



использованы для аргументации. Объективная информация служит фактами и исходными посылками, субъективная информация является источником пересматриваемых выводов, а гипотетическая информация – помогает строить предположения.

Существуют несколько формализаций теории аргументации. Например, системы абстрактной аргументации, предложенные Дангом (Dung P.M.) [Bondarenko et al., 1997], аргументационная система Лина и Шоэма (Lin F., Shoham Y.) [Lin et al., 1989], система Вресвийка (G.A.W. Vreeswijk [Vreeswijk, 1989], система аргументации Поллока (John L. Pollock) [Pollock, 1992] и некоторые другие.

Все эти подходы можно условно разделить на три типа [Besnard, 2008].

1) Абстрактные системы, предложенные Дангом и позднее развиваемые Праккенем и Сартором. В этих системах аргументы представляются как элементы множества, в котором задано бинарное отношение «атака». В этих системах авторы полностью абстрагируются от внутренней структуры аргументов и природы множества аргументов. В таких системах отсутствует механизм получения новых аргументов, и задача сводится к поиску неконфликтующих аргументов на заданном множестве аргументов.

2) Согласованные системы (coherence systems). В таких системах основной стратегией обработки противоречий в базе знаний является выделение согласованных подмножеств из всего объема информации, имеющейся в базе знаний. Обычно такие системы базируются на классической логике, хотя возможны применения и модальных, временных или дескриптивных логик.

3) Системы пересматриваемых рассуждений. В таких системах обычно происходит введение пересматриваемого следствия в качестве элемента языка. То есть кроме импликации в обычном смысле вводится ее пересматриваемый аналог. Аргументы в таких системах представляются как последовательность рассуждений, ведущих к заключению, каждый шаг которых может подвергнуться поражению.

В данной работе будет подробно рассмотрена система аргументации, основанная на пересматриваемых рассуждениях, предложенных Джоном Поллоком [Pollock, 1992].

## 1.2. Пересматриваемые рассуждения

Прежде чем переходить к изложению основного материала, дадим необходимые определения и введем обозначения [Вагин и др., 2008].

*Определение 1.* Аргумент - пара, состоящая из множества посылок и заключения. Записывать такие пары будем в следующем виде  $p/X$ , где  $p$  заключение, а  $X$  множество посылок.

Например, аргумент  $(p \rightarrow q)/\{\sim A, B\}$  означает, что из посылок  $\sim A, B$  следует  $p \rightarrow q$ . На всех

иллюстрациях будем обозначать аргументы овалами. Для аргументов с пустым множеством посылок (такие аргументы называют фактами), будем писать только заключение. Например, фактом является утверждение, что Земля вращается вокруг Солнца.

*Определение 2.* Интерес – аргумент, который мы хотим обосновать в ходе монотонного и/или пересматриваемого вывода. На графе вывода будем обозначать интересы прямоугольниками.

*Определение 3.* Граф вывода – граф, показывающий взаимосвязи между аргументами и интересами. Он отображает, из каких аргументов порождается новый аргумент. Аналогично в нем показывается, из каких интересов получаются новые интересы. А также граф вывода отображает конфликты между аргументами при пересматриваемых рассуждениях.

*Определение 4.* Дедуктивное следствие – простые, дедуктивные правила вывода, означающие, что если истинно  $P$ , то истинно и  $Q$ . Такие правила не являются пересматриваемыми. Записывать такие правила будем так:  $P \Rightarrow Q$ . На графе вывода будем отображать их обыкновенными стрелками (см. аргументы  $P$  и  $Q$  на рис 1).

*Определение 5.* Пересматриваемое следствие. Это пересматриваемые правила вывода, которые могут быть получены, например, в результате индукции или абдукции. В данной работе нас не интересует конкретный механизм получения таких выводов, поэтому такие правила подаются декларативным образом на вход программы. Аргументы, полученные в результате таких выводов, будем называть пересматриваемыми. Записывать такие правила будем так:  $M \Rightarrow N$ . На графе вывода такие связи будем отображать пунктирными стрелками, а пересматриваемые аргументы – двойным овалом (см. аргументы  $M$  и  $N$  на рис. 1).

