Верификация параллельных программных и аппаратных систем

Курс лекций

Карпов Юрий Глебович
профессор, д.т.н., зав.кафедрой
“Распределенные вычисления и компьютерные сети”
Санкт-Петербургского политехнического университета

karpov@dcn.infos.ru

Лекция 4

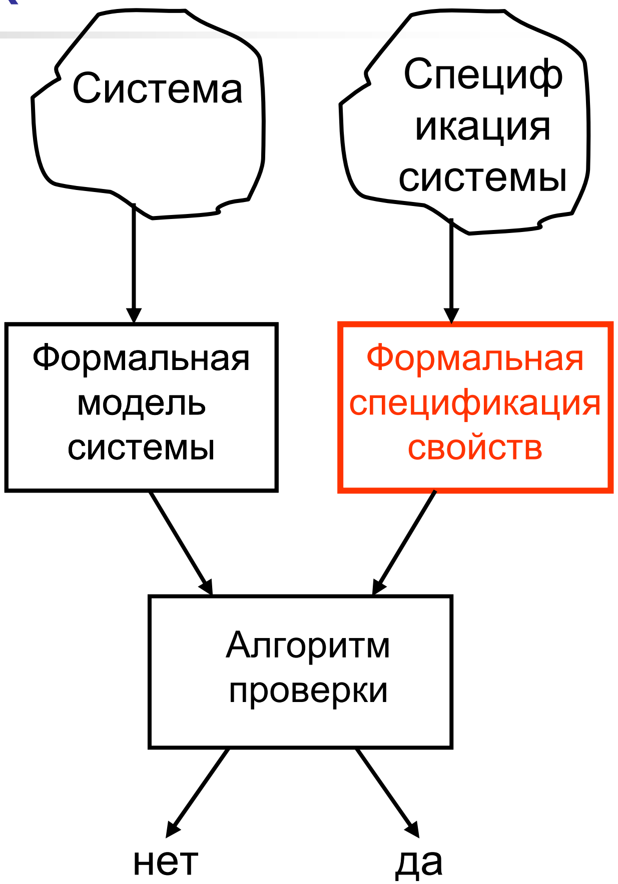
Темпоральные логики

План курса

1. Введение
2. Метод Флойда-Хоара доказательства корректности программ
3. Исчисление взаимодействующих систем (CCS) Р.Милнера
4. Темпоральные логики
5. Алгоритм model checking для проверки формул CTL
6. Автоматный подход к проверке выполнения формул LTL
7. Структура Крипке как модель реагирующих систем
8. Темпоральные свойства систем
9. Система верификации Spin и язык Promela. Примеры верификации
10. Применения метода верификации model checking
11. BDD и их применение
12. Символьная проверка моделей
13. Количественный анализ дискретных систем при их верификации
14. Верификация систем реального времени (I)
15. Верификация систем реального времени (II)
16. Консультации по курсовой работе

Что было на предыдущих лекциях

- Мотивация – большая цена ошибок в ПО
- Примеры ошибок
- Необходимость верификации
- Верификация и тестирование
- Прорыв в области верификации, основанный на изящных формальных моделях
- Успехи верификации
- Общая схема верификации
- Метод Флойда-Хоара индуктивной верификации программ обработки данных



Мы начинаем изучение другого метода – model checking

Цель данной лекции – рассмотреть формальный язык, на котором можно специфицировать свойства поведения дискретных систем (требования)

Это язык логики – но не обычной логики

Ограниченность классической логики

- Классическая логика
  - Примитивная модель истины: "черно-белая" модель, не существует степени уверенности-неуверенности, высказывания статичны, неизменны во времени $\Rightarrow$ неадекватна для высказываний о времени
  - Пример - (некоммутативность конъюнкции, A&B $\neq$ B&A):
    - "Джону стало страшно и он убил" $\Leftarrow ? \Rightarrow$ "Джон убил и ему стало страшно"
    - "Джон умер и его похоронили" $\Leftarrow ? \Rightarrow$ "Джона похоронили и он умер"
    - "Джейн вышла замуж и родила ребенка" $\Leftarrow ? \Rightarrow$ "Джейн родила ребенка и вышла замуж"
- В обычной логике высказываний не формализуются:
  - *Путин – наш президент* (истинно только в какой-то период)
  - *Мы не друзья, пока ты не извинишься*
  - *Если **m** поступит на вход в канал, то потом **m** появится на выходе*
  - *Каждый запрос к лифту с произвольного этажа, поступивший в любой момент времени, будет когда-нибудь в будущем удовлетворен*

Элементарные (атомарные) утверждения в общем случае истинны в один момент времени и ложны в другой!
Многие утверждения естественного языка нельзя выразить в обычной логике, в которой нет понятия времени

Темпоральная логика

- Определение
  - TL - это любая логическая система, которая позволяет формализовать утверждения, истинность которых изменяется со временем (не вводя явно понятие времени)
- Применения TL (используются РАЗНЫЕ TL!!)
  - ФИЛОСОФИЯ: формализм для прояснения философских вопросов о времени;
  - ЕСТЕСТВЕННЫЙ ЯЗЫК: формализм для определения семантики утверждений в естественных языках, включающих время;
  - ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ: язык для представления знаний, связанных со временем (Д. А. Поспелов (ред) "*Представление знаний о времени и пространстве в интеллектуальных системах*«, 1987);
  - ВЫЧИСЛИТЕЛЬНАЯ НАУКА: язык для выражения утверждений о временных свойствах выполнения программ
  - ТЕХНИКА: для формализации утверждений о свойствах ***поведения*** технических систем

Мы будем рассматривать TL с точки зрения верификации ПО и технических систем

Утверждения, зависящие от времени

Формализуем утверждение:

*Если сообщение **m** поступит на входе в канал, то когда-нибудь в будущем это сообщение появится на выходе*:

НаВходе(m) ⇒ *НаВыходе*(m)

Но эта формализация неадекватна! Второе утверждение истинно в тот же момент, когда истинно первое!

