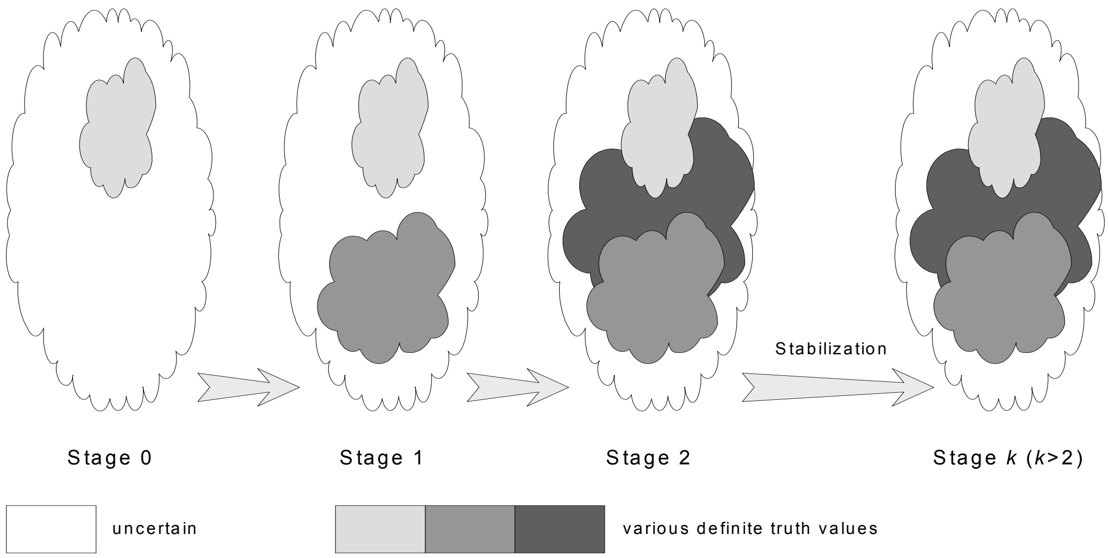


Рис. 2. Сужение области неопределенности и стабилизация

от заключения двойной чертой. Приведенное выше правило рассуждения превратится в следующее правило модификации.

$$\frac{J_{\tau}\,\mathbf{PC}(c,e),\quad \exists n\left(J_{+1}\,\mathbf{Exp}(n,c,e)\right)\&\neg\exists m\left(J_{-1}\,\mathbf{Exp}(m,c,e)\right)}{J_{+1}\,\mathbf{PC}(c,e)}$$

Назовем это правила +*Правилом для* **PC**. Название связано с тем фактом, что заключение правила содержит J-оператор $J_{+1}$ и предикатный символ **PC**.

Общий вид правила модификации следующий

$$\frac{J_{\tau}\,\varphi,\quad \Psi}{J_{\alpha}\,\varphi},$$

где $\varphi$ — внутренняя атомарная формула, $\Psi$ — внешняя формула. Формула $J_{\tau}\,\varphi$ называется *малой посылкой*, $\Psi$ — *большой посылкой*, $J_{\alpha}\,\varphi$ — *заключением правила*.

Если интерпретировать такое правило как правило вывода, можно легко прийти к противоречию. Например, если $\alpha=+1$, то после применения этого правила получим формулы $J_{\tau}\,\varphi$ и $J_{+1}\,\varphi$, которые не могут быть одновременно истинны, так как внутренняя атомарная формула не может одновременно равняться $\tau$ и +1.

Наиболее удобная и наглядная неформальная интерпретация правила модификации — это представление его в виде условного оператора.

$$\textbf{If}\ \ \varphi=\tau\ \ \textbf{And}\ \ \Psi'\ \ \textbf{Then}\ \ \textbf{Let}\ \varphi=\alpha.$$

Здесь условие $\Psi'$ получено из большой посылки $\Psi$ правила модификации заменой каждой J-атомарной формулы вида $J_{\beta}\,\theta$ на равенство $\theta=\beta\ \ \left(\beta\in V^{\mathrm{Int}}\right)$. Например +*Правилу для* **PC** будет соответствовать оператор присваивания

$$\textbf{If}\ \ \mathbf{PC}(c,e)=\tau\ \ \textbf{And}\ \ \mathrm{E}^{+}(c,e)$$

$$\textbf{Then}\ \ \textbf{Let}\ \mathbf{PC}(c,e)=+1,$$

где условие $\mathrm{E}^{+}(c,e)$ можно выразить следующим образом:

$$\exists n\left(\mathbf{Exp}(n,c,e)=+1\right)\&\neg\exists m\left(\mathbf{Exp}(m,c,e)=-1\right)$$

В случае программной реализации на процедурном языке, логическая функция, вычисляющая значение условия $\mathrm{E}^{+}(c,e)$ будет содержать циклы для обхода области определения предиката $\mathbf{Exp}$.