Понятие конфликта – основа системы аргументации. Будем рассматривать два типа конфликтов – опровержение и подрыв [Вагин и др., 2008].

*Определение 6.* Опровержение (rebutting) – ситуация, когда некоторые аргументы опровергают заключения других аргументов. Иными словами, аргумент  $A_1 = p_1/X_1$  опровергает аргумент  $A_2 = p_2/X_2$ , когда заключение  $p_1$  опровергает заключение  $p_2$ . Опровержение является симметричной формой атаки.

*Определение 7.* Подрыв (undercutting) – несимметричная форма атаки, когда один аргумент отрицает связь между посылками и заключением другого аргумента.

*Определение 8.* Подрывающие доводы. Это аргументы, поражающие связь между двумя другими аргументами, соединенными пересматриваемым следствием. Например, имеется аргумент  $E$ , подрывающий пересматриваемую связь  $C \Rightarrow D$  между аргументами  $C$  и  $D$ . Такие правила подрыва будем записывать в виде  $E \Rightarrow (C @ D)$ . На графе вывода подрывающие аргументы и пораженные ими аргументы будем соединять

жирной пунктирной стрелкой. Пораженные аргументы будем пометать темно-серым цветом (см. аргумент D и на рис. 1).

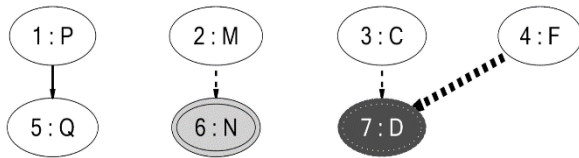


Рисунок 1 - Граф вывода

### 1.3. Обнаружение конфликтов

Обнаружение конфликтов является наиболее сложной частью системы аргументации. Основные сложности появляются из-за необходимости поддержки логики предикатов первого порядка. Если говорить про логику высказываний, то конфликт типа «опровержение» это ситуация, когда одновременно существует два аргумента –  $A$  и  $\sim A$ . Однако, для логики предикатов первого порядка все несколько сложнее. Основная идея — применение механизма унификации для обнаружения конфликтов. Подстановка  $U = \{t_1/x_1, t_2/x_2, \dots, t_n/x_n\}$ , где  $t_i$  – термы и  $x_i$  – переменные называется унификатором для выражений  $W_1$  and  $W_2$ , если  $W_1 * U = W_2 * U$ , где  $W_1 * U$  результат замены переменных  $x_1, x_2, \dots, x_n$  на термы  $t_1, t_2, \dots, t_n$  и предполагается, что  $W_1$  и  $W_2$  не содержат переменные с одинаковыми именами (если это так их необходимо переименовать) [Вагин и др., 2008]. Например выражения  $P(x_1) \& G(x_2) \rightarrow H(x_1, x_2)$  и  $P(f(a)) \& G(b) \rightarrow H(f(a), b)$  унифицируются унификатором  $U = \{f(a)/x_1, b/x_2\}$ .

Будем говорить, что два аргумента  $A_1 = p_1 / X_1$  и  $A_2 = p_2 / X_2$  вступают в конфликт типа «опровержение», если существует унификатор  $U$ , такой что:

1.  $\sim p_1 * U = p_2 * U$ .
2.  $X_1 * U \in X_2 * U$  или  $X_2 * U \in X_1 * U$ .

Аргумент  $A_1$  подрывает пересматриваемую связь между аргументами  $A_2$  и  $A_3$ :

1. Существует пересматриваемая связь между  $A_2$  и  $A_3$ .
2. Существует подрывающее правило  $E = (C @ D)$  и существуют такие унификаторы  $U_1$  и  $U_2$ , такие что:
  - а.  $E * U_1 = A_1 * U_1$
  - б.  $(C * U_1) * U_2 = (A_2) * U_2$
  - с.  $(D * U_1) * U_2 = (A_3) * U_2$

Определив все возникающие конфликты, можно определять статусы аргументов. На каждом шаге работы системы определение статусов каждого аргумента (поражается он или нет) играет ключевую роль. Введем необходимые для определения статусов поражения определения.