Смысл утверждения, который мы хотим формализовать:

Если на входе в произвольный момент времени t появилось сообщение m, то в некоторый следующий момент времени t1, такой, что t < t1, это же сообщение появится на выходе

В обычной логике *высказываний* все, относящееся ко времени, не может быть выражено

Как зависимость от времени ввести в логические утверждения?

Попытка формализации: Использование предикатов

*"Если сообщение **m** поступит на вход в канал, то когда-нибудь в будущем оно появится на выходе"*

$(\forall t \geq 0)\ [\mathit{НаВходе}(\boldsymbol{m},t) \Rightarrow (\exists t'>t)\ [\mathit{НаВыходе}(\boldsymbol{m},t')]\ ]$

"Лифт никогда не пройдет мимо этажа n, от которого поступил еще не удовлетворенный запрос"

Пусть P(t) – позиция лифта в момент t.

$(\forall t \geq 0)\ (\forall t'>t)\ [\mathit{Запрос}(n,t)\ \&\ P(t')\neq n\ \&\ (\exists\ t1: t\leq t1\leq t')P(t1)=n \Rightarrow$
$(\exists\ t_{об}: t\leq t_{об}\leq t')\ \mathit{Обслужен}(n,t_{об})]$

"Мы не друзья, пока ты не извинишься"

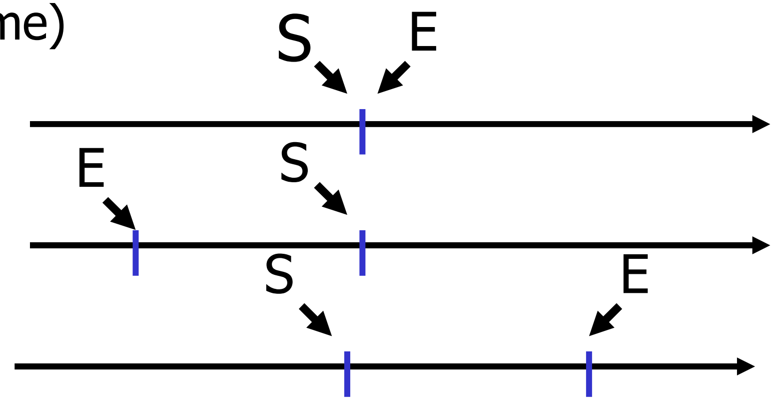
$(\forall\ t > 0)[\ (\ (\forall t_1: 0<t_1< t)\ \neg \mathit{Извиняешься}(\mathit{ты}, t_1)\ ) \Rightarrow \neg\ \mathit{Друзья}(я, ты, t)\ ]$

В предикатной логике громоздкая нотация, тяжелый формальный аппарат

Попытка формализации:
Темпоральный анализ естественного языка

Как формализовать глагольные времена??

Введем два момента времени:
- S – момент разговора (Speech Time)
- E – момент наступления события (Event time)



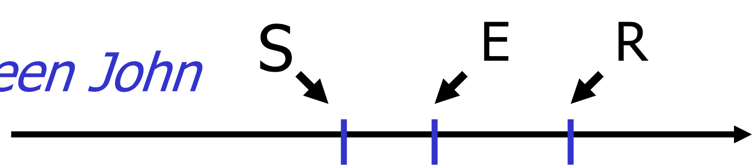
Можно формализовать три времени:

настоящее	(S=E)	Я вижу Джона
прошедшее	(E<S)	Я видел Джона
будущее	(S<E)	Я увижу Джона

Как формализовать Future Perfect: *I will have seen John*??

Райхенбах ввел еще один момент: точку референции R – время, на которое ссылаемся (Reference time)

I will have seen John



(*) H. Reichenbach. *Elements of Symbolic Logic*, 1947

Темпоральный анализ естественного языка

Чем различаются Simple Past и Present Perfect?

"I saw John”

E=R<S

В этот момент я увидел Джона

Речь идет об этом (прошлом) моменте времени

Я говорю сейчас

"I have seen John"

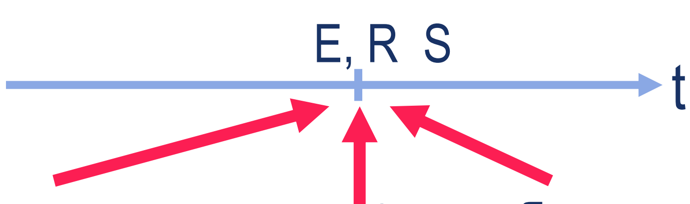
E<R=S

В этот момент я увидел Джона

Речь идет об этом (текущем) моменте времени

Я говорю сейчас

E=R=S



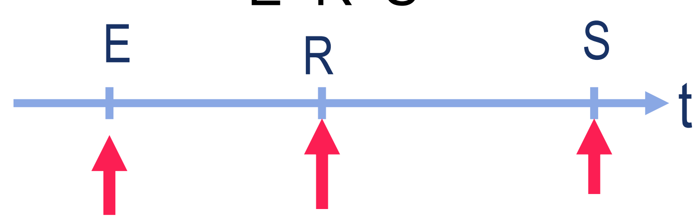
В этот момент я увидел Джона

Речь идет об этом моменте времени

Я говорю сейчас

“I see John”

E<R<S



В этот момент я увидел Джона

Речь идет об этом моменте времени

Я говорю сейчас

“I had seen John”