Неформальная интерпретация правил модификации как условных операторов очень удобна для объяснения процессов, происходящих при работе рассуждателя.

Все правили модификации суть правила для замены истинностного значения $\tau$ в таблице для внутреннего предиката на одно из внутренних истинностных значений при соблюдении условия, выраженного в большой посылке правила.

Выше было приведено правило для замены τ на +1 в таблице для внутреннего предиката **PC** (+Правило для **PC**). Сформулируем теперь остальные правила для обработки этого предиката. Сначала договоримся о неформальной интерпретации J-атомарных формул с предикатным символом **PC** (возможная причина). Выше уже было сказано, что

- $\mathrm{J}_{+1}\,\mathbf{PC}(c,e)$ интерпретируется, как «*c* является возможной причиной наступления (присутствия) события *e*».

Теперь дадим неформальную интерпретацию других J-атомарных формул с предикатным символом **PC**:

- $\mathrm{J}_{-1}\,\mathbf{PC}(c,e)$ интерпретируется, как «*c* является возможной причиной ненаступления (отсутствия) события *e*»;
- $\mathrm{J}_{0}\,\mathbf{PC}(c,e)$ интерпретируется, как «сведения о причинно-следственных отношения между *c* и *e* противоречивы»;
- $\mathrm{J}_{\tau}\,\mathbf{PC}(c,e)$ интерпретируется, как «сведений о причинно-следственных отношения между *c* и *e* недостаточно».

Дадим сначала неформальную запись правила *–Правила для* **PC**.

Пусть

- существует хотя бы один эксперимент, где вслед за событием *c* не наблюдается событие *e*,
- не существует ни одного эксперимента, где вслед за событием *c* наблюдается событие *e*.

Тогда

- событие *c* является возможной причиной ненаступления (отсутствия) события *e*.

Формальная запись *–Правила для* **PC** будет иметь следующий вид:

$$\frac{\mathrm{J}_{\tau}\,\mathbf{PC}(c,e),\quad \exists n\left(\mathrm{J}_{-1}\,\mathbf{Exp}(n,c,e)\right)\&\neg\exists m\left(\mathrm{J}_{+1}\,\mathbf{Exp}(m,c,e)\right)}{\mathrm{J}_{-1}\,\mathbf{PC}(c,e)}$$

Приведем неформальную запись 0*Правила для* **PC**.

Пусть

- существует хотя бы один эксперимент, где вслед за событием *c* наблюдается событие *e*,
- существует хотя бы один эксперимент, где вслед за событием *c* не наблюдается событие *e*.

Тогда

- сведения о причинно-следственнвх отношения между *c* и *e* противоречивы.

Формальная запись 0*Правила для* **PC** будет иметь следующий вид:

$$\frac{\mathrm{J}_{\tau}\,\mathbf{PC}(c,e),\quad \exists n\left(\mathrm{J}_{+1}\,\mathbf{Exp}(n,c,e)\right)\&\exists m\left(\mathrm{J}_{-1}\,\mathbf{Exp}(m,c,e)\right)}{\mathrm{J}_{0}\,\mathbf{PC}(c,e)}$$

Рассмотрим теперь неформальную запись τ*Правила для* **PC**.

Пусть

- не существует ни одного эксперимента, где вслед за событием *c* наблюдается событие *e*,
- не существует ни одного эксперимента, где вслед за событием *c* не наблюдается событие *e*.

Тогда

- сведений о причинно-следственных отношениях между *c* и *e* недостаточно.

Формальная запись τ*Правила для* **PC** будет иметь следующий вид:

$$\frac{\mathrm{J}_{\tau}\,\mathbf{PC}(c,e),\quad \neg\exists n\left(\mathrm{J}_{+1}\,\mathbf{Exp}(n,c,e)\right)\&\neg\exists m\left(\mathrm{J}_{-1}\,\mathbf{Exp}(m,c,e)\right)}{\mathrm{J}_{\tau}\,\mathbf{PC}(c,e)}$$

Теперь продемонстрируем применение этих правил модификации на примере. Рассмотрим некоторый вариант структуры с двумя внутренними предикатами **Exp** и **PC**, рассмотренной в подразделе Sorts and structures. От структуры, изображенной на Табл. 3 эта структура будет отличаться только таблицей для предиката **PC**. Будем предполагать, что нам ничего не известно про возможные причины событий. Итак, таблицы для внутренних предикатов рассматриваемой структуры представлены Табл. 4.

