



*Thinking is not the ability to manipulate **language**;
it's the ability to manipulate **concepts**. Computer
science **should be about concepts**, not languages.
Лесли Лэмпорт*

Курс: Цифровые технологии в научных исследованиях

ЛЕКЦИЯ 2: МОДАЛЬНАЯ ЛОГИКА АЛГОРИТМОВ



Технологии обещают много, ... но
облик будущего зависит только от нас -

СОЗИДАТЕЛЕЙ НОВЫХ ТЕХНОЛОГИЙ



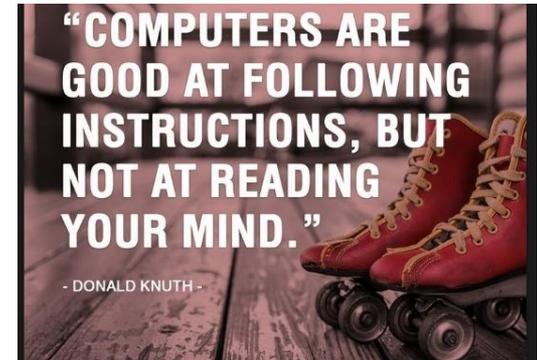
Проблема в том, что

системы, которые мы строим, ведут себя в точности так, как мы их запрограммировали какие данные обрабатывают, используя разрешенный набор операций частично-рекурсивных функций

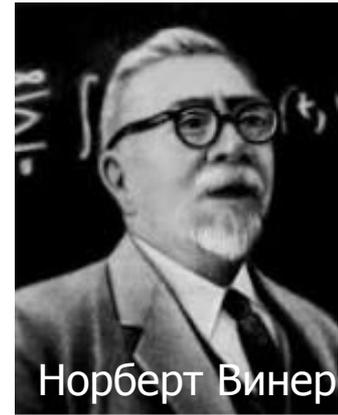
НО

совсем не так, как мы предполагали!

Человеческий мозг не в состоянии справиться со сложностью при разработке программ объемом в сотни миллионов строк кода



“Если магия вообще способна **даровать** что-либо, то она дарует именно то, что вы **попросили**, а не то, что вы подразумевали, но не сумели точно **сформулировать**.”

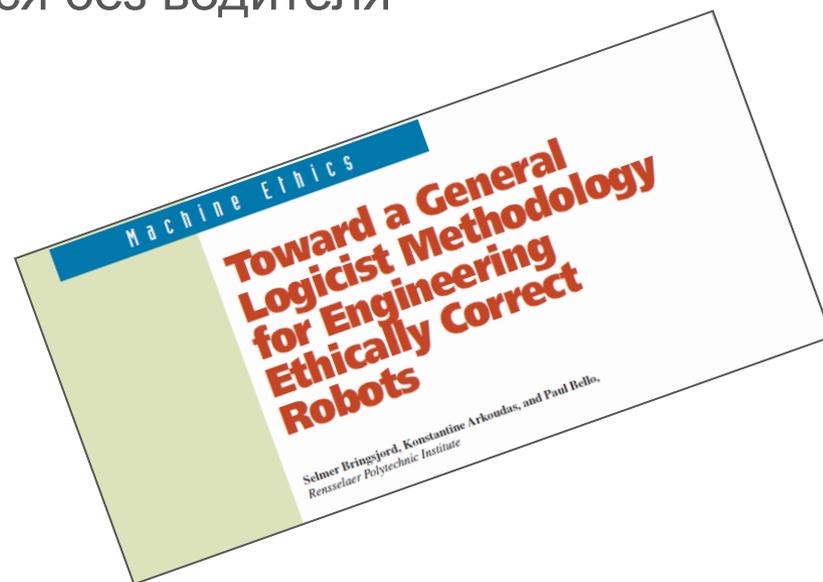
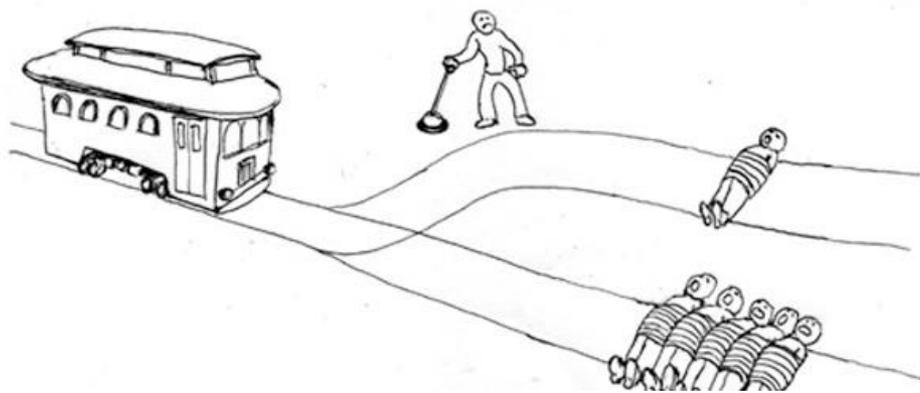