Попытка формализации: Введение модальностей

- Определение Модальной логики
  - Модальность (от лат. *modus* – вид, способ, наклонение) – это категория, определяющая отношение высказывания к действительности
  - Модальная логика - любая формальная логическая система, в которой присутствуют модальные операторы
- Примеры модальных операторов:
  - M - “*возможно, что*” (Mp – “*возможно, что* p”)
  - L - “*необходимо, что*” (Lq – “*q обязательно выполняется*”)
  - F – “*когда-нибудь в будущем будет верно, что ...*”
  - G – “всегда *в будущем будет верно, что ...*”
  - P - “*когда-то в прошлом было верно, что ...*”
- Можно определить и соотношения между модальностями:
  - $Lp \equiv \neg M \neg p$ $\qquad$ $Fp \equiv \neg G \neg p$

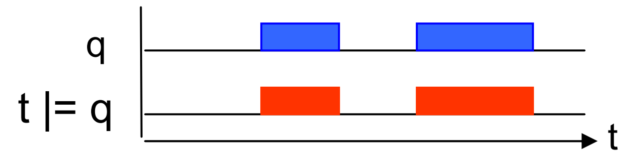
Примеры: $Mp \neq \neg M \neg p$ *Может писать ≠ не может не писать* !!!

$Lp \equiv \neg M \neg p$: *Писатель должен писать тогда и только тогда, когда он не может не писать*

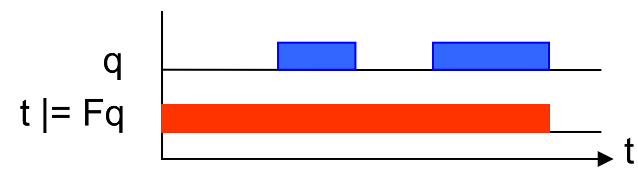
Tense Logic

Впервые - философ Diodorus Cronus. В 20 веке – Arthur Prior

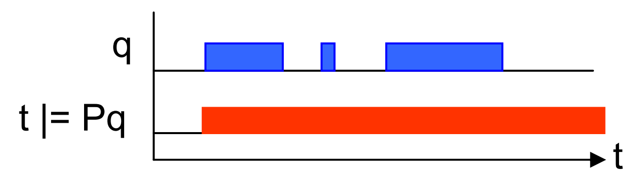
q – q выполняется *сейчас*, в момент t:



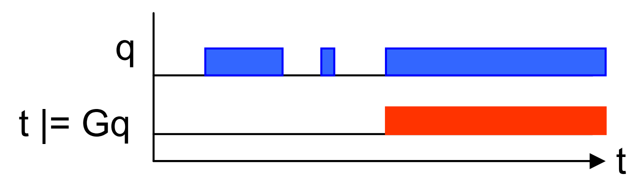
Fq – q случится в будущем:



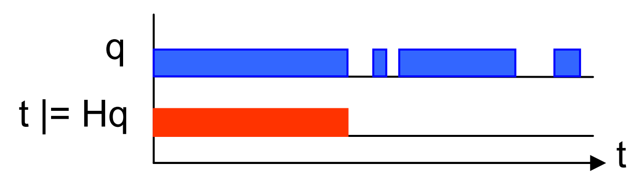
Pq – q случилось в прошлом:



Gq – q всегда будет в будущем:



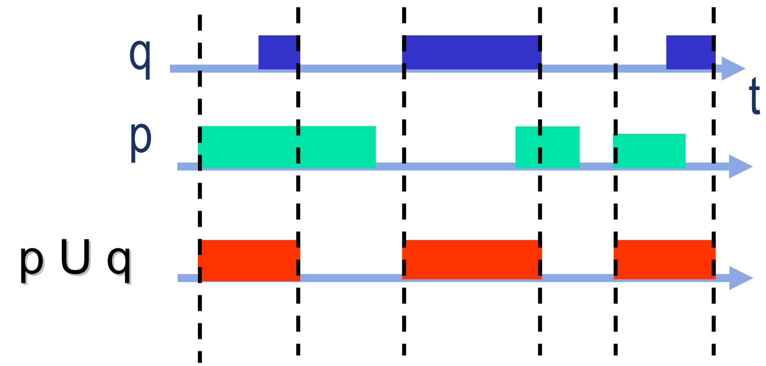
Hq – q всегда было в прошлом:



Дополнительные модальности TL: Until, X

U (Until)

pUq истинно в момент t, если
$(\exists t' \geq t)\ t' \models q\ \&\ (\forall\ t_1: t \leq t_1 < t')\ t_1 \models p$



X (Next time)

Xp истинно в момент t, если p истинно в следующий момент времени
(если считать моменты времени дискретными, то в момент t+1)

Итак, в Tense Logic четыре модальности: F, G, U, X

F и G выражаются через U:
$Fp \equiv true\ U\ p$,
$Gp \equiv \neg F \neg p$

Примеры формализаций высказываний

- *Джейн вышла замуж и родила ребенка*
  $P(\text{Джейн\_выходит\_замуж} \wedge F\ \text{Джейн\_рожает\_ребенка})$
- *Джейн родила ребенка и вышла замуж*
  $P(\text{Джейн\_рожает\_ребенка} \wedge F\ \text{Джейн\_выходит\_замуж})$
- *Джон умер и его похоронили*
  $P(\text{Джон\_умирает} \wedge XF\ \text{Джона\_хоронят})$
- *Если я видел ее раньше, то я ее узнаю при встрече*
  $G(P\ \text{Увидел} \Rightarrow G(\text{Встретил} \Rightarrow X\ \text{Узнал}))$
- *Ленин – жил, Ленин – жив, Ленин – будет жить* (В.В.Маяковский)
  $PG\ \text{Ленин\_жив}$
- *Любое посланное сообщение будет получено*
  $G(\text{Послано}(m) \Rightarrow F\ \text{Получено}(m))$
- *Вчера он сказал, что придет завтра, значит, он придет сегодня*
  $X^{-1}X\ \text{Приходит} \Rightarrow \text{Приходит}$ (истинно)