Табл. 4. Внутренние предикаты до применения модификационных правил

**Exp**

| | Arity | | Sort |
|---|---|---|---|
| N | C | E | Int |
| 1 | **A** | **P** | +1 |
| 1 | **A** | **Q** | +1 |
| 1 | **B** | **P** | $\tau$ |
| 1 | **B** | **Q** | –1 |
| 2 | **A** | **P** | +1 |
| 2 | **A** | **Q** | –1 |
| 2 | **B** | **P** | $\tau$ |
| 2 | **B** | **Q** | –1 |

**PC**

| Arity | | Sort |
|---|---|---|
| C | E | Int |
| **A** | **P** | $\tau$ |
| **A** | **Q** | $\tau$ |
| **B** | **P** | $\tau$ |
| **B** | **Q** | $\tau$ |

Значения предиката **PC** для всех пар из области определения равны $\tau$ (неопределенно). К каждой строчке таблицы для **PC** применимо одно и только одно правило модификации для **PC**. Применив эти правила, преобразуем таблицу для предиката **PC** и получим в результате точно такую же таблицу для **PC**, что изображено в Табл. 5.

Табл. 5. Результат применения правил модификации

**PC**

| Arity | | Sort |
|---|---|---|
| C | E | Int |
| **A** | **P** | +1 |
| **A** | **Q** | 0 |
| **B** | **P** | $\tau$ |
| **B** | **Q** | –1 |

Правила модификации для структуры, содержащей конечное число внутренних предикатных символов должны быть объединены в *систему правил модификации*. Система правил модификации для такой структуры должна удовлетворять следующим требованиям:

(i) для каждого внутреннего предиката *P* и каждого внутреннего истинностного значения $\alpha$ система содержит в точности одно правило модификации, заключение которого содержит предикатный символ *P* и J-оператор $\mathrm{J}_{\alpha}$;

(ii) для каждого внутреннего предиката *P*, большие посылки различных правил модификации противоречат друг другу/несовместны.

Следующее требование к правилам модификации не является обязательным, но обычно оно соблюдается:

(iii) для каждого внутреннего предиката *P* и каждого элемента его области определения истинна большая посылка хотя бы одного правила модификации для этого предиката.

Нетрудно показать, что условия (i), (ii), (iii) справедливы для предиката **PC** из приведенного выше примера.

Продолжим рассмотрение нашего примера, т.е. структуры, представленной в Табл. 4. Выше были сформулированы правила для предиката **PC**. Теперь сформулируем правила модификации для предиката **Ext**. Как и ранее, сначала будем давать неформальную запись правила рассуждения, затем формальную запись соответствующего ему правила модификации.

+Правило для **Ext**

Пусть

- *c* является возможной причиной наступления (наличия) *e*.

Тогда

- в эксперименте *x* вслед за событием *c* происходит событие *e*.

Формальная запись +*Правила для* **Ext** будет иметь следующий вид:

$$\frac{\mathrm{J}_{\tau}\,\mathbf{Exp}(x,c,e),\quad \mathrm{J}_{+1}\,\mathbf{PC}(c,e)}{\mathrm{J}_{+1}\,\mathbf{Exp}(x,c,e)}$$

Правило для **Ext**

Пусть

- *c* является возможной причиной не наступления (отсутствия) *e*.

Тогда

- в эксперименте *x* вслед за событием *c* не происходит событие *e*.

Формальная запись *Правила для* **Ext** будет иметь следующий вид:

$$\frac{\mathrm{J}_{\tau}\,\mathbf{Exp}(x,c,e),\quad \mathrm{J}_{-1}\,\mathbf{PC}(c,e)}{\mathrm{J}_{-1}\,\mathbf{Exp}(x,c,e)}$$

Правило для **Ext**

Пусть

- сведения о причинно-следственных отношениях между *c* и *e* противоречивы.

Тогда

- в эксперименте *x* вслед за событием *c* событие *e* может как наступить, так и не наступить.

Формальная запись *Правила для* **Ext** будет иметь следующий вид:

$$\frac{\mathrm{J}_{\tau}\,\mathbf{Exp}(x,c,e),\quad \mathrm{J}_{0}\,\mathbf{PC}(c,e)}{\mathrm{J}_{0}\,\mathbf{Exp}(x,c,e)}$$

τПравило для **Ext**

Пусть

- сведений о причинно-следственных отношениях между *c* и *e* недостаточно.

Тогда

- мы ничего не можем сказать о том, наступает или не наступает в эксперименте *x* вслед за событием *c* событие *e*.