...исполнение заданного
осуществляется вычислительными
машинами в высшей степени буквально ...так это написано
в программе по правилам формальной логики и
компьютерной арифметики

*«Компьютеры похожи на ветхозаветных богов:
сплошные правила и ни капли жалости».*

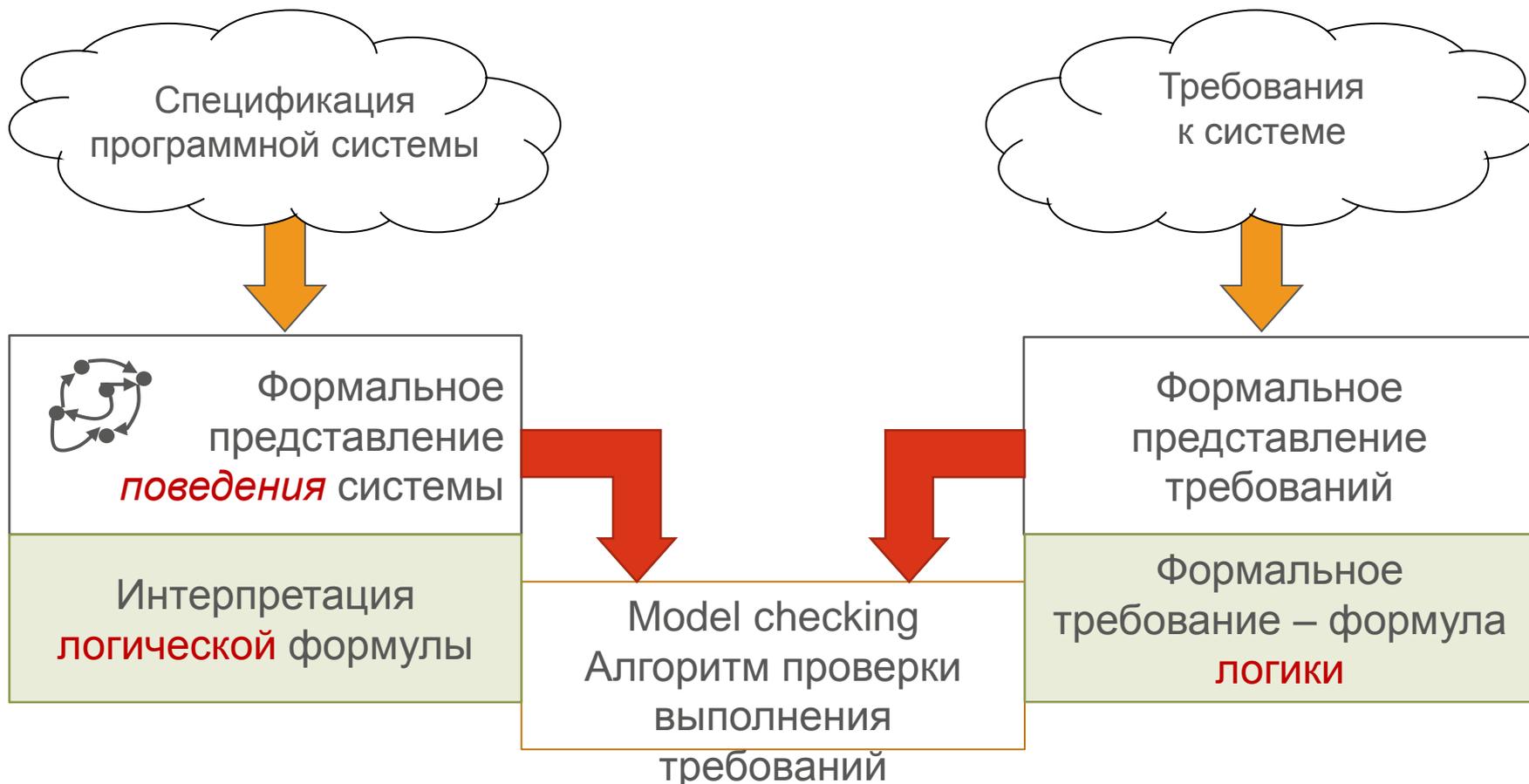
(Дж. Кэмпбелл)

