

Методы поиска решений

Кружалов Алексей Сергеевич

Московский Политехнический Университет
19 сентября 2022 г.



Поиск в пространстве состояний

Множество проблем, решаемых с помощью ИИ, можно сформулировать в терминах трех важнейших ингредиентов:

- **исходное состояние** проблемы (например, исходное состояние головоломки);
- **тест завершения** — проверка, достигнуто ли требуемое конечное состояние или найдено решение проблемы (примером может послужить правило определения, собрана ли головоломка);
- **множество операций**, которые можно использовать для изменения текущего состояния проблемы (например, шаги или перемещения фигур при сборке головоломки).

Игра в 15 («пятнашки»)

2	4	5	7
8	3	1	11
14	6		10
9	13	15	12

12	9	4	2
8	7	3	14
	1	6	11
5	13	10	15

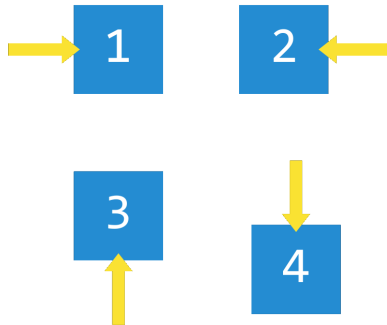
15	4	10	3
13	1	11	12
9	5	14	7
6	8		2

Игра в 15 («пятнашки»)

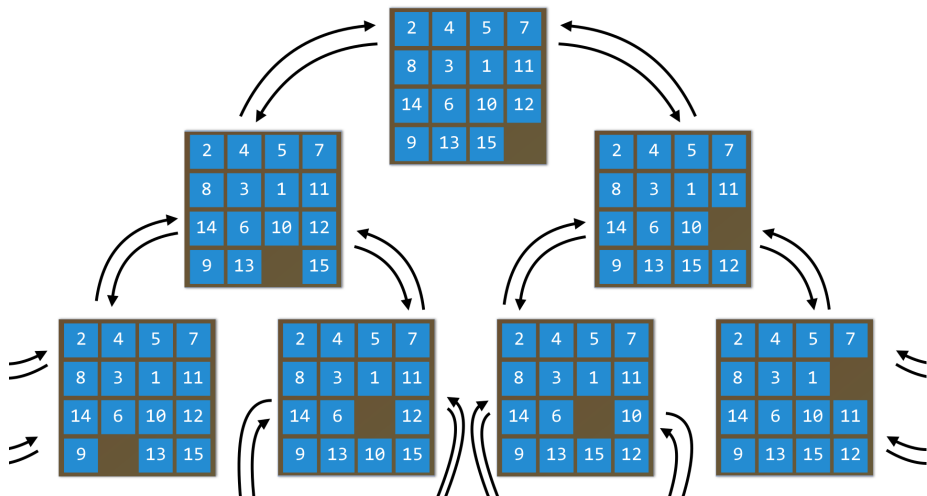
Начальное состояние

2	4	5	7
8	3	1	11
14	6		10
9	13	15	12

Множество операций



Игра в 15 («пятнашки»)

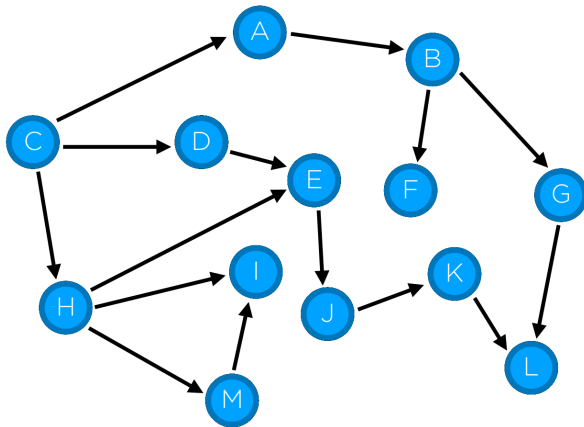


Поиск в пространстве состояний

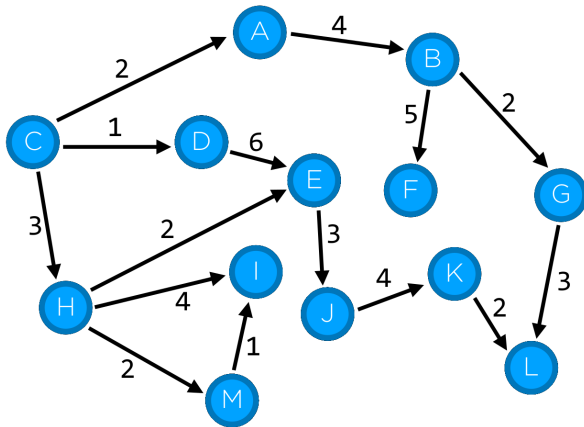
Один из способов представления пространства состояний — граф, в котором состояниям соответствуют узлы, а операциям — дуги.

