Севастопольский государственный университет
Институт информационных технологий

Методы и системы искусственного интеллекта

Бондарев Владимир Николаевич

Агенты, основанные на моделях представления знаний и логическом выводе

В.Бондарев

Знания и их представление в СИИ

Знание – это “проверенный практикой результат познания действительности, верное её отражение в мышлении человека” (философский словарь).

Знания – сложноорганизованные данные, хранимые в памяти СИИ и включающие в себя сведения об объектах и отношениях предметной области, процессах взаимодействия объектов во времени и в пространстве, правилах осуществления логического вывода.

Важным элементом этого определения является указание на то, что ***знание*** – это информация, на основе которой выполняется ***логический вывод*.**

Знания и их представление в СИИ

Свойства знаний:

1. **Внутренняя интерпретируемость** (хранятся информационные структуры, позволяющие интерпретировать содержимое памяти)
2. **Структурированность** (наличие классифицирующих отношений: род-вид, класс-подкласс);
3. **Связность** (между информационными единицами предусматриваются связи различного типа: причина – следствие, одновременно, быть рядом и др.)
4. **Семантическая метрика** (имеются шкалы, позволяющие оценить семантическую близость инф. структур).
5. **Активность** (добавление новых фактов и связей может активизировать систему) .

Знания и их представление в СИИ

Представление знаний – это способ формального выражения знаний о предметной области в компьютерно-интерпретируемой форме. Соответствующие формализмы, обеспечивающие указанное представление, называют ***моделями представления знаний***.

Модели представления знаний можно условно разделить на ***декларативные*** **и** ***процедурные.*** В ***декларативных*** моделях знания представляются в виде описаний объектов и отношений между объектами без указания в явном виде, как эти знания обрабатывать.

В ***процедурных*** моделях знания представляются алгоритмами (процедурами).

Наиболее распространенными являются следующие модели представления знаний:

логические модели;

продукционные модели;

сетевые модели;

фреймовые модели.

База знаний

Центральным компонентом агента, основанного на знаниях, является его БЗ. ***База знаний* (БЗ, KB - knowledge base)** – множество **высказываний** на **формальном языке представления знаний** (ЯПЗ).

Машина вывода	← Алгоритмы, не зависящие от предметной области
База знаний	← Контент, специфичный для предметной области

Должен существовать определенный способ добавления новых высказываний в БЗ, а также способ извлечения знаний, которые в ней содержатся.

Стандартными названиями для этих операций являются:

Tell – сообщает агенту, что он должен знать;

Ask - запрашивает, что делать (ответ на запрос к БЗ, переданный с помощью Ask, должен **следовать** из того, что было сообщено БЗ).

Обе операции связаны с проведением **логического вывода,** т.е. могут потребовать получения новых высказываний из старых.

Агенты могут рассматриваться **на уровне знаний** т.е., что **они знают** , независимо от того как они реализованы

или **на уровне реализации**, т.е. на уровне структур данных в БЗ и алгоритмов, которые ими манипулируют

Простой агент, основанный на знаниях

Агент принимает на вход результаты восприятия и возвращает действие. Агент поддерживает базу знаний (KB), которая может первоначально содержать некоторые фоновые знания.

```
function KB-AGENT(percept) returns действие action
    static: KB, база знаний
            t, счетчик, обозначающий время, инициализированный 0
    TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
    action ← ASK(KB, MAKE-ACTION-QUERY(t))
    TELL(KB, MAKE-ACTION-SENTENCE(action, t))
    t ← t + 1
    return action
```

В процессе поиска ответа на запрос Ask проводятся исчерпывающие рассуждения в отношении текущего состояния мира, результатов возможных действий и т.д. Кроме этого, агент регистрирует свой выбор с помощью второй операции Tell и выполняет действие. Вторая операция Tell необходима для передачи в базу знаний информации о том, что гипотетическое действие action действительно было выполнено. Подробные сведения о механизме логического вывода скрыты внутри функций Tell и Ask.

В.Бондарев

Пример PEAS описания: Мир Вампуса

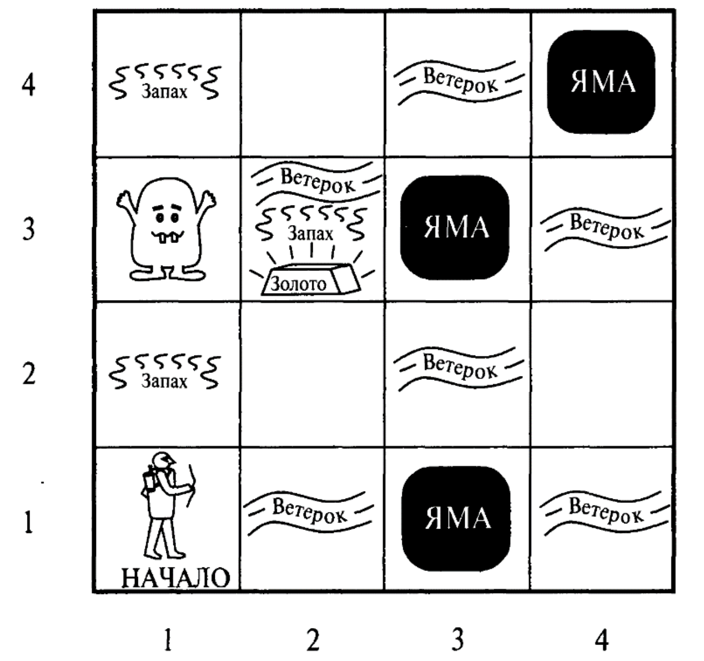
Эффективность (баллы):
золото +1000; смерть (вампус, яма) -1000;
-1 за шаг; -10 за использование стрелы;

Среда:

- квадраты с Вампусом и соседние (но не по диагонали) с ними, имеют запах;
- в квадратах, соседних с ямой, ветерок;
- в квадратах, где золото, агент видит блеск;
- выстрел стрелой убивает Вампуса, если агент столкнулся с ним;
- используется единственная стрела;
- агент схватывает золото, если находится в том же квадрате;
- при отпускании золото падает в том же квадрате.

Действия: левый поворот, правый поворот, вперед, захват, отпускание, выстрел.