Формальная запись τ*Правила для* **Ext** будет иметь следующий вид:

$$\frac{\mathrm{J}_{\tau}\,\mathbf{Exp}(x,c,e),\quad \mathrm{J}_{\tau}\,\mathbf{PC}(c,e)}{\mathrm{J}_{\tau}\,\mathbf{Exp}(x,c,e)}$$

Таблица для предиката **Ext** (Табл. 4) содержит только две строчки, где значение предиката равно τ. К этим строчкам применимо τ*Правило для* **Ext** (Табл. 5). В результате применения этого правила таблица для предиката **Ext** не изменится.

Заметим, что сначала правила применяются к предикату **PC**, затем к предикату **Ext**. Другой порядок применения правил будет неестественным. Мы сначала должны выявить закономерность, затем сделать предсказание (сформировать гипотезу) о наступлении или не наступлении события.

Порядок обработки (обхода) внутренних предикатов является частью системы правил модификации и существенным компонентом модификационного исчисления и модификационной теории.

## 2.4. Описания состояний

В предыдущем подразделе был рассмотрен один из важных компонентов модификационного исчисления — систему правил модификации. В этом подразделе рассматривается еще один важный компонент — описание состояния. Название «описание состояния» связано с той идеей, что при движении от незнания к знанию в процессе рассуждений, познающий субъект проходит через ряд относительно стабильных *состояний знания*.

Описание состояние — это совокупность формул, которые будут играть роль аксиом модификационного исчисления.[4] Вообще говоря, описание состояния — это полное и однозначное описание структуры-прототипа (термин структура-прототип введен в подразделе Logical background). Описание состояние имеет единственную (с точностью до изоморфизма) модель. Эта модель и есть структура-прототип.

Описание состояния делится на несколько групп аксиом. Рассмотрим аксиомы описания состояния на примере описания структуры-прототипа, представленной в Табл. 4.

- Группа *инъективности* гарантирует, что любой объект предметной области имеет не более одного имени, т.е., разным константам (именам) одного и того же сорта обязательно должны соответствовать разные объекты:
  - $1 \neq 2$,
  - $\mathbf{A} \neq \mathbf{B}$,
  - $\mathbf{P} \neq \mathbf{Q}$.
- Группа *сюръективности* гарантирует, что каждый объект предметной области имеет хотя бы одно имя, т.е. не существует объектов, кроме поименованных посредством констант словаря:

[4] В параграфе «Техника формализации» мы определили понятие исчисления таким образом, что оно не подразумевает наличия собственных аксиом исчисления. Исчисление определяется только набором правил. В соответствии с этим определением собственные аксиомы исчисления следует интерпретировать как правила с пустым множеством посылок.

– $\forall n(n = 1 \vee n = 2)$,
– $\forall c(c = \mathbf{A} \vee c = \mathbf{B})$,
– $\forall e(e = \mathbf{P} \vee e = \mathbf{Q})$.

Здесь $n$, $c$, $e$ суть переменные сортов N, C, E, соответственно.

- Группа *описание операций* содержит пустое множество аксиом, так как структура-прототип из Табл. 4 не имеет операций.
- Группа *описание предикатов* делится на две подгруппы (по количеству предикатов):
  – подгруппа *описание предиката* **Exp** содержит следующие аксиомы:
  $J_{+1}\,\mathbf{Exp}(1,\mathbf{A},\mathbf{P})$, $J_{+1}\,\mathbf{Exp}(1,\mathbf{A},\mathbf{Q})$,
  $J_{\tau}\,\mathbf{Exp}(1,\mathbf{B},\mathbf{P})$, $J_{-1}\,\mathbf{Exp}(1,\mathbf{B},\mathbf{Q})$,
  $J_{+1}\,\mathbf{Exp}(2,\mathbf{A},\mathbf{P})$, $J_{-1}\,\mathbf{Exp}(2,\mathbf{A},\mathbf{Q})$,
  $J_{\tau}\,\mathbf{Exp}(2,\mathbf{B},\mathbf{P})$,
  – подгруппа *описание предиката* **PC** содержит следующие аксиомы:
  $J_{\tau}\,\mathbf{PC}(\mathbf{A},\mathbf{P})$, $J_{\tau}\,\mathbf{PC}(\mathbf{A},\mathbf{Q})$,
  $J_{\tau}\,\mathbf{PC}(\mathbf{B},\mathbf{P})$, $J_{\tau}\,\mathbf{PC}(\mathbf{B},\mathbf{Q})$ (Табл. 4).

Теперь у нас есть все необходимые понятия для того, чтобы определить модификационное исчисление.