Чаще всего в задачах поиска в пространстве состояний применяются **неявные графы**. В отличие от **явных графов**, структура неявных графов изначально неизвестна (т. е. не задана в явном виде, например, при помощи матрицы смежности), а вычисляется динамически на основе начальной вершины и правил перехода между состояниями.

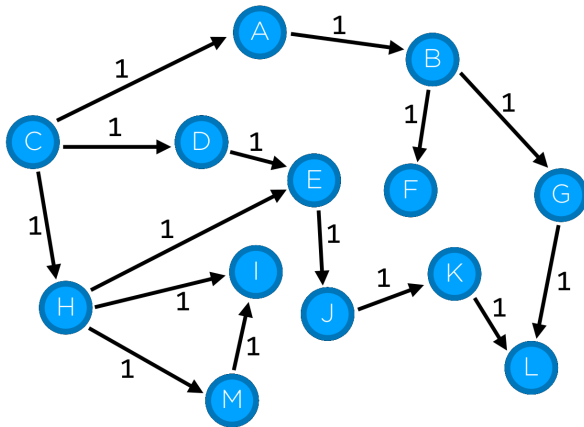
Поиск в пространстве состояний



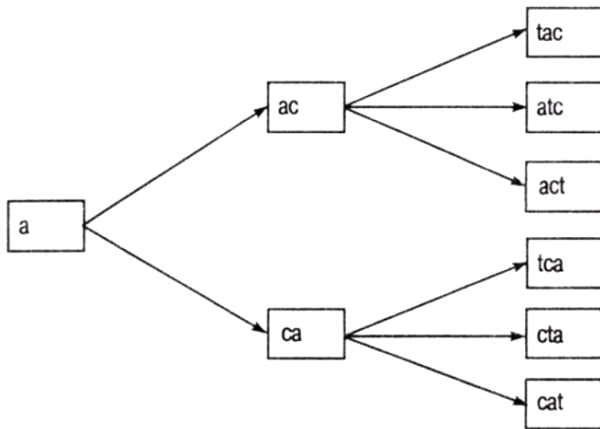
Поиск в пространстве состояний



Поиск в пространстве состояний



Поиск в пространстве состояний



Поиск в пространстве состояний

Решением задачи поиска в пространстве состояний будет являться последовательность операций, которая позволяет перейти от начального состояния к конечному (целевому).

Стоимостью решения называется длина пути от начальной вершины до конечной, вычисленная в соответствии с весами дуг графа состояний.

Оптимальным решением является решение, обладающее наименьшей стоимостью среди всех возможных решений.

Поиск в пространстве состояний

Для оценки алгоритмов поиска используются такие понятия как полнота и оптимальность.

Полнота — свойство алгоритма всегда находить решение, если оно существует.

Оптимальность — свойство алгоритма всегда находить решение с наименьшей стоимостью.

Поиск в пространстве состояний

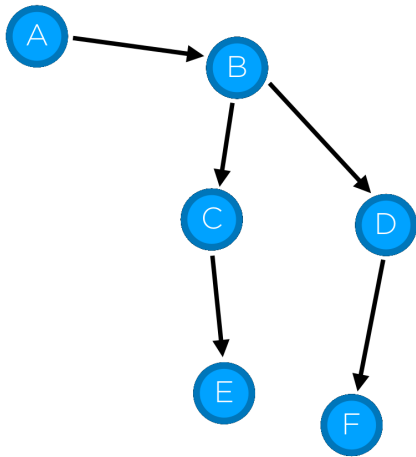
Алгоритм поиска путём последовательного формирования возможных состояний и их проверки на предмет удовлетворения критерию достижения целевого состояния получил название **алгоритм порождения и проверки** (англ. generate and test).

Так как данный алгоритм не использует никакой дополнительной информации о решаемой задаче, он относится к категории **неинформированных методов поиска**.