К 2025 году в мире ежегодно будет продаваться 15 млн автомобилей с автопилотами, которые будут обходиться без водителя

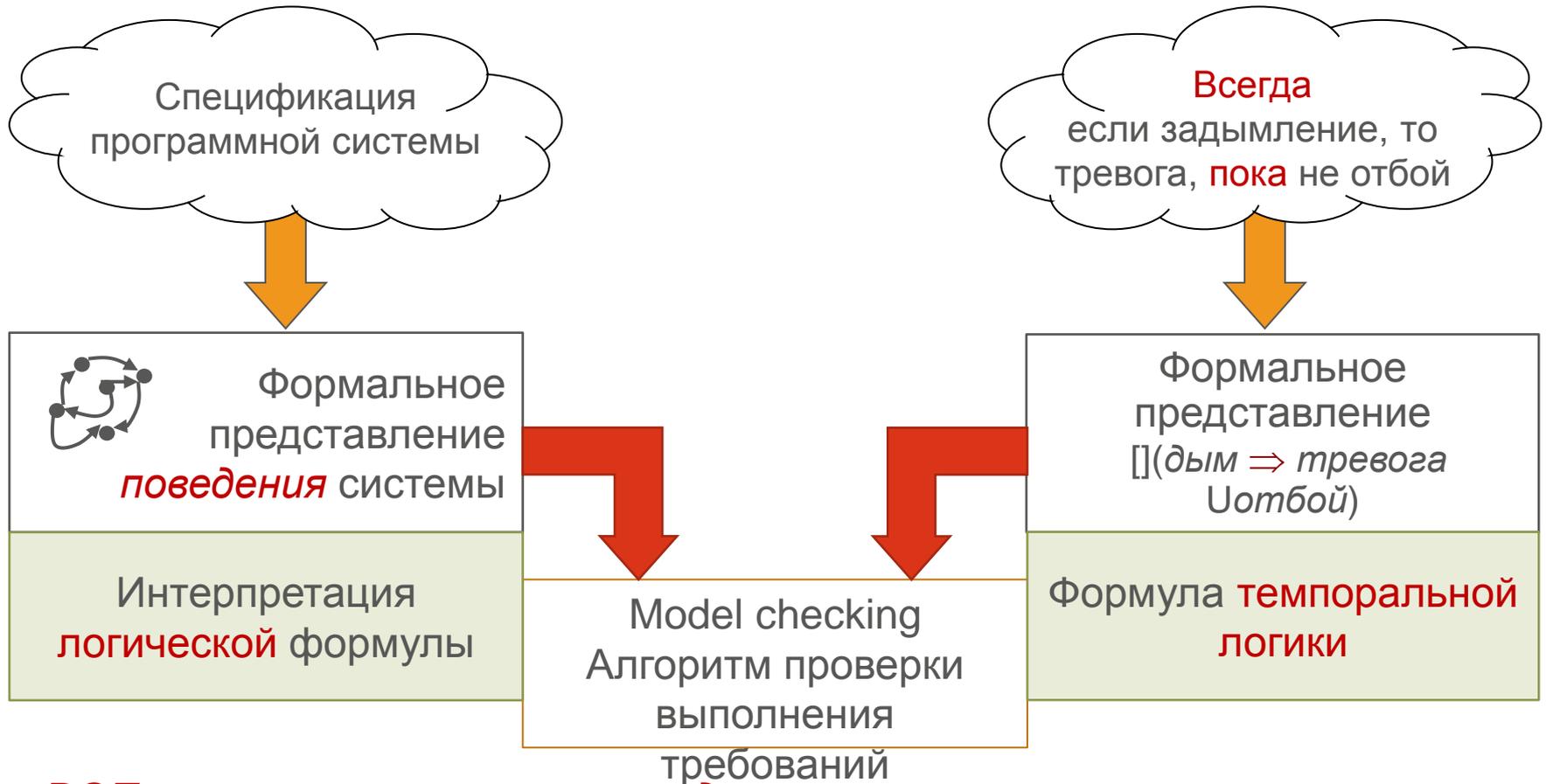


Какую альтернативу выберет автопилот, **если одна из них неизбежна**: задавит одного ребенка или пять мужиков?