### 2. Степени обоснования в пересматриваемых рассуждениях.

Прежде чем переходить к способам и алгоритмам вычисления степеней обоснования,

рассмотрим как они могут задаваться и откуда получены. В данной статье для задания степеней обоснования используется числовая шкала  $[0,1]$ , где 0 соответствует пораженному аргументу, 1 наиболее обоснованному аргументу. Степени обоснования могут быть двух типов:

- 1) Степени обоснования исходных аргументов;
- 2) Степени обоснования пересматриваемых правил.

Первый тип степеней обоснования присваивается каждому исходному аргументу, и представляет собой некую оценку достоверности источника, из которого получен данный аргумент. Например, по телевизору сказали, что вероятность осадков 70%. Соответственно мы можем построить аргумент  $A1: \text{Завтра(дождь)}$  со степенью обоснования 0.7. Степени обоснования будем записывать функцией  $Jus(A)$ . То есть для приведенного примера  $Jus(\text{Завтра(дождь)}) = 0.7$ . Конкретные механизмы получения степеней обоснования зависят прежде всего от предметной области. Например, это могут быть статистические данные (в 90% этот источник дает верные данные) или экспертные оценки (вероятность роста акций 60%).

Второй тип степеней обоснования связан с пересматриваемыми правилами. Как уже говорилось выше, часто пересматриваемые правила появляются в результате формализации знаний эксперта вида «Если  $A$ , то чаще всего  $B$ ». Такие правила так же могут нести в себе некоторую числовую оценку.

Например, применение анальгина в 85% приводит к снижению температуры тела пациента (формально  $R1$ :

$$(\forall x) \text{ прием(анальгин, x) } \Rightarrow \text{ понижение\_температуры(x) }.$$

Одновременное использование обоих типов степеней обоснования является довольно сложной задачей и требует дополнительных исследований. В данной статье ограничимся рассмотрением степеней обоснования первого типа – для изначально заданных аргументов.

Итак, нам необходимо задать функцию  $Jus(A)$  для вычисления любого из аргументов в графе вывода. Будем считать, что для изначальных аргументов эта величина является определенной. На значение этой функции будут оказывать влияние два фактора – дерево вывода аргумента (т.е. степень обоснования аргументов, которые использовались в выводе данного аргумента) и конфликты с другими

аргументами. Для удобства рассмотрим эти два фактора отдельно:  $Jus_{anc}(A)$  – унаследованная степень обоснования и  $Jus_{con}(A)$  – на сколько конфликт подрывает обоснование аргумента. Пусть  $Anc = \{Anc_i\}, i \in 1..n$  – множество аргументов  $Anc_i$ , учествовавших в выводе аргумента  $A$ ,  $n$  – количество таких аргументов.

$$Jus_{anc}(A) = \min_{i \in 1..n} \{Jus(Anc_i)\} \quad (2.1)$$

Формулу (2.1) называют принципом слабой связи [Pollock, 2001]. Надо отметить, что это не единственный подход к вычислению степени обоснования, в ряде работ применяется байесовский подход (см. например [Haenni et al., 1999]).

Отметим, что из формулы (2.1) следует, что, если производить вычисление степеней обоснования рекурсивно, начиная от исходно заданных аргументов, то можно искать минимум, не из всех аргументов в базе, а только на предыдущем шаге. Таким образом, если у аргумента один предок, то его унаследованная степень обоснования будет равна степени обоснования его предка.

Если при вычислении  $Jus_{anc}$  ищутся наиболее слабые аргументы, то при определении того, насколько конфликт уменьшает обоснования, используется наиболее сильные аргументы. Пусть  $A_{confl}$  – множество аргументов, вступающих в конфликт с  $A$ ,  $n = |A_{confl}|$ , тогда

$$Jus_{con}(A) = \begin{cases} \max_{i \in 1..n} \{Jus_{anc}(A_{confl_i})\}, & n > 0 \\ 0, & \text{в противном случае} \end{cases} \quad (2.2)$$