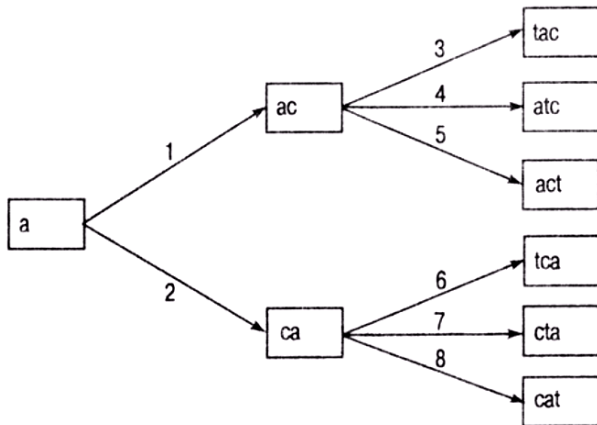
Алгоритм имеет два основных варианта: **поиск в глубину** (англ. depth-first search) и **поиск в ширину** (англ. breadth-first search). Отличаются варианты порядком формирования состояний.

Стратегии обхода вершин

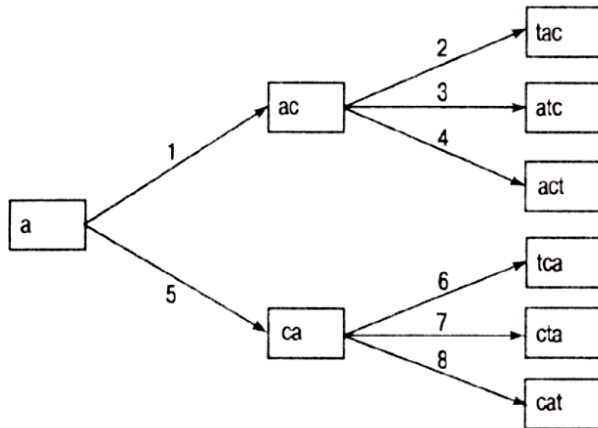
Цель — найти путь от A до E.



Стратегии обхода вершин



Стратегии обхода вершин



Стратегии обхода вершин

Алгоритм **поиска в ширину** отыскивает решение, путь к которому на графе — кратчайший, если таковое существует. Другими словами, он находит кратчайший путь между исходным состоянием и решением. Алгоритмы, обладающие таким свойством, называются разрешимыми.

Алгоритм **поиска в глубину** может быстрее найти решение, особенно, если при его выполнении используются эвристики для выбора очередной ветви. Но этот алгоритм может никогда не закончиться, если пространство состояний бесконечно.

Эвристический поиск

Информированный поиск (эвристический поиск) — стратегия поиска решений в пространстве состояний, в которой используются знания, относящиеся к конкретной задаче.

Самая простая форма эвристического поиска — это **восхождение на гору**. В процессе поиска в программе используется некоторая **оценочная (эвристическая) функция** $h(n)$, с помощью которой можно грубо оценить, насколько «хорошим» (или «плохим») является текущее состояние. Затем можно применить ту же функцию для выбора очередного шага, переводящего систему в следующее состояние.

Восхождение на гору

Основной алгоритм, реализующий идею восхождения на гору, можно сформулировать следующим образом.

- (1) Находясь в данной точке пространства состояний, применить правила порождения нового множества возможных решений, например множества ходов фигур, допустимых в данной позиции.
- (2) Если одно из новых состояний является решением проблемы, прекратить процесс. В противном случае перейти в то состояние, которое характеризуется наивысшим значением оценочной функции. Вернуться к шагу (1).

Поиск по первому наилучшему совпадению

Другая форма эвристического поиска – **поиск по первому наилучшему совпадению** (англ. best-first search). Так же, как и во варианте восхождения на гору, в нашем распоряжении имеется оценочная функция, с помощью которой можно сравнивать состояния в пространстве состояний. Основное же отличие нового метода от ранее рассмотренного состоит в том, что сравниваются не только те состояния, в которые возможен переход из текущего, но и все, до которых «можно достать».

Поиск по первому наилучшему совпадению

OPEN = [Начальное состояние]

пока OPEN не пусто повторять:

1. Удалить лучший узел из OPEN, назовем его N.
2. Если N целевое состояние, делаем трассировку пути назад к начальному узлу (через записи к родителям от N) и возвращаем путь.
3. Создать список потомков узла N.
4. Оцениваем каждого потомка, добавляем его в OPEN, и записываем N как его родителя.