КАК МОЖНО ГАРАНТИРОВАТЬ, ЧТО ТРЕБОВАНИЯ БУДУТ ВЫПОЛНЯТЬСЯ ПРИ ПОВЕДЕНИИ ПРОГРАММНЫХ АГЕНТОВ?

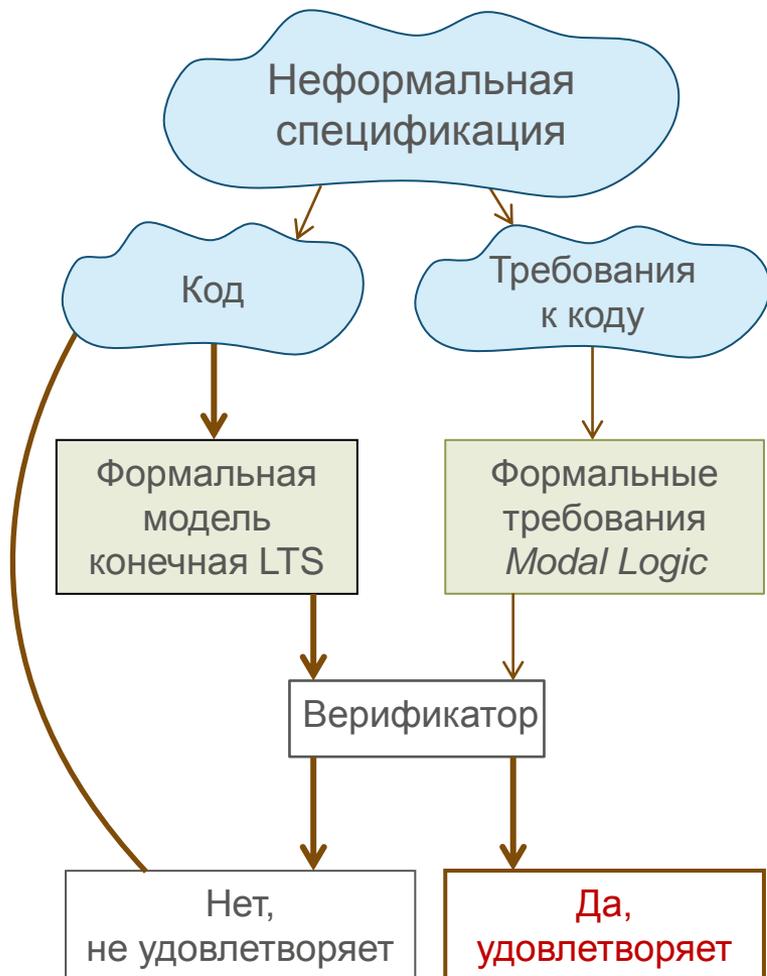


- **Model checking** – алгоритм проверки того, что логическая формула на данной интерпретации выполняется - принимает значение “истина”