## 2.5. Модификационные исчисления и модификационные теории

Модификационное исчисление определяется как упорядоченная четверка

$$\mathsf{M} = \langle V, S, M, O \rangle,$$

где $V$ — словарь констант, обозначающих объекты структуры-прототипа, каждому объекту соответствует в точности одна константа,
$S$ — описание состояния, множество формул, содержащих константы из $V$, полностью и однозначно описывающее структуру-прототип,
$M$ — система правил модификации, подходящая для внутренних предикатов структуры-прототипа,
$O$ — инъективная функция, ставящая в соответствие каждому внутреннему предикату структуры-прототипа порядковый номер, начинающийся с 1.

Вообще говоря, структура-прототип в явном виде не входит в строгое формальное определение модификационных исчислений. Но она может быть легко построена с помощью описания состояния.

Модификационная теория определяется как упорядоченная пара

$$T = \langle A, \mathsf{M} \rangle,$$

где M — модификационное исчисление,
$A$ — множество аксиом модификационной теории, каждая аксиома является внешней формулой, которую можно интерпретировать как утверждение об объектах структуры-прототипа модификационного исчисления M.

Модификационную теорию можно интерпретировать как расширение модификационного исчисления. Следует заметить, что аксиомы модификационной теории не могут принести ничего принципиально нового, если они не противоречат аксиомам из описания состояния. Описание состояния является исчерпывающим для структуры-прототипа. Все, что можно доказать о структуре-прототипе можно доказать, опираясь только на описание состояния.

Однако аксиомы модификационной теории, содержащие общие утверждения могут сильно ускорить процесс получения следствий. Это обстоятельство особенно важно для автоматизации рассуждений.

В качестве примера можно рассмотреть модификационное исчисление для структуры-прототипа, представленной в Табл. 4.

- Словарь $V = \{1, 2, \mathbf{A}, \mathbf{B}, \mathbf{P}, \mathbf{Q}\}$.
- Описание состояния $S$ есть множество аксиом из подраздела State descriptions.
- Система правил модификации $M$ для этого модификационного исчисления подробно описана в подразделе Modification rules.
- Функция $O$ такова, что $O(\mathbf{PC}) = 1$, $O(\mathbf{Ext}) = 2$

## 2.6. Алгоритм рассуждений: семантический подход

Процесс когнитивных рассуждений можно интерпретировать как процесс преобразования структуры-прототипа с помощью правил модификации. Описание алгоритма для этого процесса включает ряд вложенных циклов.

Опуская подробности, алгоритм когнитивных рассуждений можно представить так, как это сделано на Рис. 3. На этом рисунке через P[i] обозначен внутренний предикат с порядковым номером i. Через N обозначено количество внутренних предикатных символов.

Процесс рассуждений разбит на *стадии*. Номер стадии на Рис. 3 обозначен через s. На нулевой стадии просто принимаются к сведению факты, описывающие соответствующие предикаты. На всех следующих стадиях происходит обработка внутренних предикатов. Эта обработка происходит в том порядке, который задан порядковыми номерами предикатов в модификационном исчислении (т.е. функцией *O*). Период обработки одного предиката называется *модулем*.

Обработка предиката состоит в применении правил модификации к тем строкам таблицы, представляющей предикат, в которых содержится значение τ (неопределено). Правила модификации позволяют заменять значение «неопределено» на какое либо определенное значение. Область неопределенности предиката при этом сужается (движение от незнанию к знанию).

Процесс рассуждений заканчивается при достижении стабилизации (условие Stabilisation_Flag = TRUE). Стабилизация достигается тогда, когда к строкам таблиц, входящим в область неопределенности предиката, становится возможным применять только τПравила, сохраняющие неопределенные значения.

## Вывод

Описанный выше алгоритм находит отражение в понятии вывода в модификационном исчислении. Строгое определение вывода в модификационном исчислении достаточно громоздко, поэтому в этой статье оно не приводится. Будет рассказано о структуре вывода и приведен пример короткого вывода в модификационном исчислении, приведенном в качестве примера в подразделе Modification calculi and modification theories.

Вывод в модификационном исчислении представляет собой конечную последовательность записей (кортежей). Каждая запись в выводе содержит пять полей:

− внешнюю формулу,

− основание, по которому эта формула включена в вывод,

− номер стадии, внутри которой произошло добавление записи в вывод,

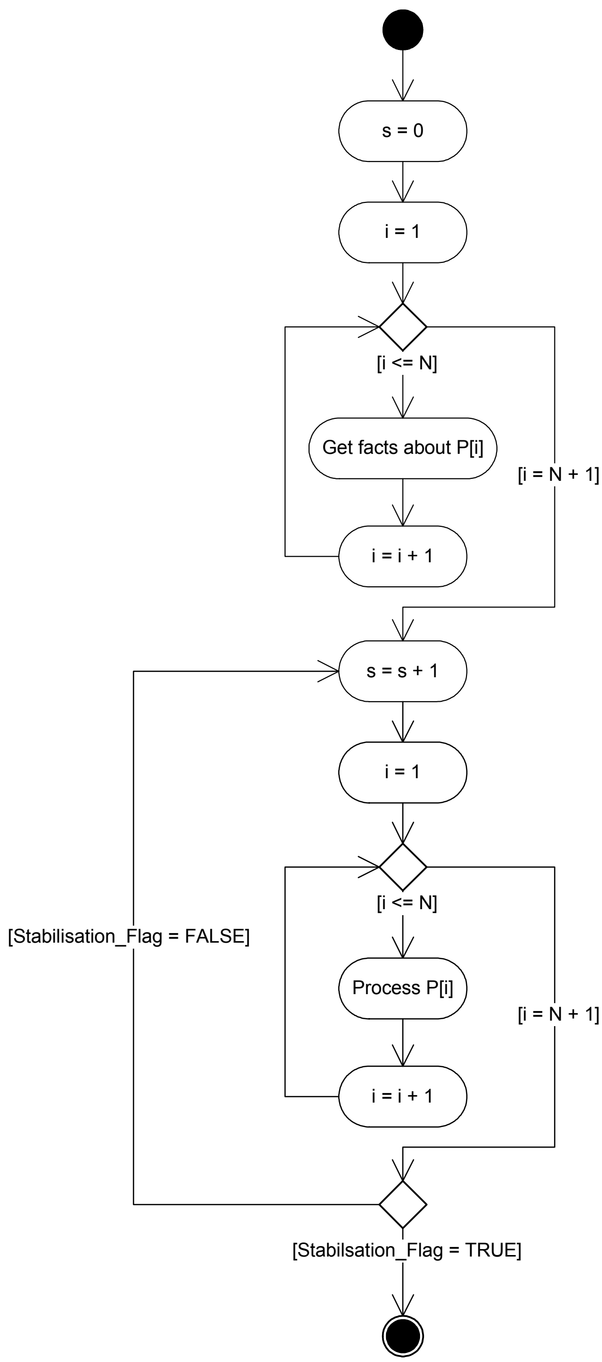


Рис. 3. Алгоритм когнитивных рассуждений

– номера модуля, внутри которого произошло добавление записи в вывод,

– порядковый номер того элемента вывода, который деактивировал (скрыл) данную запись (если запись не деактивирована, то это поле равно 0).

Под модулем будем понимать процесс обработки одного внутреннего предиката. Стадия включает обработку всех внутренних предикатов структуры. Порядок обработки предикатов определяется модификационным исчислением. Этот порядок один и тот же у всех выводов в одном модификационном исчислении. Он не может измениться внутри вывода.

Вывод можно представить в виде таблицы, каждая строка которой представляет запись. Для удобства будем дописывать к этой таблице слева столбец с порядковыми номерами строк.

Вывод разбит на стадии. Число стадий не ограничено. Каждая стадия разбита на модули. Количество модулей равно количеству внутренних предикатов структуры-прототипа. В каждом модуле можно выделить две фазы, но некоторые фазы могут оказаться пустыми.

Первая фаза модуля называется *фазой модификации*. В этой фазе разрешено применять как обычные дедуктивные правила, так и правила модификации. Но правила модификации можно применять только к текущему предикату модуля. Фаза модификации заканчивается, когда в выводе присутствуют результаты применения правил модификации к каждому элементу из области неопределенности текущего предиката модуля.

Применение правила модификации не сводится к просто к добавлению записи в вывод. Если в результате применения такого правила значение $\tau$ будет заменено на определенное значение, то должна быть деактивирована строка вывода, содержащая малую посылку этого правила (т.е. формулу вида $J_\tau \varphi$). Также необходимо деактивировать все строки, *дедуктивно зависимые* от строки с малой посылкой, т.е. те строки, которые были получены с помощью только дедуктивных правил из строки с малой посылкой.

Вторая фаза модуля наступает после завершения фазы модификации. Она называется *фазой дедукции*. В этой фазе можно использовать только обычные дедуктивные правила.

Фазу модификации можно интерпретировать как *быструю* фазу рассуждений, так как применение модификационных правил (аналогов правил правдоподобных рассуждений) может привести к новым, неожиданным результатам.

Фазу дедукции можно толковать как *медленную* фазу рассуждений, так как, используя только достоверные/дедуктивные правила, невозможно получить ничего принципиально нового.

Новый модуль начинается с первого применения правила модификации к следующему по порядку предикату. И начинается он с фазы модификации.

Если область неопределенности какого либо внутреннего предиката окажется пустой, то обработки этого предиката не происходит (происходит переход к следующему предикату) и обе фазы соответствующего модуля считаются пустыми.