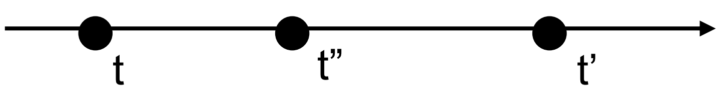
Логика предикатов и Tense Logic

*Если сообщение **m** поступило на вход в канал, то когда-нибудь в будущем **m** появится на выходе*

$(\forall t \geq 0)\ [\textit{НаВходе}(\boldsymbol{m},t) \Rightarrow (\exists t'>t)\ [\textit{НаВыходе}(\boldsymbol{m},t')]\ ]$

$\mathbf{G}[\ \textit{НаВходе}(\ m\ ) \Rightarrow \mathbf{XF}\,\textit{НаВыходе}(\ m\ )\ ]$

Лифт никогда не пройдет мимо этажа n, от которого поступил еще не удовлетворенный запрос



Пусть P(t,n): – позиция лифта в момент t равна n

$(\forall t \geq 0)\ \textit{Запрос}\ (t,n)\ \&\ (\exists\ t': t' > t)\ (\forall\ t': t \leq t'' < t')\ \neg \textit{Обслужен}(t'', n) \Rightarrow \neg P(t'', n)$

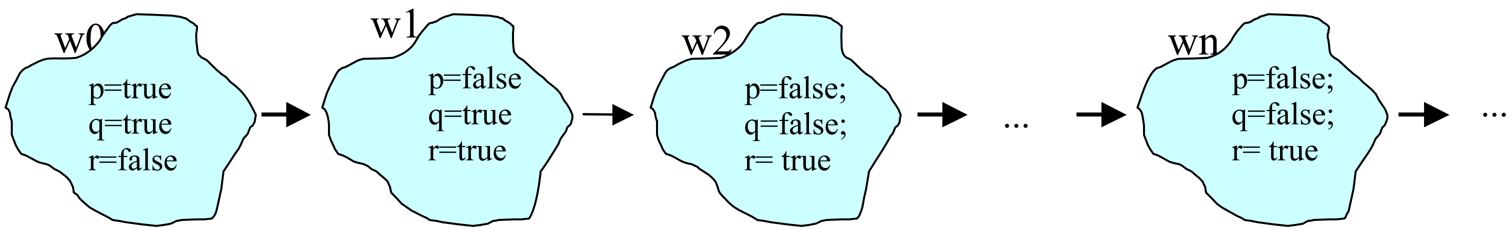
$G\ [\textit{Запрос}(\ n\ ) \Rightarrow \neg\ P(n)\ U\ \textit{Обслужен}(\ n\ )\ \ ]$

Спецификация свойств в TL – ясная, четкая, компактная

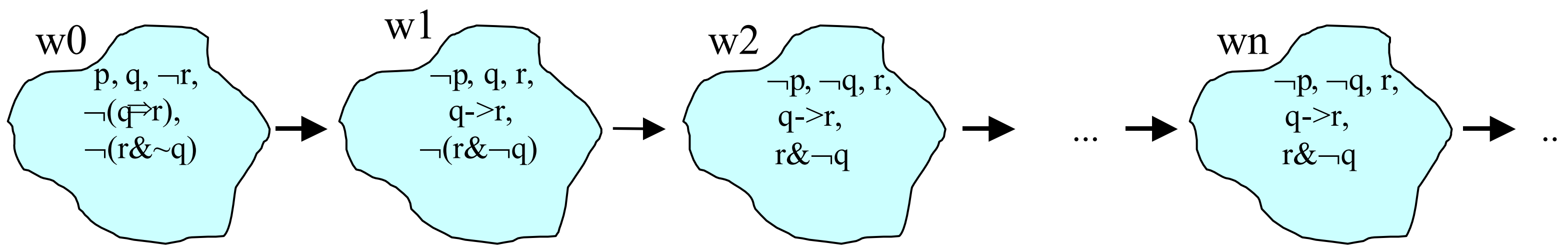
LTL в дискретном времени

Амир Пнуэли (1977)

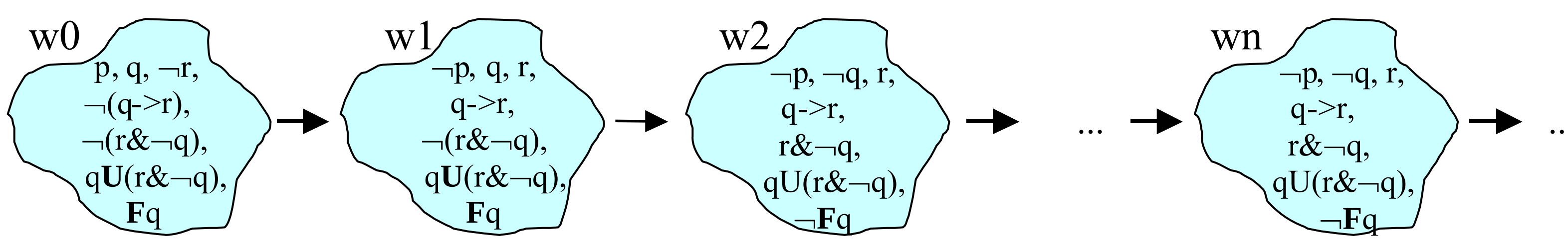
Последовательность “миров”, в каждом свое понимание истинности:



В каждом мире произвольная логическая формула истинна, либо нет:



Это же справедливо и для произвольной темпоральной формулы:



На цепочке миров как на целом объекте выполняются формулы p, q,¬r,¬(r&¬q), q**U**(r&¬q), Fq, ... *потому что они истинны в w0*

Формализация высказываний в TL

- *"Dum spiro, spero"* - *пока живу – надеюсь*

  G*(я_живу ⇒ я_надеюсь)*
- *"Мы придем к победе коммунистического труда!"*

  F *коммунистический_труд_победил!*
- *"Сегодня он играет джаз, а завтра Родину продаст!"*

  он_играет_джаз ⇒ X*он_продает_Родину* – слишком буквально

  G(*он_играет_джаз ⇒* FX*он_продает_Родину*)
- Пусть p = "*я люблю Машу*", q ="*я люблю Дашу*"

  Fp – *"я когда-нибудь обязательно полюблю Машу"*

  qUp – *"я полюблю Машу, а до этого буду любить Дашу"*

  FGp – *"когда-нибудь в будущем я полюблю Машу навечно"*

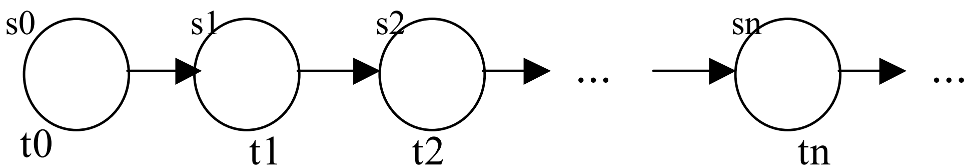
  GFq– *"я буду бесконечно влюбляться в Дашу "*
- "*Раз Персил – всегда Персил*"

  G(*Персил ⇒* G*Персил*) – *раз попробовав, будешь всегда*

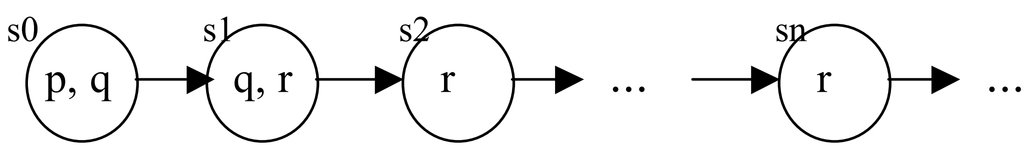
  (по английски G(*Φ ⇒* G*Φ*) *once Φ, always Φ)*

LTL и анализ дискретных технических систем

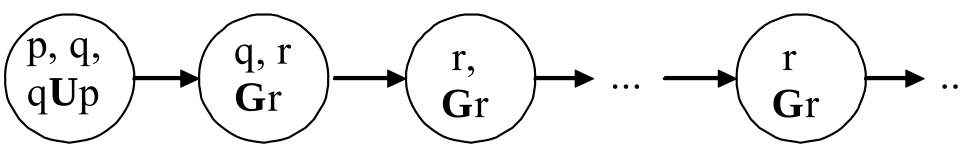
Последовательность "*миров*" в TL можно трактовать как бесконечную последовательность состояний дискретной системы, а отношение достижимости – как дискретные переходы системы:



Атомарные формулы - базисные свойства процесса в состояниях:



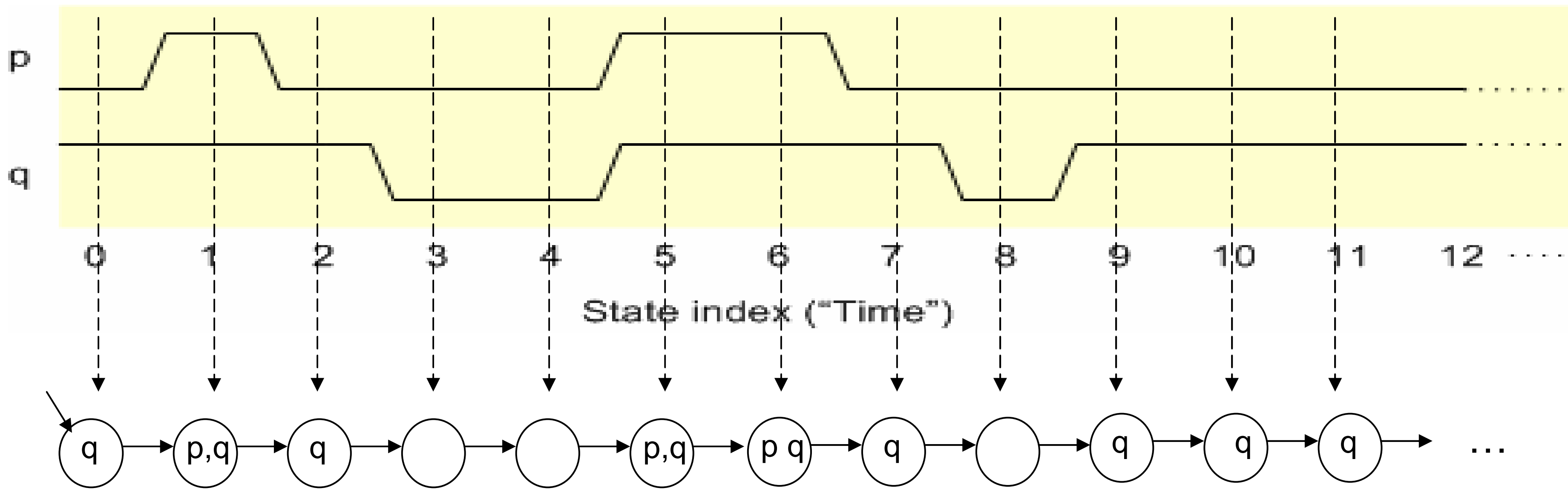
Производные темпоральные формулы в состояниях – это свойства динамического процесса, характеризующие вычисление в будущем:



qUp – выполняется на вычислении

Gr – не выполняется на вычислении

Спецификация свойств логических схем



$\sigma_0 \models \neg p$

$\sigma_0 \models \mathrm{F}\ \neg q$

$\sigma_0 \models \mathrm{G}(p \rightarrow q)$

$\sigma_0 \models \mathrm{q\ U\ p}$

$\sigma_0 \models \mathrm{FG}(\neg \mathrm{p} \wedge \mathrm{q})$

$\sigma_0 \models \neg p$

$\sigma_0 \models \Diamond \neg q$

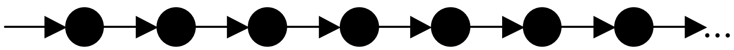
$\sigma_0 \models \Box(p \rightarrow q)$

$\sigma_0 \models q\ \mathrm{U}\ p$

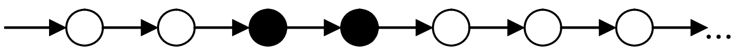
$\sigma_0 \models \Diamond \Box(\neg \mathrm{p} \wedge \mathrm{q})$

Примеры формул LTL для дискретных систем

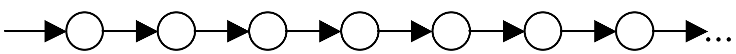
G q - *всегда в будущем*



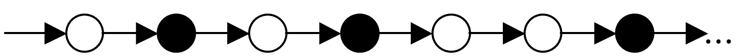
F q - *хотя бы раз в будущем*



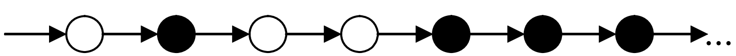
¬F q – *никогда в будущем*



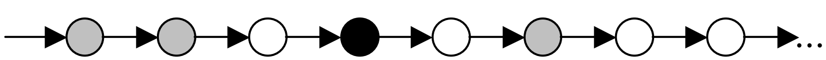
GFq – *бесконечно много раз в будущем*



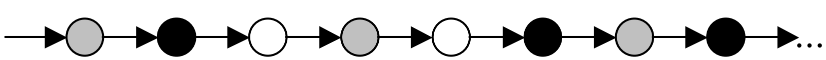
FGq – *с какого-то момента постоянно*



p⇒Fq – *на p в s_0 будет реакция q когда-нибудь в будущем*



G[p⇒Fq] – *всегда на p будет реакция q*



Linear Temporal Logic (LTL)

- *Язык формальной логики* имеет:
  синтаксис (правила построения формул) и
  семантику (правила, определяющие истинностное значение формул)

- Определение синтаксиса LTL задается грамматикой:
  - Формула ϕ PLTL*:* это :
    - атомарное утверждение p, q, ...,
    - или Формулы, связанные логическими операциями $\vee$, $\neg$
    - или Формулы, связанные темпоральными операторами **U**, **X**

$$\text{Грамматика: } \phi ::= p \mid \phi_1 \vee \phi_2 \mid \neg\phi \mid \mathbf{X}\phi \mid \phi_1 \ \mathbf{U}\ \phi_2$$

Другие (выводимые) темпоральные операторы :

$$\mathbf{F}p \equiv \text{true } \mathbf{U}\ p$$
$$\mathbf{G}p \equiv \neg\mathbf{F}\ \neg p$$

Прошлое при анализе технических систем менее важно

Семантика операторов LTL

Обозначения:

$\sigma = s_0 s_1 s_2 s_3 \ldots$; $\sigma_i \models \varphi \equiv$ в s_i вычисления σ истинно φ

Базовые операторы $\vee, \neg$, U, X

$\sigma_i \models p$ iff в состоянии s_i истинно p

$\sigma_i \models \neg\phi$ iff $\sigma_i \not\models \phi$

$\sigma_i \models \phi \vee \varphi$ iff $\sigma_i \models \phi$ или $\sigma_i \models \varphi$

$\sigma_i \models X\,\phi$ iff $\sigma_{i+1} \models \phi$

$\sigma_i \models \phi\, U\, \varphi$ iff $(\exists j \geq i)\ \sigma_j \models \varphi$ и $(\forall\ k: i \leq k < j)\ \sigma_k \models \phi$

Выводимые операторы Fp, Gp

$\sigma_i \models G\,\varphi$ iff $(\forall j \geq i)\ \sigma_j \models \varphi$

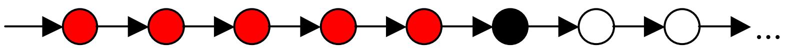
$\sigma_i \models F\,\varphi$ iff $(\exists j \geq i)\ \sigma_j \models \varphi$

Естественно определить истинность темпоральной формулы относительно начального состояния вычисления σ, т.е.
Φ выполняется на вычислении σ iff $\sigma_0 \models \Phi$

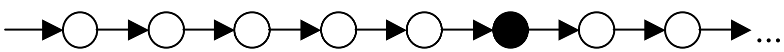
Связь между операторами LTL

Определение операторов LTL через neXtTime

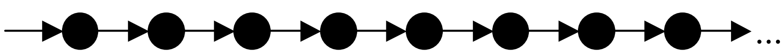
- $pUq \equiv q \vee p \wedge Xq \vee p \wedge Xp \wedge XXq \vee \ldots$



- $Fq \equiv q \vee Xq \vee XXq \vee \ldots$



- $Gq \equiv q \wedge Xq \wedge XXq \wedge \ldots$



Рекурсивное определение операторов LTL

- $pUq \equiv q \vee p \wedge X(pUq)$
- $Fq \equiv q \vee XFq$
- $Gq \equiv q \wedge XGq$

Пример задачи

- Свойства оператора X

  Для любого ψ, $X^k X^r\psi = X^{k+r}\psi$

- **Задача.** Если сегодня понедельник, какой день будет после дня, который будет перед днем, который будет перед завтрашним днем?