В формуле (2.2) используется  $Jus_{anc}$  для того, чтобы верно обрабатывать случаи, когда между аргументами есть конфликт типа опровержение. Итак, окончательно:

$$Jus(A) = \begin{cases} Jus_{anc}(A) - Jus_{con}(A), & Jus_{anc}(A) > Jus_{con}(A) \\ 0, & \text{в противном случае} \end{cases}$$

### 3. Алгоритмы

#### 3.1. Алгоритм поиска конфликтов

Входные данные: Граф вывода, список подрывающих следствий **undercut\_rules**

Выходные данные: Граф вывода с конфликтами

**Шаг 1.** Для каждого аргумента  $Arg_1 = a_1/X_1$  проверить не существует ли  $Arg_2 = a_2/X_2$ , такого, что для  $a_1$  и  $\sim a_2$  существует унификатор  $U$  и

$X_1 * U \in X_2 * U$  или  $X_2 * U \in X_1 * U$ . Если такой унификатор существует, добавить конфликт между  $Arg_1$  и  $Arg_2$ .

**Шаг 2.** Для каждого подрывающего правила  $uc\_rule$  вида  $p \Rightarrow (q @ r)$  из **undercut\_rules** выполнить:

**Шаг 3.** Взять левую часть правила -  $p$ . Для каждого аргумента  $Arg = a/X$ , проверить существует ли унификатор  $U_{left}$  для выражений  $a$  и  $p$ .

**Шаг 4.** Для тех аргументов  $Arg$ , которые унифицируемы с левой частью подрывающего правила выполнить следующие шаги:

**Шаг 4.1.** Применить унификатор  $U_{left}$  к  $q$  и  $r$  ( $q_u = q * U_{left}$  и  $r_u = r * U_{left}$ ).

**Шаг 4.2.** Составить массив

$X = [\{xq_1, xr_1\}, \{xq_2, xr_2\}, \dots, \{xq_n, xr_n\}]$  состоящий из пар аргументов, соединенных пересматриваемой. Для каждого элемента этого массива выполнить:

**Шаг 4.2.1** Проверить существует ли унификатор  $U_{right}$  для  $q_u$  и  $xq_i$ . Если найденный унификатор так же является унификатором для  $r_u$  и  $xr_i$ , добавить новый конфликт в граф вывода.

#### 3.2. Алгоритм вычисления степеней обоснования.

Процедура **recalculate\_justification\_degree()**;

Входные данные: Граф вывода.

Выходные данные: Граф вывода

**Шаг 1.** Составить очередь  $Qlist$  вычисления степеней обоснования из аргументов входящих в состав графа вывода, начиная от изначально заданных аргументов.

**Шаг 2.** Пока очередь  $Qlist$  не пуста извлечь из нее очередной элемент  $q$  и выполнить процедуру  $Jus(q)$ .

Процедура **Jus(argument q, [argument ex])**;

Входные данные: Аргумент  $q$ , необязательный параметр – аргумент  $ex$ , который не учитывается при определении степени обоснования аргумента  $q$ , граф вывода.

Выходные данные: Граф вывода

**Шаг 1.** Если  $q$  – изначально заданный аргумент, то положить  $Jus_{anc}(q)$  равным изначально заданной степени доверия для этого аргумента. В противном случае из всех  $Anc_i$  являющихся непосредственным предком  $q$  (то есть  $q$  получается из  $Anc_i$  за один шаг) найти такой  $Anc\_min$ , что его степень обоснования минимальна. Положить  $Jus_{anc}(q) = Jus(Anc\_min)$ .

**Шаг 2.** Если аргумент  $q$  не имеет конфликтов, тогда положить  $Jus_{con}(q) = 0$ . В противном случае составить множество  $\{A_i\}$  состоящее из аргументов имеющих конфликт с  $q$ . Если параметр  $ex$  задан, исключить его из множества  $\{A_i\}$ . Если к моменту вычисления  $Jus(q)$  среди  $A_i$  нет аргументов с неопределенной степенью обоснования, то среди них найти тот аргумент  $A\_max$ , у которого  $Jus(A\_max)$  максимален. Положить  $Jus_{con}(q) = \max(Jus(A\_max), Jus_{anc}(ex))$ , перейти к шагу 4. В противном случае (т.е. среди  $\{A_i\}$  есть аргументы с неопределенными степенями

обоснования) выполнить шаг 3.