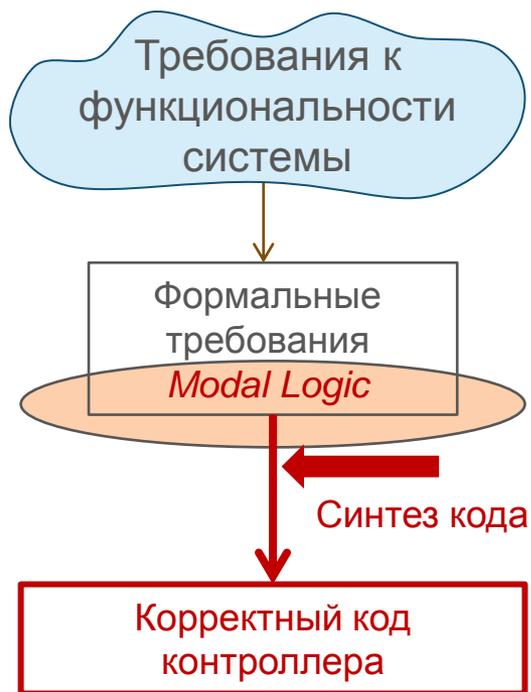


ВСЕ алгоритмы управления должны проверяться относительно их БУДУЩЕГО поведения во времени

Классический подход



Синтез супервизоров



В области построения контроллеров для координации действий распределенных дискретных 'событийных' систем получены интересные результаты



ПОЛИТЕХ

СОВРЕМЕННАЯ ПАРАДИГМА РЕШЕНИЯ ИНЖЕНЕРНЫХ ПРОБЛЕМ: ИСПОЛЬЗОВАНИЕ ФОРМАЛЬНЫХ МОДЕЛЕЙ

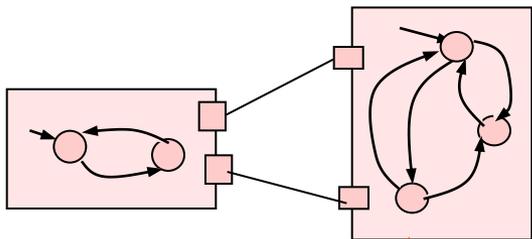
**Модель
должна быть
адекватной**

Формальная
модель

$$A=(S,U,V, \delta,\varphi);$$

$$\delta: U \times V \rightarrow S$$

...



Анализ,
оптимизация,
эквивалентные
преобразования

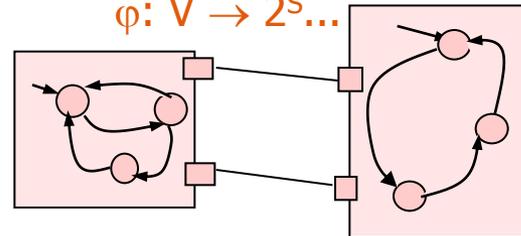


Оптимизированная,
преобразованная модель

$$A=(S,U,V, \delta,\varphi);$$

$$\delta: U \times V \rightarrow S;$$

$$\varphi: V \rightarrow 2^S \dots$$

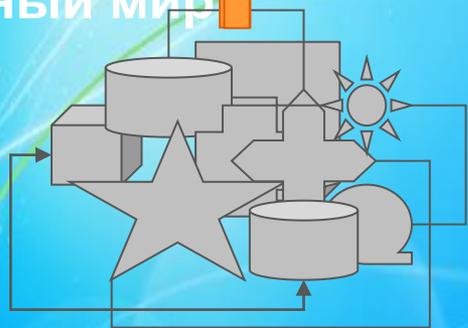


Мир моделей

Абстрагирование

Конкретизация

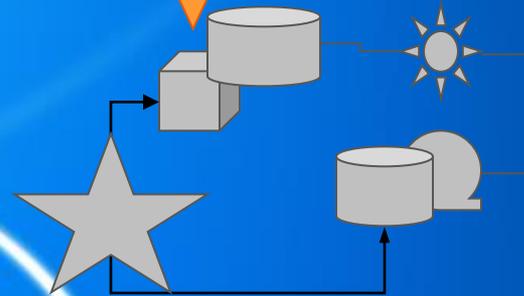
Реальный мир



Проблема



Эксперименты
(метод "тыка")



Решение



ПОЛИТЕХ

СОВРЕМЕННАЯ ПАРАДИГМА РЕШЕНИЯ ИНЖЕНЕРНЫХ ПРОБЛЕМ: ИСПОЛЬЗОВАНИЕ ФОРМАЛЬНЫХ МОДЕЛЕЙ

**Модель
должна быть
адекватной**

Формальная
модель

Анализ,
оптимизация,
... ..