закончить

Алгоритм A*

В алгоритме A* в качестве оценочной функции используется функция

$$f(n) = g(n) + h(n),$$

где

$g(n)$ – функция длины пути от начальной вершины до вершины n ,

$h(n)$ – функция эвристической оценки расстояния от вершины n до конечной.

Алгоритм A*

Алгоритм A* является оптимальным при условии, что

- $h(n)$ – **допустимая эвристическая оценка** (никогда не переоценивает расстояние до целевой вершины) и
- $h(n)$ – **монотонная (преемственная)** эвристика, т. е. для любого состояния n и его потомка n' должно выполняться неравенство $g(n) + h(n) \leq g(n') + h(n')$.

Эвристический поиск: выводы

- проблему любой сложности, в принципе, можно свести к проблеме поиска в пространстве состояний, если только удастся ее формализовать в терминах начального состояния, конечного состояния и операций перехода в пространстве состояний;
- поиск в пространстве состояний должен направляться определенным образом представленными знаниями о конкретной предметной области.

Эвристический поиск: выводы

Большинство исследователей весьма скептически относятся к возможности использования в прикладных системах таких методик поиска решений проблем, как «порождение и проверка» и «восхождение на гору». Серьезные технические сложности программной реализации оценочных функций навели на мысль, что такая методика недооценивает возможности узкоспециальных знаний в конкретной предметной области и переоценивает возможности обобщенного подхода к воспроизведению механизмов человеческого мышления. Весьма мало вероятно, что сегодня существовала бы такая область исследований, как экспертные системы, если бы удалось найти общие принципы решения проблем, которые можно было бы применять, отвлекаясь от специфики конкретной предметной области.

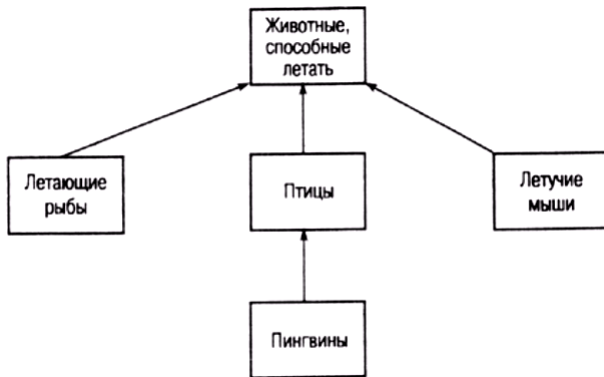
Схемы представления знаний

Цель исследований по представлению знаний – выяснить, каким образом и многообразие сведений об отдельных фактах, и общие принципы построения окружающего нас мира можно использовать в компьютерной программе, которая ориентирована на построение логического рассуждения, направленного на достижение определенной цели. Эти исследования включали использование конструкций следующих видов:

- правил в форме, «если имеет место это условие, то примени этот оператор»;
- разного рода сетей, в которых узлы соответствуют концепциям, а дуги — отношениям между ними;
- логических формул, представляющих отдельные факты и принципы.

Схемы представления знаний

Семантические цепи представляют собой средство представления знаний, базирующееся на формализме теории графов.



Представление знаний

В области экспертных систем **представление знаний** означает систематизированную методику описания на машинном уровне того, что знает человек-эксперт, специализирующийся в конкретной предметной области.

Этапы обработки знаний

1. Знания в памяти человека как результат мышления.
2. **Материальные носители знаний (учебники, методические пособия).**
3. Поле знаний – условное описание основных объектов предметной области, их атрибутов и закономерностей, их связывающих.
4. Знания, описанные на языках представления знаний.
5. База знаний на материальных носителях информации.

Представление знаний

Одна из целей создания специальных моделей представления знаний – избавиться от той **лексической или структурной неоднозначности**, которая присуща естественному человеческому языку.

Пример

Сообщение «Посещение тетушки может быть надоедливym» можно трактовать и как утверждение, что «надоедает наносить визиты тетушке», и как утверждение, что «надоедает, когда тетушка наносит визит».

Представление знаний

Один из парадоксов искусственного интеллекта состоит в том, что многие задачи **поиска смыслового содержания**, которые легко решаются человеком, очень трудно реализовать на машине и наоборот.

Пример

Рассмотрим следующую фразу: «Молоток ударил графин, и он разбился». К чему относится «он» в этой фразе? Для нас ответ очевиден, и мы даже не замечаем неоднозначности в этой фразе. Но как в общем смысле машина будет интерпретировать эту фразу? Предположение, что «он» относится к последнему по порядку следования в предложении существительному, не всегда срабатывает. Например: «Графин ударился о камень, и он разбился.»