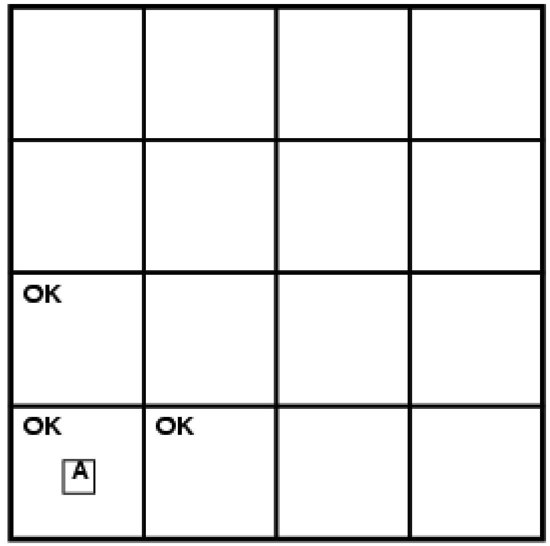
Сенсоры - Stench (запах), Breeze (ветерок), Glitter (блеск), Bump (удар о стену), Scream (крик Вампуса) : список восприятия [Stench, Breeze, None, None, None].



Основная сложность для агента состоит в том, что он не знает конфигурацию среды; для преодоления этого незнания нельзя обойтись без **логических рассуждений.**

В.Бондарев

Исследование Мира Вампуса

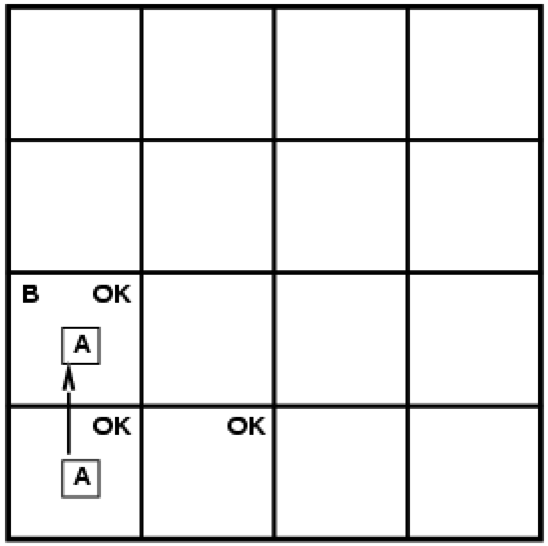


Обозначение высказываний из БЗ

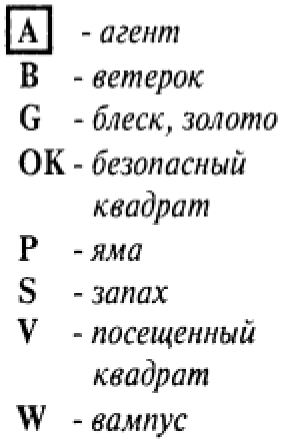
A - *агент*
B - *ветерок*
G - *блеск, золото*
OK - *безопасный квадрат*
P - *яма*
S - *запах*
V - *посещенный квадрат*
W - *вампус*

Проследим за агентом, действующим на основе знаний, который изучает среду Вампуса. Первоначальная БЗ агента содержит правила существования в этой среде, которые были описаны выше; в частности, агент знает, что находится в квадрате [1,1] и что квадрат [1,1] является безопасным. ***Мы увидим, как расширяются знания агента по мере того, как поступают результаты новых актов восприятия и выполняются действия.***

Исследование Мира Вампуса

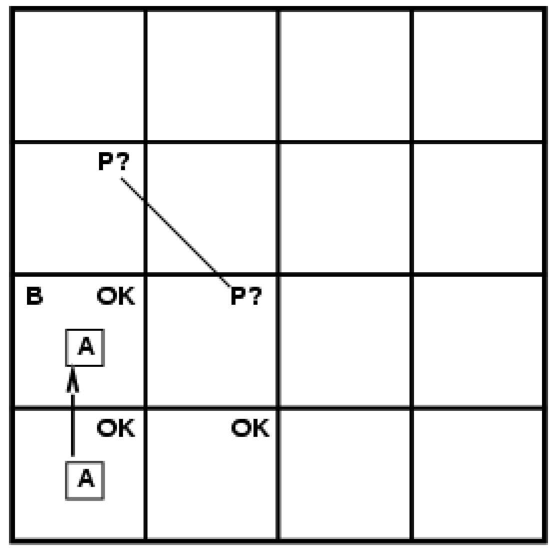


Обозначение высказываний из БЗ



На основании того факта, что в квадрате [1,1] не было ни неприятного запаха, ни ветерка, агент может сделать вывод, что квадраты [1,2] и [2,1] свободны от опасности. Для указания этого они отмечены обозначением OK. Осторожный агент переходит только в такой квадрат, о котором известно, что в нем есть отметка OK. Предположим, что агент решил двинуться вперед, в квадрат [1,2], и была создана сцена, показанная на рис.

Исследование **Мира Вампуса**

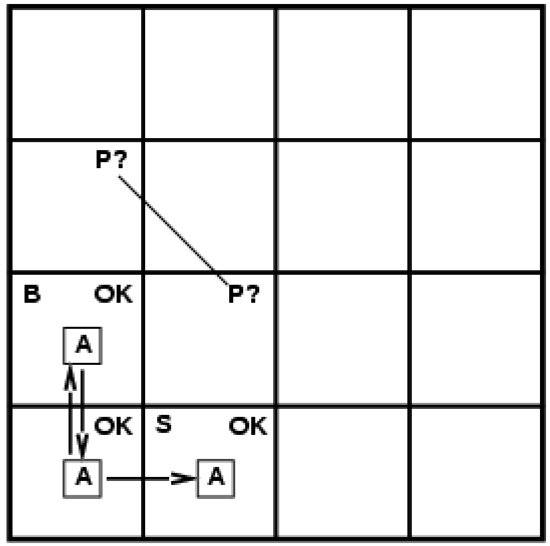


Обозначение высказываний из БЗ

$\boxed{A}$ - *агент*
B - *ветерок*
G - *блеск, золото*
OK - *безопасный квадрат*
P - *яма*
S - *запах*
V - *посещенный квадрат*
W - *вампус*

Агент обнаруживает ветерок в квадрате [1,2], поэтому в одном из соседних квадратов должна быть яма. По правилам игры яма не может находиться в квадрате [1,1], поэтому она должна быть в квадрате [2,2], или [1,3], или в том и другом. Обозначение P? указывает на возможность наличия ямы в этих квадратах.