Оптимизированная,
преобразованная модель

$F = ABC \vee$
 $AB \neg C \vee$
 $A \neg BC \vee$
 $\neg ABC$



При построении
цифровых устройств
формальная модель –
булева логика

$F = AB \vee BC \vee AC$

Абстрагирование

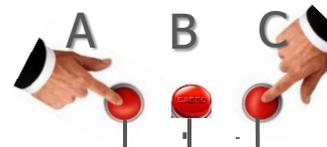
Интерпретация

Мир моделей

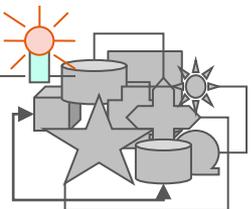
Реальный мир



Проблема:
*прибор,
засчитывающий
удар, если его
зафиксирует
большинство из 3-х
судей*



Решение



**В XXI веке большинство приложений логики
будет приходиться
на новый вид логики: на
МОДАЛЬНУЮ ЛОГИКУ**

Утверждение: Джон умер, и его похоронили.

Для формализации этих утверждений классическая логика неадекватна

“Джон умер, и его похоронили” ~~≠~~ “Джона похоронили, и он умер”

Нарушение коммутативности:

$$A \wedge B \not\equiv B \wedge A$$

Характеристики истины: *`всегда`, `верю`, `разрешено`,
`знаю`, `когда-нибудь в будущем`,*



ИСТИНЫ НЕПРЕЛОЖНЫЕ И ОТНОСИТЕЛЬНЫЕ

В **классической** логике понятие истины полностью очищено от любых сопутствующих соображений и оттенков.

Атомарные утверждения представляют факты:

-- каждое из них либо **истинно**, либо **ложно**.

Сложные утверждения – СТАТИЧЕСКИЕ отношения между фактами.

$\varphi = \text{“Если истинно } p \rightarrow q \text{ и ложно } q, \text{ то ложно и } p\text{”}$

В **модальной** логике можно уточнять, когда $\varphi = И$.

Утверждение φ истинно **для кого?**

Кто знает, что φ истинно?

В каких ситуациях φ истинно?

При каких условиях?

В какие моменты времени?

Иногда важно конкретизировать



ЧТО ТАКОЕ МОДАЛЬНОСТЬ?

- ЭТО *характеризация* истины

- Вася --- болен.

будет когда-то
 Врач знает, что
 в будущем всегда будет
 был когда-то
 всегда был
 имеет право быть
 я считаю, что

Обозначения:

$\square \varphi = \varphi$ - '*сильная*' характеристика истины φ (*необходимость*)

$\diamond \varphi = \varphi$ - '*слабая*' характеристика истины φ (*возможность*)

- В модальной логике 'сильные' и 'слабые' характеристики истины обозначаются одинаково:

'сильная' характеристика истины

\square "Вася болен"

'слабая' характеристика истины

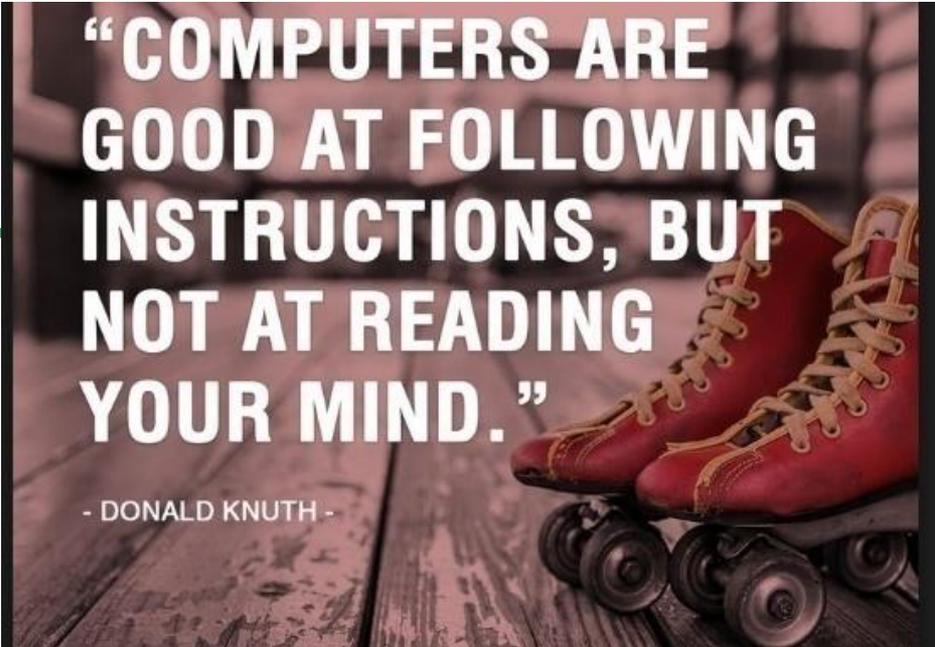
\diamond "Вася болен"