Представление знаний

Представление предполагает определенную **организованность знаний**. Представление знаний должно позволить извлекать их в нужной ситуации с помощью относительно несложного и более-менее естественного механизма. Простого перевода информации (знаний) в форму, пригодную для хранения на машинных носителях, здесь явно недостаточно. Для того чтобы можно было достаточно быстро извлекать те элементы знаний, которые наиболее пригодны в конкретной ситуации, база знаний должна обладать достаточно развитыми средствами индексирования и контекстной адресации. Тогда программа, использующая знания, сможет управлять последовательностью применения определенных «элементов» знания, даже не обладая точной информацией о том, как они хранятся.

Этапы обработки знаний

1. Знания в памяти человека как результат мышления.
2. Материальные носители знаний (учебники, методические пособия).
3. **Поле знаний – условное описание основных объектов предметной области, их атрибутов и закономерностей, их связывающих.**

Поле знаний, как первый шаг к формализации, представляет модель знаний о предметной области, в том виде, в каком её сумел выразить аналитик на некотором «своём» языке L .

4. Знания, описанные на языках представления знаний.
5. База знаний на материальных носителях информации.

Универсальный язык Лейбница

Лейбниц задумал и попытался разработать проекты *lingua characteristica* (язык, на котором все знания могут быть выражены формально) и *calculus ratiocinator* (исчисления умозаключений).

Он полагал, что после реализации этих проектов споры, обычные между философами, станут так же невозможны, как невозможны они между вычислителями. Вместо спора они возьмут в руки перья и скажут: «Будем вычислять».



Готфрид Вильгельм
Лейбниц
(1646-1716)

Семиотика

Лейбниц наметил основные контуры учения о символах, которые в соответствии с его замыслами в XVIII веке развивал Г. Ламберт, который дал имя науке «семиотика» (наука о знаках). В последнее время сложилась новая ветвь семиотики – прикладная семиотика.

Языки семиотического моделирования являются первым приближением к языкам представления знаний. Изменчивость и условность знаков делают семиотическую модель применимой к сложным сферам человеческой деятельности.



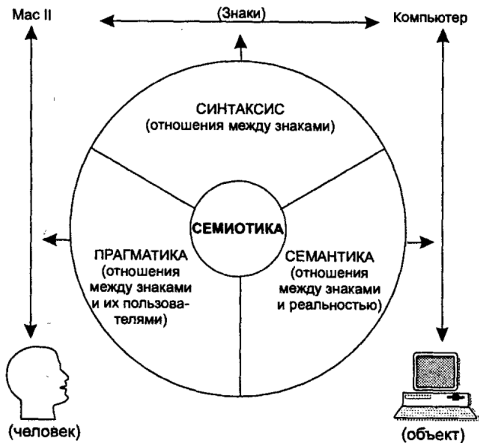
Иоганн Генрих
Ламберт
(1728-1777)

Структура семиотики

Традиционно семиотика включает:

- **синтаксис** (совокупность правил построения языка или отношения между знаками);
- **семантику** (связь между элементами языка и их значениями или отношения между знаками и реальностью);
- **прагматику** (отношения между знаками и их пользователями).

Структура семиотики

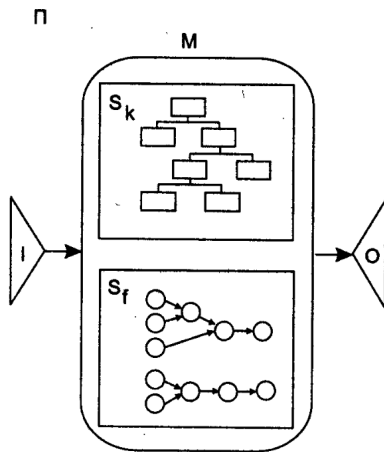


Семиотическая модель поля знаний

Синтаксис. Синтаксическую структуру поля знаний можно представить как

$$\Pi = (I, M, O),$$

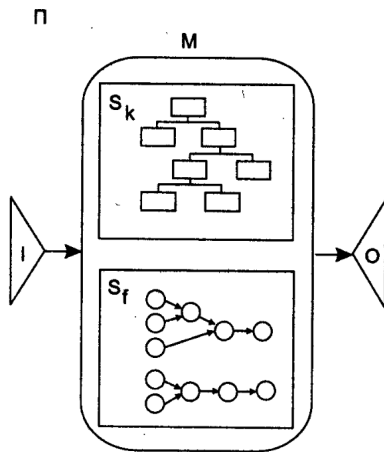
где I – структура исходных данных, O – структура выходных данных, M – операциональная модель предметной области, на основе которой происходит преобразование I в O .