Исследование Мира Вампуса

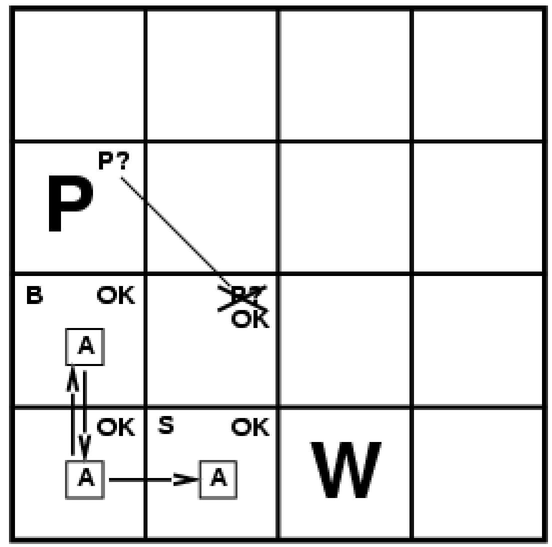


Обозначение высказываний из БЗ

- A - *агент*
- B - *ветерок*
- G - *блеск, золото*
- OK - *безопасный квадрат*
- P - *яма*
- S - *запах*
- V - *посещенный квадрат*
- W - *вампус*

В данный момент известен только один квадрат с отметкой ОК, который еще не был исследован. Поэтому благоразумный агент поворачивается кругом и возвращается в квадрат [1,1], а затем переходит в квадрат [2,1].

Исследование Мира Вампуса

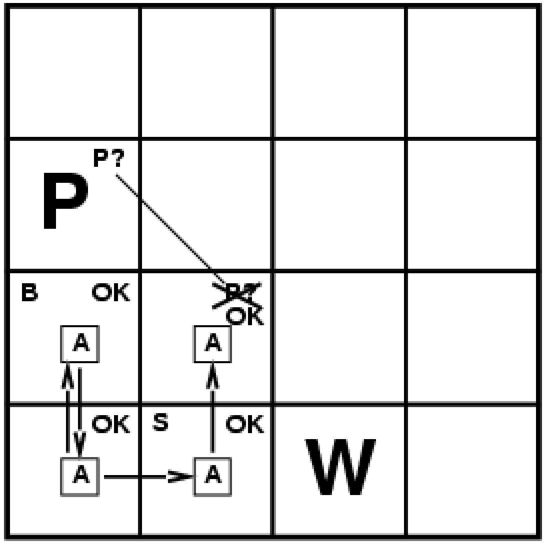


Обозначение высказываний из БЗ

- A - *агент*
- B - *ветерок*
- G - *блеск, золото*
- OK - *безопасный квадрат*
- P - *яма*
- S - *запах*
- V - *посещенный квадрат*
- W - *вампус*

Восприятием в квадрате [2,1] является [Stench, None, None, None, None]. Наличие запаха в квадрате [2,1] означает, что рядом есть вампус. Но вампус не может находиться в квадрате [1,1] по правилам игры и в квадрате [2,2] (поскольку агент обнаружил бы запах, находясь в квадрате [1,2]). Поэтому агент делает вывод, что вампус находится в квадрате [3, 1]. На это указывает обозначение W. К тому же из отсутствия восприятия Breeze в квадрате [2,1] следует, что в квадрате [2,2] нет ямы.

Исследование Мира Вампуса

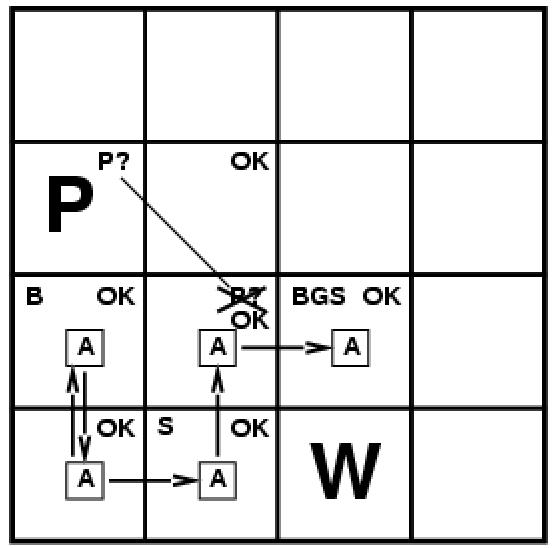


Обозначение высказываний из БЗ

- $\boxed{A}$ - *агент*
- B - *ветерок*
- G - *блеск, золото*
- OK - *безопасный квадрат*
- P - *яма*
- S - *запах*
- V - *посещенный квадрат*
- W - *вампус*

Теперь агент доказал сам себе, что в квадрате [2,2] нет ни ямы, ни Вампуса, поэтому может обозначить этот квадрат меткой OK, чтобы в него перейти.

Исследование Мира Вампуса

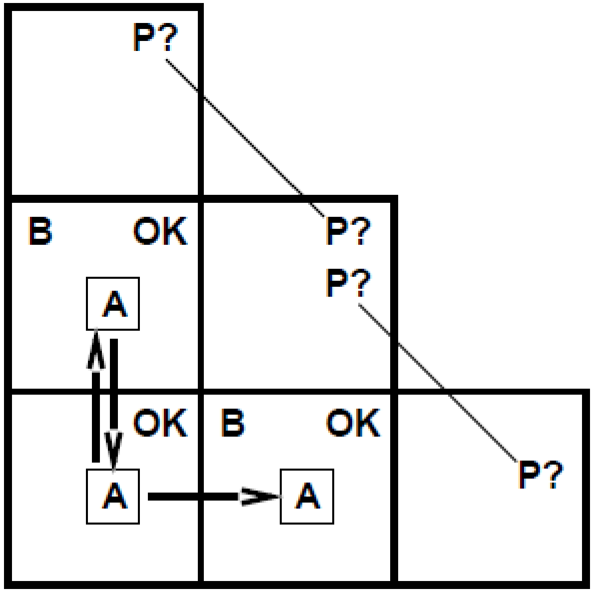


Обозначение высказываний из БЗ

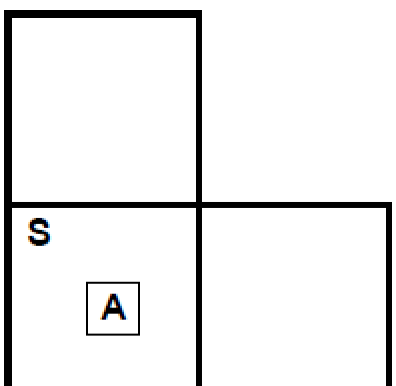
- A - *агент*
- B - *ветерок*
- G - *блеск, золото*
- OK - *безопасный квадрат*
- P - *яма*
- S - *запах*
- V - *посещенный квадрат*
- W - *вампус*

Состояние знаний агента в квадрате [2, 2] не показано, мы просто предполагаем, что агент повернулся и перешел в квадрат [3, 2], в результате чего было получено состояние, показанное на рис. В квадрате [3,2] агент обнаруживает блеск, поэтому должен схватить золото и тем самым закончить игру.