Особую роль играет нулевая стадия вывода. Эту стадию можно назвать стадией интродукции/инициализации. На нулевой стадии правила модификации не применяются. Вместо этих правил применяются аксиомы из описания состояний, описывающие текущий предикат модуля.

Каждый модуль нулевой стадии также разбивается на две фазы. Но модификационная фаза в нулевой стадии является, фактически, стадией добавления фактов, описывающих текущий предикат модуля. После того, как область неопределенности текущего предиката описана (добавлены все аксиомы вида $J_\tau \varphi$, где $\varphi$ содержит символ текущего предиката модуля), быстрая фаза модуля завершается и происходит переход к дедуктивной (медленной) фазе.

Завершим этот подраздел примером. Полная запись вывода в модификационных структурах весьма громоздка. Поэтому приведем пример сокращенной записи. В этой сокращеной записи будем в качестве основания для добавления строки писать просто Deduction, подразумевая под этим (возможно, длинную) цепочку применения аксиом и правил вывода классического первопорядкового исчисления.

В качестве примера рассмотрим сокращенную запись вывода в модификационном исчислении из подраздела «Модификационные исчисления и модификационные теории». Структура-прототип этого модификационного исчисления представлена в Табл. 4

Табл. 4. Сокращенная запись вывода в модификационном исчислении

| # | Formula | Reason | Stage | Module | Hide by |
|---|---|---|---|---|---|
| 1 | $\mathrm{J}_{\tau}\,\mathbf{PC}(\mathbf{A},\mathbf{P})$ | Axiom | 0 | 1 | 17 |
| 2 | $\mathrm{J}_{\tau}\,\mathbf{PC}(\mathbf{A},\mathbf{Q})$ | Axiom | 0 | 1 | 21 |
| 3 | $\mathrm{J}_{\tau}\,\mathbf{PC}(\mathbf{B},\mathbf{P})$ | Axiom | 0 | 1 | 0 |
| 4 | $\mathrm{J}_{\tau}\,\mathbf{PC}(\mathbf{B},\mathbf{Q})$ | Axiom | 0 | 1 | 27 |
| 5 | $\mathrm{J}_{+1}\,\mathbf{Exp}(1,\mathbf{A},\mathbf{P})$ | Axiom | 0 | 2 | 0 |
| 6 | $\mathrm{J}_{+1}\,\mathbf{Exp}(1,\mathbf{A},\mathbf{Q})$ | Axiom | 0 | 2 | 0 |
| 7 | $\mathrm{J}_{\tau}\,\mathbf{Exp}(1,\mathbf{B},\mathbf{P})$ | Axiom | 0 | 2 | 0 |
| 8 | $\mathrm{J}_{-1}\,\mathbf{Exp}(1,\mathbf{B},\mathbf{Q})$ | Axiom | 0 | 2 | 0 |
| 9 | $\mathrm{J}_{+1}\,\mathbf{Exp}(2,\mathbf{A},\mathbf{P})$ | Axiom | 0 | 2 | 0 |
| 10 | $\mathrm{J}_{-1}\,\mathbf{Exp}(2,\mathbf{A},\mathbf{Q})$ | Axiom | 0 | 2 | 0 |
| 11 | $\mathrm{J}_{\tau}\,\mathbf{Exp}(2,\mathbf{B},\mathbf{P})$ | Axiom | 0 | 2 | 0 |
| 12 | $\mathrm{J}_{-1}\,\mathbf{Exp}(2,\mathbf{B},\mathbf{Q})$ | Axiom | 0 | 2 | 0 |
| 13 | $\forall n\,(n=1\vee n=2)$ | Axiom | 0 | 2 | 0 |
| 14 | $\exists n\,(\mathrm{J}_{+1}\,\mathbf{Exp}(n,\mathbf{A},\mathbf{P}))$ | 5, Deduction | 0 | 2 | 0 |
| 15 | $\neg\exists m\,(\mathrm{J}_{-1}\,\mathbf{Exp}(m,\mathbf{A},\mathbf{P}))$ | 5, 9, 13, Deduction | 0 | 2 | 0 |
| 16 | $\exists n\,(\mathrm{J}_{+1}\,\mathbf{Exp}(n,\mathbf{A},\mathbf{P}))\,\&\,\neg\exists m\,(\mathrm{J}_{-1}\,\mathbf{Exp}(m,\mathbf{A},\mathbf{P}))$ | 14, 15, Deduction | 0 | 2 | 0 |
| 17 | $\mathrm{J}_{+1}\,\mathbf{PC}(\mathbf{A},\mathbf{P})$ | 1, 16, +Rule for PC | 1 | 1 | 0 |
| 18 | $\exists n\,(\mathrm{J}_{+1}\,\mathbf{Exp}(n,\mathbf{A},\mathbf{Q}))$ | 6, Deduction | 1 | 1 | 0 |
| 19 | $\exists m\,(\mathrm{J}_{-1}\,\mathbf{Exp}(m,\mathbf{A},\mathbf{Q}))$ | 10, Deduction | 1 | 1 | 0 |
| 20 | $\exists n\,(\mathrm{J}_{+1}\,\mathbf{Exp}(n,\mathbf{A},\mathbf{Q}))\,\&\,\exists m\,(\mathrm{J}_{-1}\,\mathbf{Exp}(m,\mathbf{A},\mathbf{Q}))$ | 18, 19, Deduction | 1 | 1 | 0 |
| 21 | $\mathrm{J}_{0}\,\mathbf{PC}(\mathbf{A},\mathbf{Q})$ | 2, 20, 0Rule for PC | 1 | 1 | 0 |
| 22 | $\neg\exists n\,(\mathrm{J}_{+1}\,\mathbf{Exp}(n,\mathbf{B},\mathbf{P}))\,\&\,\neg\exists m\,(\mathrm{J}_{-1}\,\mathbf{Exp}(m,\mathbf{B},\mathbf{P}))$ | 7, 11, 13, Deduction | 1 | 1 | 0 |
| 23 | $\mathrm{J}_{\tau}\,\mathbf{PC}(\mathbf{B},\mathbf{P})$ | 3, 22, τRule for PC | 1 | 1 | 0 |
| 24 | $\exists n\,(\mathrm{J}_{-1}\,\mathbf{Exp}(n,\mathbf{B},\mathbf{Q}))$ | 8, Deduction | 1 | 1 | 0 |
| 25 | $\neg\exists m\,(\mathrm{J}_{+1}\,\mathbf{Exp}(m,\mathbf{B},\mathbf{Q}))$ | 8, 12, 13, Deduction | 1 | 1 | 0 |
| 26 | $\exists n\,(\mathrm{J}_{-1}\,\mathbf{Exp}(n,\mathbf{B},\mathbf{Q}))\,\&\,\neg\exists m\,(\mathrm{J}_{+1}\,\mathbf{Exp}(m,\mathbf{B},\mathbf{Q}))$ | 24, 25, Deduction | 1 | 1 | 0 |
| 27 | $\mathrm{J}_{-1}\,\mathbf{PC}(\mathbf{B},\mathbf{Q})$ | 4, 26, –Rule for PC | 1 | 1 | 0 |
| 28 | $\mathrm{J}_{\tau}\,\mathbf{Exp}(1,\mathbf{B},\mathbf{P})$ | 7, 23, τRule for Ext | 1 | 2 | 0 |
| 29 | $\mathrm{J}_{\tau}\,\mathbf{Exp}(1,\mathbf{B},\mathbf{P})$ | 11, 23, τRule for Ext | 1 | 2 | 0 |