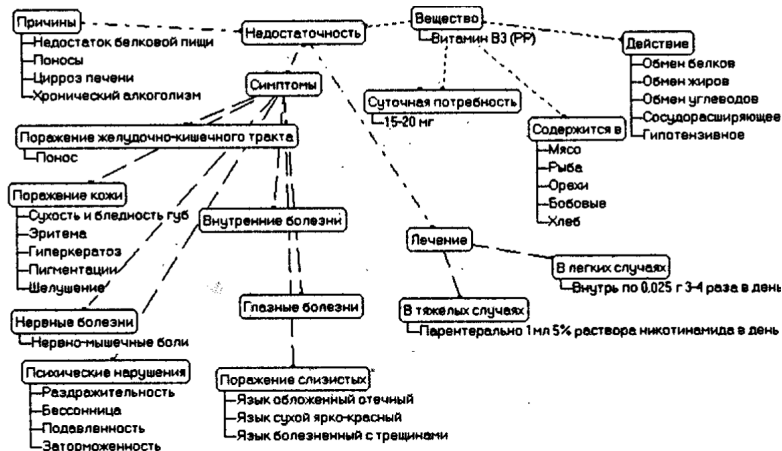
Семиотическая модель поля знаний

Операциональная модель M может быть представлена как совокупность концептуальной структуры S_k , отражающей понятийную структуру предметной области, и функциональной структуры S_f , моделирующей схему рассуждений эксперта:

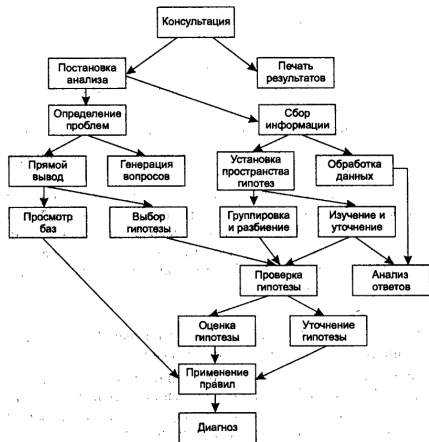
$$M = (S_k, S_f).$$



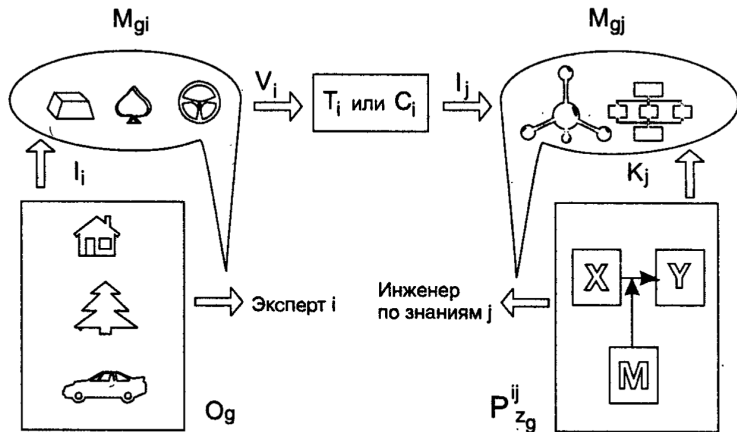
Концептуальная составляющая поля знаний



Функциональная составляющая поля знаний



Семантика поля знаний



Прагматика поля знаний

Под **прагматикой поля знаний** будем понимать практические аспекты разработки и использования поля знаний, то есть как от хаоса черновиков и стенограмм сеансов извлечения знаний перейти к стройной или хотя бы ясной модели.

Рекомендуемые материалы

1. Питер Джексон. Введение в экспертные системы // М.: «Вильямс», 2001. — 624 С.
2. CS50's Introduction to Artificial Intelligence with Python
3. Нильсон Н. Искусственный интеллект. Методы поиска решений // Москва: Мир, 1973, 273 С.
4. Гаврилова, Т. А. Базы знаний интеллектуальных систем . Учебник для вузов / Т.А. Гаврилова, В. Ф. Хорошевский.// — СПб.: Питер, 2000. – 384 с.
5. Голубниченко А. Н. Г. В. Лейбниц и автоматизация рассуждений // Вестник ОмГУ. 2011. №3.