Исследование Мира Вампуса



Бриз в (1,2) и (2,1)
→ нет безопасных действий
Предположим, что ямы распределены равномерно,
В (2,2) яма - более высокая вероятность (2 свидетельства)



Запах в (1,1) – нет безопасных действий.
Можно использовать стратегию принуждения:
- стрелять прямо вперед;
- Вампус был там → мертв → безопасно;
- Вампуса там не было → безопасно.

В.Бондарев

Логические модели

Логика служит основным средством представления знаний. **Знания логических агентов** всегда являются определенными — каждое высказывание в этом мире является либо истинным, либо ложным, хотя агент может не знать о существовании некоторых высказываний.

Достоинством логических моделей представления знаний является наличие четкого синтаксиса и широко принятой формальной семантики, а также теоретически обоснованных процедур автоматического вывода.

Недостатком данных моделей является невозможность получения заключений в областях, где требуются **правдоподобные выводы**. Кроме этого, такие модели характеризуются **монотонным характером** вывода, т.е. в базу знаний добавляются только истинные утверждения, что исключает возможность противоречий. На практике часто встречаются **немонотонные рассуждения**, которые трудно реализовать в рамках логической модели.

Формальные системы (ФС)

В основе логических моделей представления знаний лежит понятие **формальной системы** (теории) , которая задается четверкой

$$\Phi=\{T, P, A, R\},$$

где ***T*** – множество базовых элементов; ***P*** – множество синтаксических правил; ***A*** – множество аксиом; ***R*** – множество правил вывода.

На **множество базовых эл-тов** ***T*** никаких ограничений не накладывается. Важно, чтобы для ***T*** **существовала процедура** проверки принадлежности некоторого элемента множеству ***T***.

Множество синтаксических правил ***P*** позволяет строить из элементов ***T*** синтаксически правильные совокупности базовых элементов. Требуется, чтобы **существовала процедура**, которая позволяла бы за конечное число шагов дать однозначный ответ на вопрос, является ли данная совокупность элементов из ***T*** синтаксически правильной. Такие совокупности называют ***правильно построенными формулами*** (ППФ).

В.Бондарев

Формальные системы

Среди всех ППФ выделяют некоторое подмножество **аксиом** A. При этом должна **существовать процедура**, позволяющая для любой ППФ решить вопрос, является ли она аксиомой данной ФС.

Множество *R* – это конечное множество отношений между ППФ, называемых ***правилами вывода***. **Правило вывода** – это отношение на множестве формул. Если из формул ***F1, F2, …,Fn*** непосредственно выводится формула ***F***, то это часто записывается в виде

$$\frac{F_1, F_2, \ldots, F_n}{F},$$

где формулы ***F1, F2, …,Fn*** называются ***посылками правила***, а ***F*** – его ***следствием (заключением).***

В.Бондарев

ФС: вывод

Выводом формулы ***B*** из формул ***A1,A2,..,An*** в ФС ***Ф называется последовательность ППФ F1, F2, …,Fm*** такая, что ***Fm=B***, а для любого i ($1 \le i \le m$) формула ***Fi*** есть либо аксиома ФС, либо одна из исходных формул ***A1,A2,..,An***, либо непосредственное следствие формул ***F1, F2, …,Fi-1***, полученное с помощью правил вывода.

Некоторая ППФ ***B*** является ***выводимой*** в ФС, если существует вывод, в котором последней формулой является ***B***.
Сокращенно вывод ***B*** из ***A1, A2, …,An*** в теории ***Ф*** будем записывать в виде

$$A1, A2, \ldots, An \mid\!-_{\text{ф}} B,$$

где ***A1, A2, …,An*** называются посылками или гипотезами вывода. Если теория (ФС) ***Ф*** подразумевается, то ее обозначение обычно опускают .

ФС: интерпретации и модели

Интерпретацией ФС ***Ф*** в области интерпретации ***O*** называется функция ***I:Ф→O,*** которая каждой формуле ФС однозначно сопоставляет некоторое содержательное высказывание относительно объектов множества ***O*.**

Это высказывание может быть истинным или ложным (или не иметь истинностного значения). Если высказывание является истинным, то говорят, что формула **выполняется** в данной интерпретации.

Интерпретация ***I*** также называется **моделью** ***M*** множества формул**,** если все формулы этого множества выполняются в интерпретации ***I*.**

ФС: логическое следствие

Формула **α** называется **логическим следствием** множества формул ***B***, если **α** выполняется в любой модели (интерпретации) ***B***:

$$B \Rightarrow \alpha \text{ или } B \models \alpha$$

Например, ***KB* |= α** означает, что **α** следует из базы знаний ***KB***, е.е. **α** истинно во всех интерпретациях (мирах) , в которых ***KB*** истинна.
Если база знаний ***KB*** содержит 2 утверждения:

“Гиганты победили” и “Красные победили”,

то отсюда, например, следует *“Гиганты победили”* - верное утверждение.

Например, из x + y =4 следует 4=x + y, т.к.
в любой модели, в которой x+y=4 справедливо, также справедливо, что 4=x+y.