Строки с 1 по 16 занимает нулевая стадия вывода — стадия интродукции/ инициализации. Строки 1–4 содержат аксиомы, описывающие предикат **PC**. Строки 5–12 содержат аксиомы, описывающие предикат **Exp**. Строка 13 содержит аксиому из описания состояния (одну из аксиом сюръективности), необходимую нам для доказательства несуществования экспериментов с заданной последовательностью событий.

В строках 14–16 с помощью дедукции выводится большая посылка +Правила для **PC**. В строке 17 это правило применяется и деактивируется строка 1.

В строках 17–27 идет обработка предиката **PC**. Сочетаются дедуктивные правила и правила модификации. Одновременно с применением правила модификации деактивируются строки 1, 2 и 4.

В двух последних строках (28 и 29) обрабатывается предикат **Exp**. Применяются $\tau$-правила.

Если попытаться продолжить этот вывод, то можно убедиться в том, что достигнута стабилизация и области неопределенности предикаторы **PC** и **Exp** уменьшить уже не удается.

Заметим, что более подробно, с разнообразными примерами, как формально, так и неформально, вывод в модификационном исчислении будет рассмотрен в последующих работах.

## Литература


1. Аристотель. Сочинения, т. 2. М.: 1978.
2. Гаек П, Гавранек Т. Автоматическое образование гипотез.— М.: Наука, 1984. 280 с..
3. Милль Д.С. Система логики силлогистической и индуктивной.— М.: Книжное дело, 1900. 781 с.
4. Пирс Ч. Рассуждение и логика вещей. М.: Изд-во РГГУ, 2005. 371 с.
5. Финн В.К. Синтез познавательных процедур и проблема индукции //НТИ. Сер. 2. 1999. № 1–2, С. 8–45.