**Шаг 3.** Для каждого  $A_i$ , для которого не определена степень доверия выполнить:

**Шаг 3.1.** Если  $A_i$  имеет конфликт с  $q$ , но аргумент  $q$  не имеет с ними конфликта (несимметричная форма конфликта) выполнить  $Jus(A_i)$ . В противном случае выполнить  $Jus(A_i, q)$ , т.е. рассчитать  $Jus(A_i)$  без учета конфликта с  $q$ .

**Шаг 4.** Положить  $Jus(q) = Jus_{anc}(q) - Jus_{con}(q)$ , если  $Jus_{anc}(q) > Jus_{con}(q)$ . В противном случае  $Jus(q) = 0$ .

#### 4. Пример работы системы аргументации

**Имеется следующая задача:** Предприниматель хочет вложить некоторую сумму в развивающийся бизнес и получить от этого доход. Есть компания CMP, которая занимается IT бизнесом. По статистике IT рынок находится на подъеме и 75% IT компаний показывают положительную динамику развития. Компания CMP занимается разработкой программного обеспечения под операционные системы семейства Linux. В России менее 10% пользователей Linux и следовательно спрос на продукцию будет не велик. Однако 80 процентов производства компании направлено на международный рынок программного обеспечения для веб серверов, которые в основном работают на Linux.

Запишем условия формально на языке ЛППП.

- A1. IT(CMP) justification: 0,75
- A2. под\_Linux(CMP) justification: 1
- A3. рынок\_сбыта(для\_серверов) justification: 0,8
- A4.  $(\forall x)(инвестировать(x) \rightarrow \text{доход})$

- R1.  $(\forall x)(IT(x)) \Rightarrow (V x) \text{рост\_акций}(x)$
- R2.  $(\forall x)(\text{рост\_акций}(x) \Rightarrow \text{инвестировать}(x))$
- R3.  $(\forall x)(\text{под\_Linux}(x) \Rightarrow \text{плохой\_спрос}(x))$
- R4.  $(\forall x)(\text{плохой\_спрос}(x) \Rightarrow \sim \text{инвестировать}(x))$
- R5.  $(\forall x) \text{рынок\_сбыта(для\_серверов)} \Rightarrow ((\text{под\_Linux}(x)) @ \text{плохой\_спрос}(x))$

На рисунке 2 приведен граф вывода для данной задачи.

Рассмотрим решение данной задачи поэтапно.

- 1) Из аргумента A1: IT(CMP) и правила вывода R1 получается пересматриваемый аргумент A6:  $Jus(A6) = Jus(A1) = 0.75$ .
- 2) Из аргумента A2: под\_Linux(CMP) и правила вывода R3 получается пересматриваемый аргумент A8:  $Jus(A8) = Jus(A2) = 1$ .
- 3) Аргумент A8:  $\text{плохой\_спрос}(CMP)$  с помощью правила R5 вступает в конфликт типа подрыв с аргументом A3:  $\text{рынок\_сбыта(для\_серверов)}$ . Согласно формуле 1.3  $Jus(A3) = Jus_{anc}(A3) - Jus(A8) = 0.2$
- 4) Из аргумента A6 и правила R2 получается аргумент A7:  $Jus(A7) = Jus(A6) = 0.75$ .
- 5) Из аргумента A8 и правила вывода R4 получается аргумент A9:  $Jus(A9) = 0$ .
- 6) Аргументы A9 и A7 вступают в конфликт типа опровержение, степень обоснования аргумента A7 выше, и следовательно  $Jus(A7) = 0.55$ , а аргумент A9 становится полностью пораженным с  $Jus(A9) = 0$ .
- 7) Используя правила обратного вывода из интереса In5 получается интерес In11, из которого получается интерес In14.