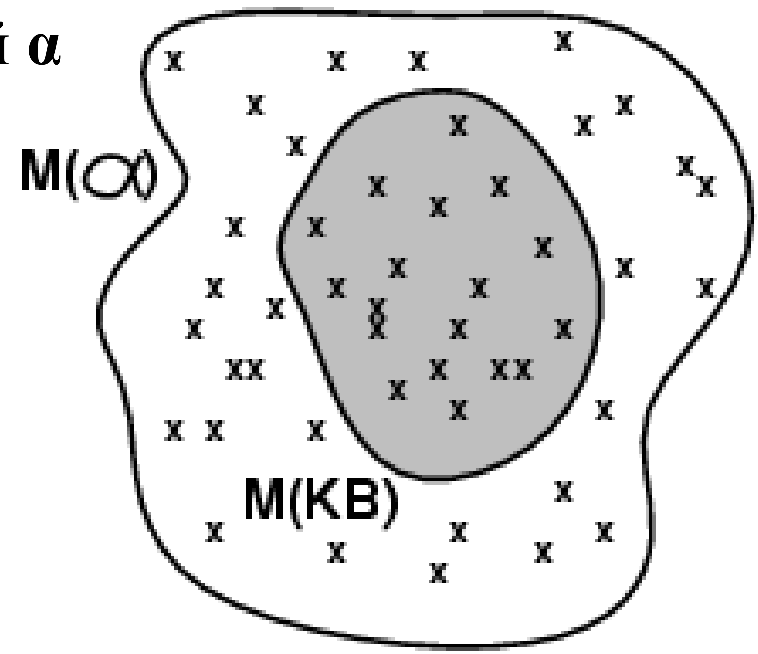
Следование это отношение между формулами.

ФС: модели

Логики оперируют моделями, которые структурируют мир путем оценивания истинности утверждений.

Можно сказать, что ***m*** модель утверждения **α**, если **α** истинно в ***m***

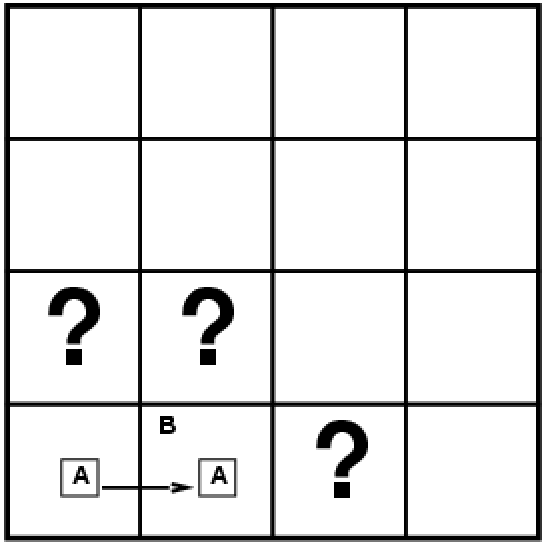
Пусть ***M(α)*** -- множество всех моделей **α**



Тогда ***KB*** |= **α,** е.е. ***M(KB)*** $\subseteq$ ***M*(α)**

Выводы в мире Вампуса

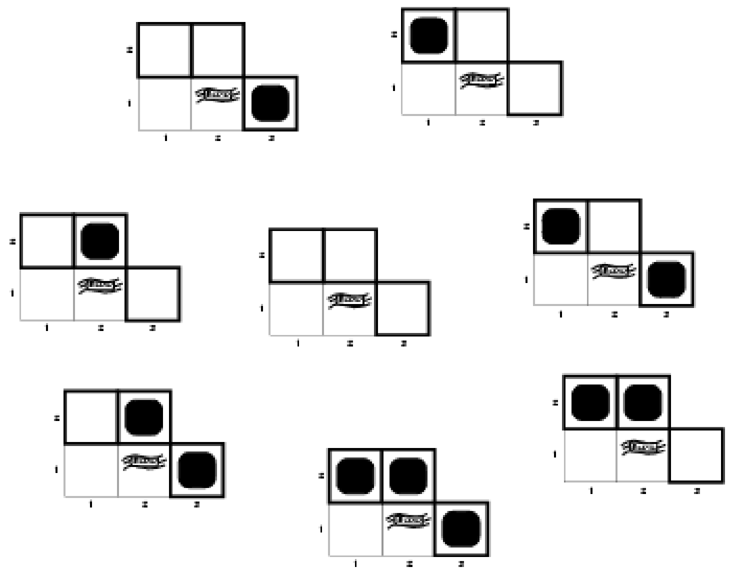
Ситуация, когда агент ничего не обнаружил в квадрате [1,1], а в квадрате [2,1] почувствовал ветерок. Эти восприятия в сочетании со знаниями агента о правилах мира Вампуса составляют его БЗ



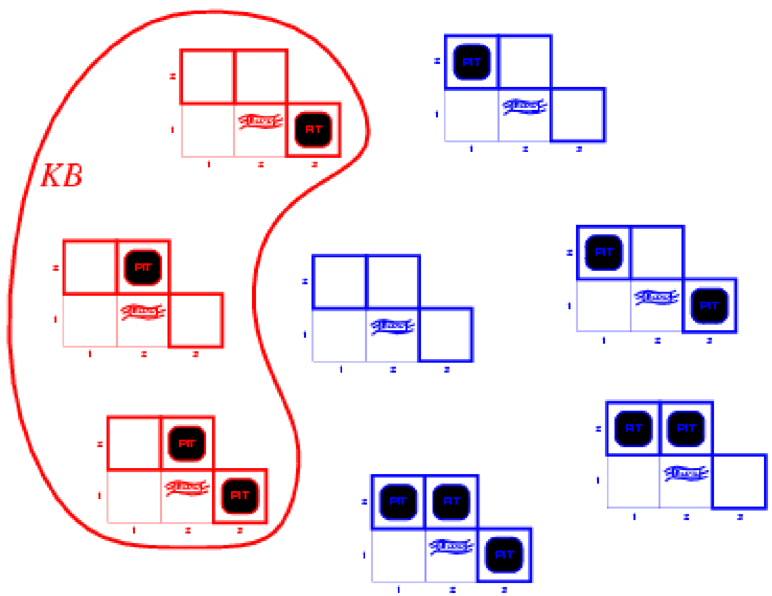
Агенту необходимо узнать (кроме всего прочего), имеются ли ямы в соседних квадратах, [1,2], [2,2] и [3,1].