  сегодня = понедельник

$$\underbrace{X\,(\underbrace{X^{-1}(\underbrace{X^{-1}(\underbrace{X\ \text{сегодня}}_{\text{завтра}})}_{\text{перед завтра}})}_{\text{перед ним}})}_{\text{после него}} = X^0\text{сегодня} = \text{сегодня} = \text{понедельник}$$

Линейное и ветвящееся время

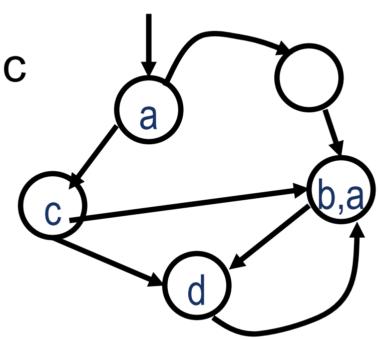
По заданной бесконечной цепочке состояний с определенным в каждом состоянии набором истинных AP нужно вычислить значение булевых и темпоральных формул. Как вычисления представить конечным образом?

Мы живем в линейном мире, в LTL формализован взгляд на время, как на линейную последовательность (дискретных) возрастающих значений. Но поведения информационных систем имеют альтернативы

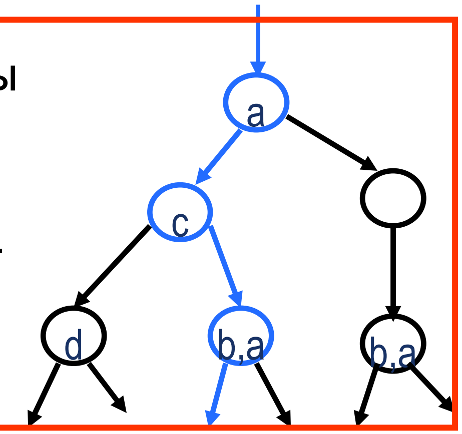
Для формализации этого введена структура Крипке

Структура Крипке – это модель, представляющая конечным образом бесконечные цепочки состояний с наборами атомарных утверждений и с альтернативным выбором – фактически, с ветвящимся временем

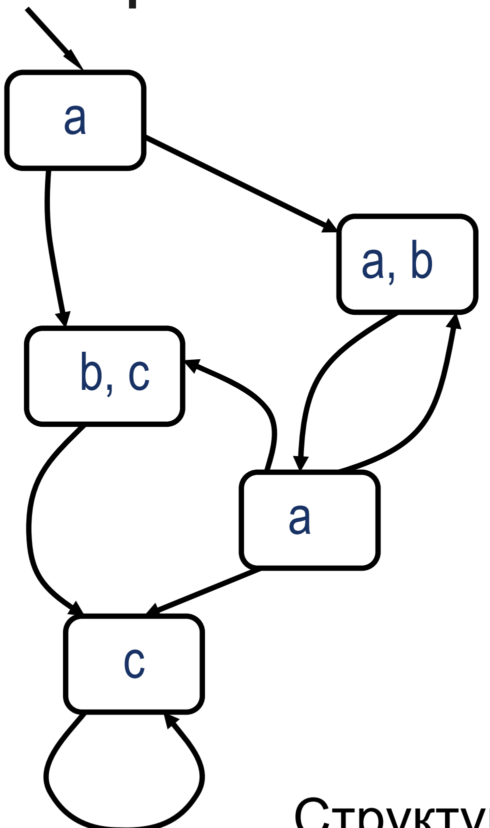
Структура Крипке – система переходов с помеченными состояниями и непомеченными переходами



Развертка структуры Крипке определяет бесконечные цепочки состояний - возможные ВЫЧИСЛЕНИЯ



Структура Крипке – интерпретация формул TL



Структура Крипке – это конечный автомат с непомеченными переходами, с каждым состоянием которого связано некоторое множество простых утверждений, истинных в этом состоянии

- Формально: $M = (S, S_0, R, L)$, где:
  - S – конечное множество состояний
  - S_0 - множество начальных состояний
  - $R \subseteq S\text{x}S$ – множество переходов; $(\forall s)(\exists s')\text{: } (s,s') \in R$
  - AP – множество атомных утверждений
  - $L\text{: } S \to 2^{AP}$ – функция пометок
- Путь в M – любая бесконечная цепочка $s_0 s_1 s_2 s_3 \ldots$

Структуру Крипке можно считать расширением КА, в котором существенны только возможные последовательности смены состояний при произвольных входах (вычисления)

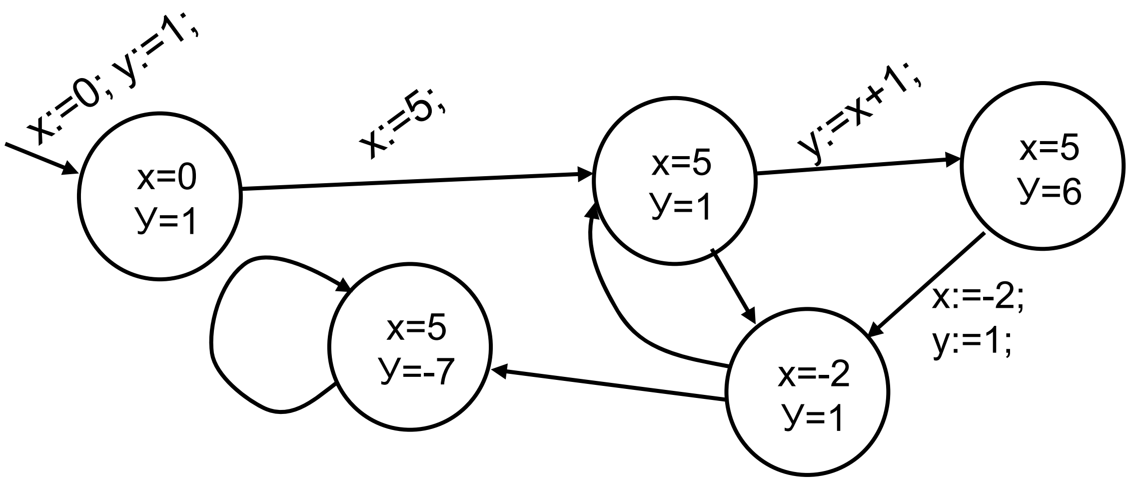
S. A.Kripke. *Semantical consideration on modal logic*//Acta Philosophica Fennica, v.16, 1963