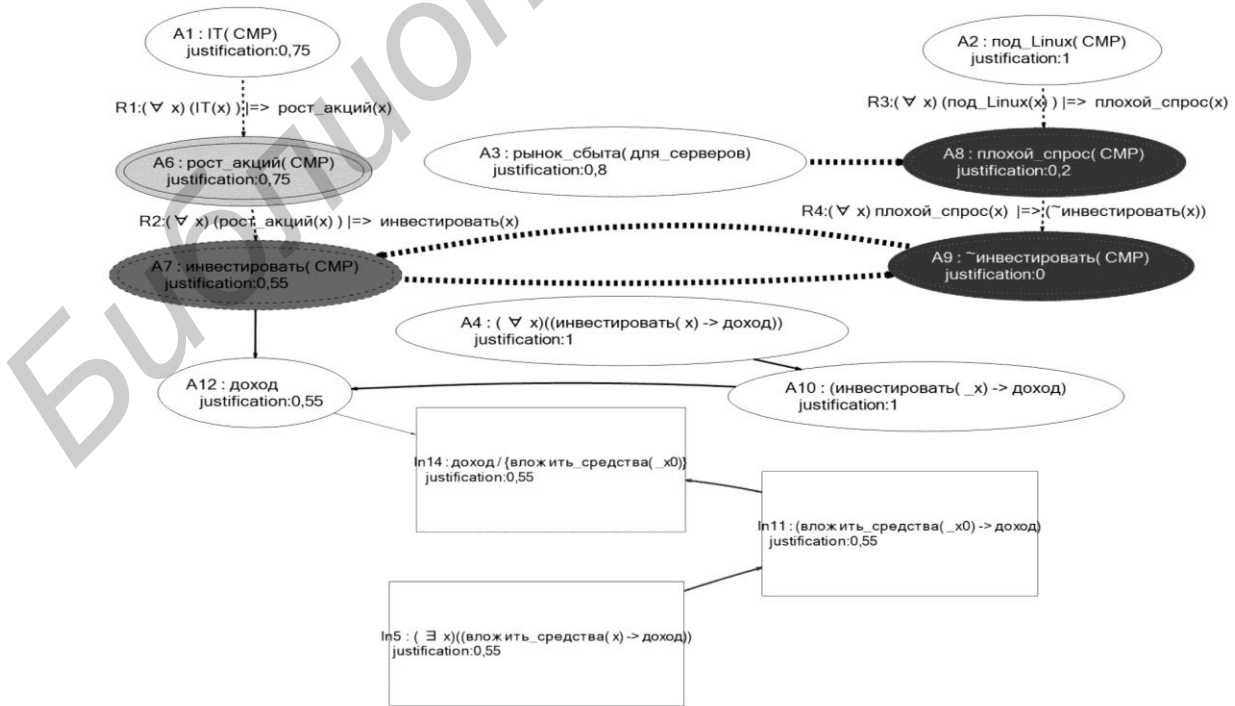


Рисунок 2 - Пример решения задачи со степенями обоснования

8) Из аргумента A10: Инвестировать( $x$ ) $\rightarrow$ доход, аргумента A7: Инвестировать(CMP) применяя унификацию и правило Modus Ponens получаем аргумент A12. Согласно принципу слабой связи  $Jus(A12) = 0.55$ . Этот аргумент подтверждает интерес In14.

9) Окончательно получаем, что интересы подтверждены со степенью обоснования 55%. То есть инвестиция в компанию CMP принесет доход с вероятностью 55%.

## Заключение

Конфликты в базах знаний являются серьезной проблемой при построении любых интеллектуальных систем. Поиск, анализ и, возможно, разрешение этих конфликтов – важная и интересная задача. В данной статье был рассмотрен один из подходов к данной проблеме – использование аргументации, а именно теории пересматриваемых рассуждений. Были рассмотрены теоретические основы пересматриваемых рассуждений и дан обзор системы, основанной на них. Кроме того, применение механизма степеней обоснования позволило решить важную задачу – давать не только качественные оценки (“поражен” или “не поражен”), но и количественные. Разработанная программная система справилась со многими задачами, не решаемыми с точки зрения классической логики. Она была протестирована и отлажена на многих тестовых задачах из статьи Врейсвика “Interpolation of Benchmark Problems in Defeasible Reasoning” [Vreeswijk, 1995].