3 бинарных состояния $\Rightarrow$ дают 8 возможных моделей (интерпретаций)

Модели в мире Вампуса



Модели в мире Вампуса

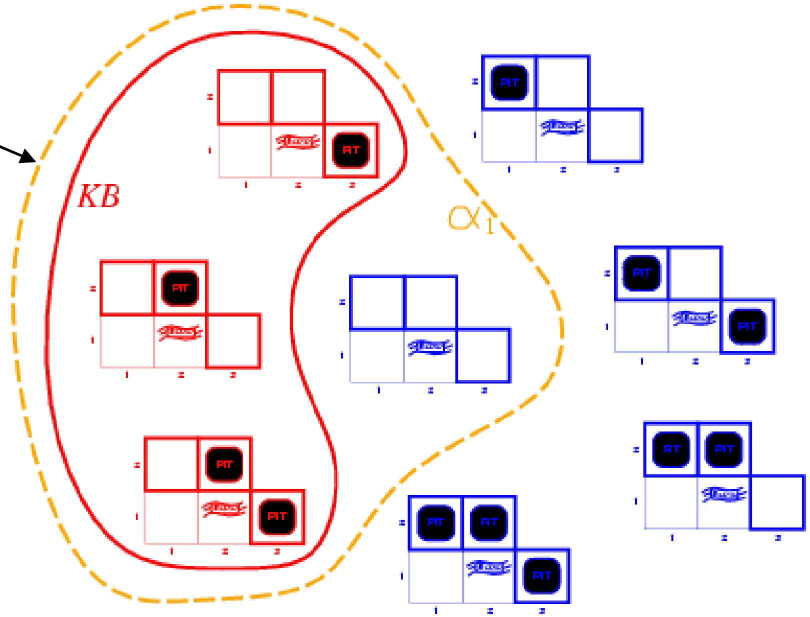


- *KB* определяется правилами мира Вампуса + восприятия -- в квадрате [1,1] ничего не обнаружено, а в [1,2] – ветерок

Модели в мире Вампуса

Док-во заключения проверкой по моделям

«В квадрате[1,2] нет ямы",

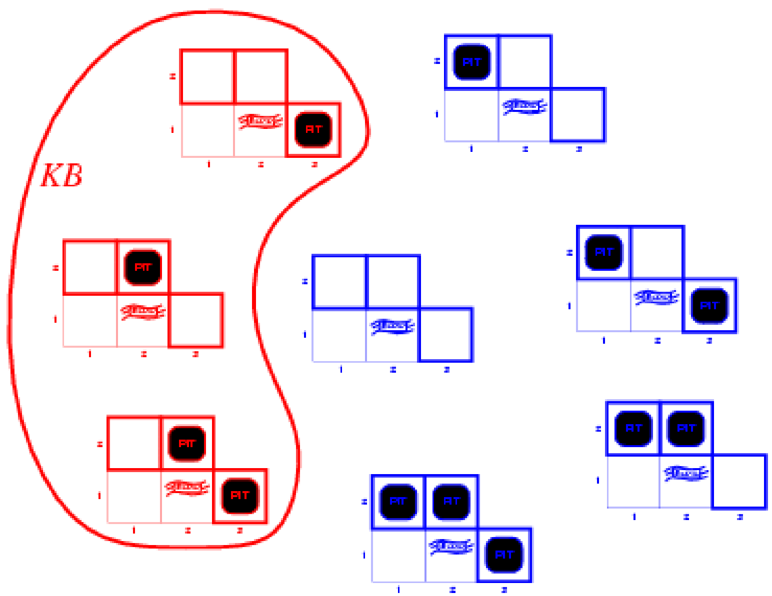


- *Состояние KB* = правила мира Вампуса + восприятия
- α_1 = «в квадрате [1,2] нет ямы", $KB \models \alpha_1$, доказывается путем ***проверки по моделям*** (перебор всех возможных моделей для проверки того, что утверждение является истинным во всех моделях),

В.Бондарев

Модели в мире Вампуса

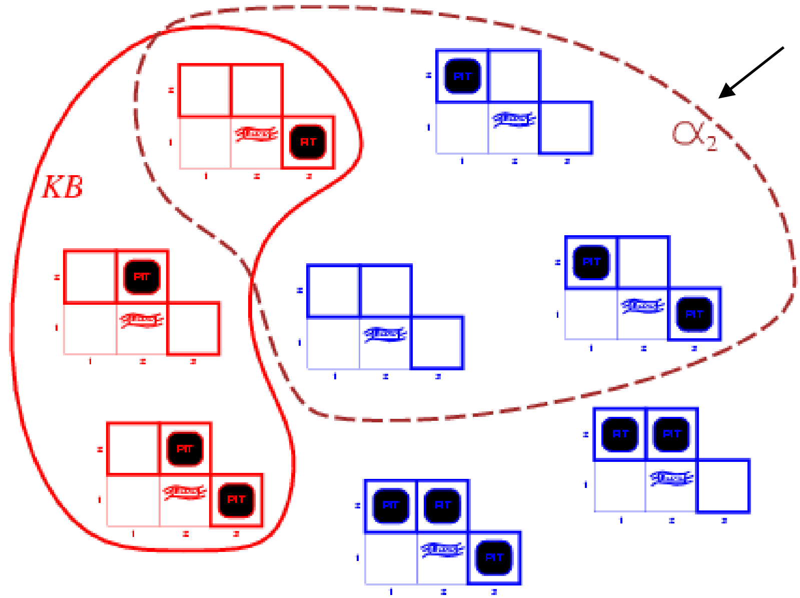
Док-во заключения проверкой по моделям



- *Состояние KB* = правила мира Вампуса + восприятия

Модели в мире Вампуса

Док-во заключения проверкой по моделям



«В квадрате[2,2] нет ямы"

- *Состояние KB* = правила мира Вампуса + восприятия
- α_2 = «В квадрате[2,2] нет ямы", $KB \not\models \alpha_2$ - агент не может прийти к заключению, что в квадрате [2,2] нет ямы. (К тому же он не может сделать и вывод, что в квадрате [2,2] есть яма.)

Непротиворечивость и полнота (определения)

- ***KB*** $\vdash_i$ $\boldsymbol{\alpha}$ - утверждение $\boldsymbol{\alpha}$ может быть логически выведено из ***KB*** с помощью **процедуры** **i** , т.е. алгоритм логического вывода i позволяет вывести логическим путем высказывание $\boldsymbol{\alpha}$ из базы знаний ***KB.***

- **Непротиворечивость**: процедура ***i*** непротиворечива, если вывод ***KB*** $\vdash_i$ $\boldsymbol{\alpha}$, позволяет получать утверждения ***KB*** $\models$ $\boldsymbol{\alpha}$**,** которые действительно следуют из БЗ (генерируются только верные утверждения). Алгоритм проверки по моделям является непротиворечивым.