В эпоху ИИ такие “оценки” истин можно учитывать и со стороны искусственных активных объектов – агентов:

- **агенты изменяют** свое состояние на основании своих знаний выполняя действия; такие изменения происходят *во времени*;
- **агенты могут “знать”** или *“не знать”* факты; действуя в коллективе с другими агентами они должны использовать свои знания того, что могут знать другие агенты;
- **агенты должны** определять *“запрещенность”*, *“разрешенность”* или *“обязательность”* выполнения действий в рамках норм, законов и предписаний.

Модальная логика – формальный базис
искусственного интеллекта и МАС



“COMPUTERS ARE
GOOD AT FOLLOWING
INSTRUCTIONS, BUT
NOT AT READING
YOUR MIND.”

- DONALD KNUTH -

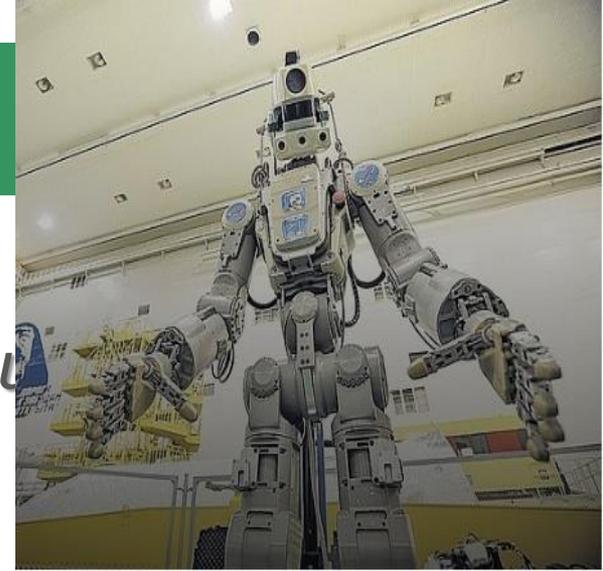
Дональд Кнут:

ЭТО НЕ случайность!

Это проявления глубокой концептуальной
проблемы программирования:
невозможно разработать сложную
программу без ошибок.



ИТАК: ОШИБКИ ПРИ СОЗДАНИИ ПО
НЕИЗБЕЖНЫ!?



*Программы, которые мы строим,
ведут себя в точности так, как мы их
запрограммировали!*

***НО не обязательно так, как мы
предполагали!***

Человеческий мозг не в состоянии справиться со сложностью при разработке современных программ (управления) в десятки миллионов строк кода

**Человек уже утратил контроль над
разрабатываемыми им сложными системами**

"It is an incredible privilege to be a software developer. You know, how cool is that?"

At the same time, it's a deep responsibility, because what we do does change the world".



Grady Booch

*... то, что мы делаем, **ДЕЙСТВИТЕЛЬНО** меняет мир*

Мы морально ответственны за те системы, которые мы разрабатываем и выпускаем в мир!

Сможем ли мы, специалисты в области компьютерных наук, *"творить"* новый мир таким, чтобы он был понятным, предсказуемым, комфортным, **БЕЗОПАСНЫМ??**