## Библиографический список

- [Besnard, 2008] Besnard P., Hunter A. Elements of Argumentation. – Cambridge MA: MIT Press, 2008. – 298 p.
- [Bondarenko et al., 1997] Bondarenko A., Dung P.M., Kowalski R.A., Toni F. An Abstract Argumentation-Theoretic Framework for Defeasible Reasoning // Artificial Intelligence. – 1997. – Vol. 93, № 1-2. – P. 63-101..
- [Lin et al., 1989] Lin F., Shoham Y. Argument Systems. A Uniform Basis for Nonmonotonic Reasoning // Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning. – San Mateo CA: Morgan Kaufmann Publishers Inc, 1989. – P. 245-355.
- [Vreeswijk, 1989] Vreeswijk G.A.W. Abstract Argumentation Systems // Artificial Intelligence. – 1997. – Vol. 90. – P. 225-279.
- [Pollock, 1992] Pollock J.L. How to Reason Defeasibly // Artificial Intelligence. – 1992. – Vol. 57. – P. 1-42.
- [Вагин и др., 2008] Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В. Достоверный и правдоподобный вывод в интеллектуальных системах / Под ред. В.Н. Вагина, Д.А. Поспелова. 2-е издание дополненное и исправленное. – М.: Физматлит, 2008. – 712 с.
- [Pollock, 2001] Pollock J.L. Defeasible reasoning with variable degrees of justification, // Artificial Intelligence. Artificial Intelligence. – 2001. – Vol. 133. – P. 233-282.
- [Haenni et al., 1999] Haenni R., Kohlas J., Lehmann N. Probabilistic Argumentation Systems, Handbook of Defeasible Reasoning and Uncertainty Management Systems // Algorithms for Uncertainty and Defeasible Reasoning. – 1999. – Vol. 5. – P. 221–287.
- [Vreeswijk, 1995] Vreeswijk G.A.W. Interpolation of Benchmark Problems in Defeasible Reasoning // Proceedings of the 2nd World Conference on the Fundamentals of Artificial Intelligence (WOCFAI 95). – Paris, France, 1995. – P. 453-468.

## METHODS AND ALGORITHMS FOR CALCULATING JUSTIFICATION DEGREES IN ARGUMENTATION SYSTEMS

Morosin O.L. \*

\*Moscow Power Engineering Institute

omorsik@gmail.com

This paper provides a brief overview of approaches to the formalization of argumentation systems. Opportunities of application of justification degrees are also observed. Justification degrees allow us to solve various argumentation problems that need numerical estimation of an answer. In contrast to classical logic, defeasible reasoning allows us to draw conclusions on the contradictory and incomplete sets of propositions.

## Introduction

Most practical problems are rather poor formalized, and may contain inconsistent, uncertain and implausible information. The methods of the classical logics could not be applied for solving such problems. It is proposed to use the methods of argumentation to cope with uncertain and inconsistent data.

## Main Part

There are several formalizations of the argumentation theory. In this paper, we will consider the development of defeasible argumentation systems that use the first order logic as an underlying language and based on the theory of defeasible reasoning proposed by John Pollock. All conclusions are not considered as reliable and may be revised at a later stage of reasoning when new knowledge (or even new conclusions from existing knowledge) appears.

The concept of the conflict is the main point in theory of defeasible reasoning. Moreover, the possibility of conflicts is the reason why defeasible reasoning is considered. The paper contains two types of conflict: rebutting and undercut. Defeasible reasoning allows us to find and analyze such conflicts.

Opportunities of application of justification degrees are also observed. Justification degrees allow us to solve various argumentation problems that need numerical estimation of an answer.

## Conclusion

Argumentation system based on defeasible reasoning could be used in many intelligent systems that use the first order logic as a method of knowledge representation. It was shown that it is a powerful tool for dealing with inconsistent data. Nevertheless, the implementation of different mechanisms of calculating justification degrees, including methods of fuzzy logic, is a topic of a future work.