- **Полнота:** процедура ***i*** является полной, если каждому истинному высказыванию ***KB*** $\models$ $\boldsymbol{\alpha}$ соответствует вывод ***KB*** $\vdash_i$ $\boldsymbol{\alpha}$, т.е. алгоритм логического вывода называется полным, если позволяет вывести любое высказывание, которое следует из базы знаний.

ФС: разрешимость

Если **существует** эффективная **процедура,** позволяющая по данной ППФ ***F*** устанавливать, существует ли ее вывод в ФС, то данная ФС называется ***разрешимой,*** в противном случае – ***неразрешимой***.

ФС называется ***полуразрешимой***, если существует процедура, которая для любой формулы ***F*** дает ответ "Да", если ***F*** является выводимой и, может быть, не выдает никакого ответа, если ***F*** не является выводимой (т.е. процедура применима не ко всем формулам)

Формальные системы - ЯПЗ

Притягательной стороной ФС с точки зрения представления знаний является то, что ее можно рассматривать как генератор новых знаний. В этом случае из множества аксиом ***A***, представляющих собой знания, изначально хранящихся в базе знаний СИИ, и известных фактов выводятся с помощью правил вывода производные знания.

Рассмотрим два класса ФС, широко используемых в системах искусственного интеллекта: **исчисление высказываний и исчисление предикатов.** Данные системы используют модель ***дедуктивного вывода,*** т.е. вывода, при котором из заданной системы посылок с помощью фиксированного набора правил формируются частные заключения.

Исчисление высказываний

Высказыванием называется предложение, содержание которого можно оценить как истинное или ложное. Например, "Сегодня ясная погода", "Пять меньше трех" и т. п.

Будем обозначать высказывания прописными латинскими буквами ***A,B,C,..,X,Y,Z*** и называть их ***пропозициональными*** (лат. propositio – предложение, высказывание). Пропозициональные буквы могут принимать два значения: истина (*И*) и ложь (*Л*).

На основе простых высказываний с помощью логических связок (союзов) образуются сложные высказывания. Для обозначения данных связок используются специальные символы: $\wedge, \vee, \rightarrow, \overline{A}$ (или $\neg A$).

Исчисление высказываний

Таблица истинности простейших высказываний

A	B	$\overline{A}$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
Л	*Л*	*И*	*Л*	*Л*	*И*	*И*
Л	*И*	*И*	*Л*	*И*	*И*	*Л*
И	*Л*	*Л*	*Л*	*И*	*Л*	*Л*
И	*И*	*Л*	*И*	*И*	*И*	*И*

Пример импликации: “*Если я пойду на стадион (А), то встречу друга (В)*”. В импликации первый элемент A называется ***антецедентом*** (лат. antecedens – предшествующий), а второй элемент B – ***консеквентом*** (consequens – последующий).

Эквиваленция читается “если и только если A, то B”. Она истинна тогда и только тогда, когда A и B имеют одно и то же истинностное значение. Эквиваленцией будет следующее высказывание: “*Я пойду на стадион (А), тогда и только тогда, когда встречу друга (В)*”. Для обозначения эквиваленции употребляется также знак $\equiv$.

Исчисление высказываний

Исчисление высказываний как ФС

1. Множество базовых элементов ***T*** состоит из:
- пропозициональных переменных – *A, B, C,…*;
- логических констант – “истина”(*И*) и “ложь”(*Л*);
- символов логических операций – $\wedge, \vee, \rightarrow, \leftrightarrow, \neg$;
- скобок – ( ).

2. ППФ определяются с помощью следующих правил:
а) всякая пропозициональная буква есть ППФ;
б) логические константы *И* и *Л* – это ППФ;
с) если F_1 и F_2 ППФ, то– $\overline{F_1}, F_1 \wedge F_2, F_1 \vee F_2, F_1 \rightarrow F_2, F_1 \leftrightarrow F_2$ также ППФ;
- других правил образования ППФ нет.

Правила а) и б) определяют ***элементарные формулы*** (***атомы***), а правило с) указывает, как из элементарных формул строить новые формулы.

Исчисление высказываний

Пусть формула ***F*** состоит из атомов ***A1,A2,…,An***. Под ***интерпретацией*** формулы будем понимать приписывание атомам ***A1,A2,…,An*** истинностных значений.

Формулы, которые истины во всех интерпретациях называются ***тавтологиями*** или ***общезначимыми*** формулами. Для указания того, что формула является тавтологией, используется знак $\models$.
Например, $\models \; A \vee \overline{A}$.

Формула исчисления высказываний называется ***противоречием,*** если она принимает значение “ложь” во всех интерпретациях.
Примером противоречия является $A \wedge \overline{A}$.

Различные подстановки в тавтологию, независимо от их конкретного содержания, всегда дают истинные высказывания.
Поэтому тавтологии рассматривают как логически истинные схемы рассуждений, которые играют роль ***законов*** (modus) исчисления высказываний.

В.Бондарев

Исчисление высказываний

В исчислении высказываний также рассматриваются *отношения* между двумя и более формулами. Определим отношения равносильности и логического следования.

Две формулы называют ***равносильными*** **(**эквивалентными), если они принимают одинаковые значения на всех наборах входящих в них переменных (интерпретациях). Для обозначения равносильности применяется знак $\Leftrightarrow$ (или знак тождества $\equiv$, или знак равенства $=$), например, $(A \to B) \Leftrightarrow (\overline{A} \vee B)$.

Процесс дедуктивного вывода базируется на понятии логического следования. Формула ***B*** является ***логическим следствием*** *формулы* ***A*** (пишут $A \Rightarrow B$ или $A \models B$), если ***B*** истинна на всех наборах значений переменных (интерпретациях), на которых истинна ***A***. Можно показать, что , $A \Rightarrow B$ если и только если импликация $A \to B$ является тавтологией, т.е. $\models A \to B$.

В.Бондарев

Исчисление высказываний

Понятие логического следствия можно обобщить на совокупность высказываний

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n \Rightarrow B \quad ,$$

т.е. ***B*** логически следует из истинных высказываний ***A1,A2,…,An***.
 Можно показать, что формула ***B*** является логическим следствием последовательности формул ***A1, A2,…, An*** тогда и только тогда, когда формула

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n \wedge \overline{B} \qquad (1)$$

является **противоречием**.
Таким образом, доказательство того, что формула *B* является логическим следствием конечной последовательности формул *A1, A2,…, An*, сводится к доказательству **общезначимости** формулы

$$\models A_1 \wedge A_2 \wedge \ldots \wedge A_n \to B$$

или доказательству **противоречивости** формулы (1)

Исчисление высказываний

3. Аксиомы. Известны различные системы аксиом. Аксиомы Клини, 1952:

1) $A \to (B \to A)$;
2) $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$;
3) $(A \wedge B) \to A$;
4) $(A \wedge B) \to B$;
5) $(A \to (B \to (A \wedge B))$;
6) $A \to (A \vee B)$;
7) $B \to (A \vee B)$;
8) $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$;
9) $(A \to B) \to ((A \to \overline{B}) \to \overline{A})$;
10) $\overline{\overline{A} \to A}$.

Данная система аксиом обладает следующими важными свойствами: полнотой, непротиворечивостью и независимостью.

Теорема о полноте исчисления высказываний утверждает, что если ППФ ***A*** общезначима, то она является **выводимой.**

Исчисление высказываний

4. Множество правил вывода ***R*** исчисления высказываний задается двумя правилами: правилом подстановки и правилом заключения.

Правило подстановки. Если ***Ф*** – выводимая формула (тавтология), содержащая букву ***A***, то, заменив в ней всюду букву ***A*** на произвольную ППФ ***B***, получим также выводимую формулу (тавтологию). Данное правило можно записать в виде:

$$\frac{\Phi(A)}{\Phi(B)}$$

Правило заключения (modus ponens). Если ***A*** и $A \to B$ – выводимые формулы (тавтологии), то ***B*** – также тавтология. Данное правило записывается в виде:

$$\frac{A, A \to B}{B}.$$

Исчисление высказываний

В исчислении высказываний часто применяют **дополнительные правила вывода**, которые значительно сокращают путь доказательства теорем. Например, ***правило силлогизма***:

$$\frac{(A \to B), (B \to C)}{A \to C}.$$

Применение правил вывода для получения следствий опирается на свойство ***монотонности*** исчисления высказываний. Монотонность означает, что добавление в БЗ нового множества выводимых высказываний (формул), например M_2, не повлияет на выводимость формулы $\boldsymbol{A}$, т.е. если $M_1 \mid\!- A$, то и $\{M_1 \cup M_2\} \mid\!- A$.

В заключение отметим, что исчисление высказываний – это **разрешимая формальная система**. Это следует из теоремы о полноте исчисления высказываний, которая утверждает, что в исчислении высказываний выводима любая общезначимая ППФ.

Агенты, основанные на пропозициональной логике

Агент в мире Вампуса начинает свою работу с базы знаний, в которой описаны общие "законы" мира Вампуса:

$\neg P_{1,1}$ - в квадрате [1,1] нет ямы;
$\neg W_{1,1}$ - в квадрате [1,1] нет Вампуса;
$B_{x,y} \Leftrightarrow (P_{x,y+1} \vee P_{x,y-1} \vee P_{x+1,y} \vee P_{x-1,y})$ - как возникает ветерок;
$S_{x,y} \Leftrightarrow (W_{x,y+1} \vee W_{x,y-1} \vee W_{x+1,y} \vee W_{x-1,y})$ – как возникает запах;
$W_{1,1} \vee W_{1,2} \vee \ldots \vee W_{4,4}$ - по меньшей мере имеется один Вампус.

Существует самое большее один Вампус: (*при наличии любых двух квадратов один из них обязательно должен быть свободным от Вампуса*):
$\neg W_{1,1} \vee \neg W_{1,2}$ - существует самое большее один Вампус ;
$\neg W_{1,1} \vee \neg W_{1,3}$ - существует самое большее один Вампус ;
….
$\Rightarrow$ 64 пропозициональных символа, 155 высказываний.

```
function PL-Wumpus-Agent(percept) returns действие action
    inputs: percept, список результатов восприятия в форме
                [Stench,Breeze,Glitter] (["Неприятный запах",
                 "Ветерок","Блеск"])
    static: KB, база знаний, первоначально содержащая лишь определения
                "законов функционирования" мира вампуса
            x, y, orientation, позиция агента (первоначально 1,1) и его
                               ориентация (первоначально Right - смотрит
                               вправо)
            visited, массив, показывающий, какие квадраты были посещены,
                     первоначально содержащий значения false
            action, последнее по времени действие агента, первоначально
                    не определено
            plan, намеченная последовательность действий, первоначально
                  пустая

    обновить значения x, y, orientation, visited
       с учетом действия action
    if stench then Tell(KB, S_x,y) else Tell(KB, ¬S_x,y)
    if breeze then Tell(KB, B_x,y) else Tell(KB, ¬B_x,y)
    if glitter then action ← grab
    else if plan не пуст then action ← Pop(plan)
    else if для некоторого периферийного квадрата [i,j]
            результат Ask(KB, (¬P_i,j ∧ ¬W_i,j)) имеет значение true or
            для некоторого периферийного квадрата [i,j] результат
            Ask(KB, (P_i,j ∨ W_i,j)) имеет значение false then do
          plan ← A*-Graph-Search(Route-Problem([x,y],orientation,
                                       [i,j],visited))
          action ← Pop(plan)
    else action ← случайно выбранный шаг
    return action
```

Заключение

Логические агенты применяют вывод к базе знаний для получения новой информации и принятия решений.

Основные понятия логики:

синтаксис: определяет формальную структуру предложений;

семантика: определяет истинность предложений в соответствующей модели;

следствие: определяет истинность одного предложения при заданной истинности другого;

вывод: соответствует цепочке предложений при доказательстве других предложений;

непротиворечивость: процедура вывода генерирует только верные утверждения;

полнота: процедура вывода генерирует любое высказывание, выводимое из БЗ;

Мир Вампуса требует способности представлять частичную информацию, отрицания, рассуждения по шаблонам и т. д.

Логике высказываний не хватает выразительной силы.