Структура Крипке для программ

```
begin
x:=0; y:=1;
while x+z < 5 do
{ x:=5;
if z=1 then y:=x+1;
x:= -2; y:=1;
}
y:= x*y-5; x:=5;
end
```

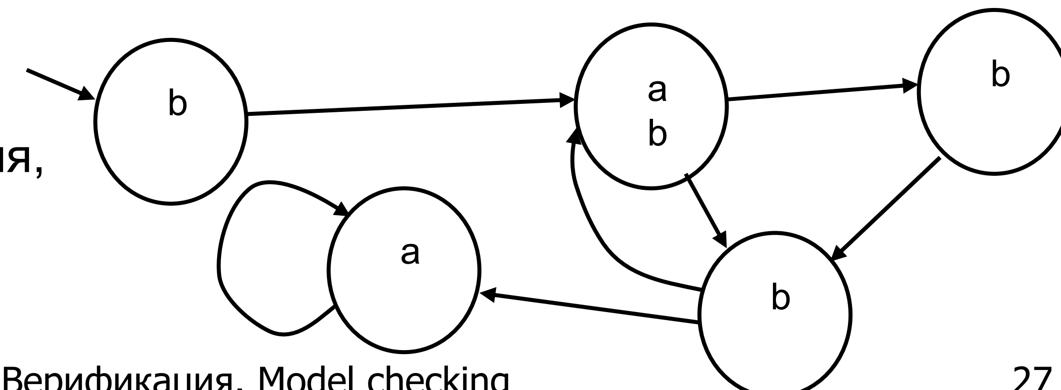
Состояние программы – вектор значений ее переменных И метки (pc)

Переходы – изменение переменных программы операторами И/ИЛИ только pc:



Пусть атомарные утверждения, *ИНТЕРЕСУЮЩИЕ НАС:*

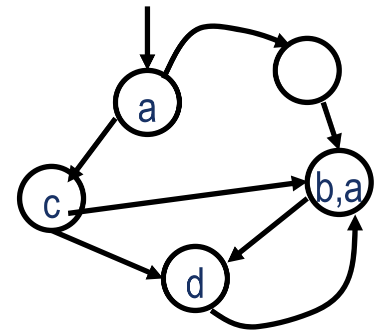
a=x>y; b=|x+y|<3



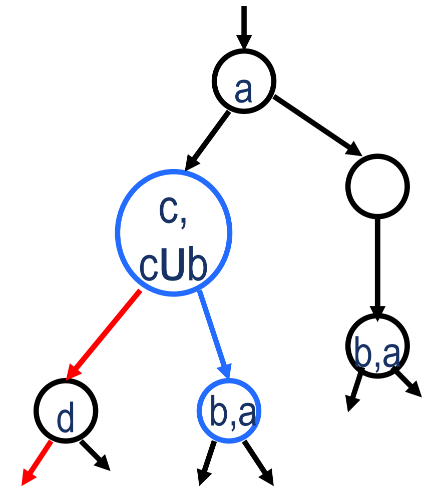
Как идеи TL применить к ветвящемуся времени?

Каждое состояние может иметь не одну, а множество цепочек – продолжений и является корнем своего дерева историй (вычислений)

Но как понимать формулы LTL: **FG**p, p**U**q, … в состоянии s?



Общий метод – ввести квантор "*пути*" (path quantifier)



E $\phi \equiv$ "*существует такой путь из данного состояния, на котором LTL формула* ϕ *истинна*"

A $\phi \equiv$ "*для всех путей из данного состояния LTL формула* ϕ *истинна*"

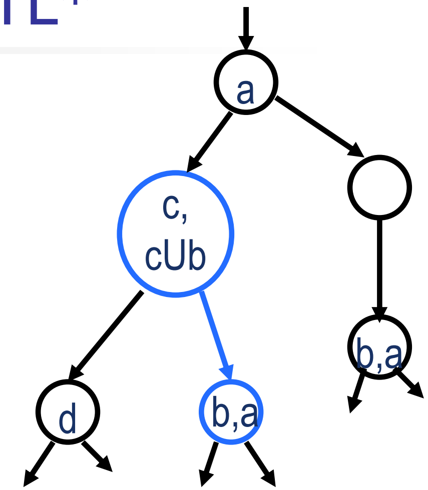
Очевидно, A $\phi \equiv \neg$E $\neg\phi$

Формулы TL можно разделить на два класса:

- *ф-лы состояний* – характеризуют одно состояние
- *ф-лы пути* - характеризуют какой-то путь

(Надо обязательно дополнительно указать, какой это путь!)

Общая логика ветвящегося времени – CTL*



Темпоральные логики ветвящегося времени рассматривают возможные вычисления (пути на дереве) - траектории на развертке структуры Крипке

CTL* – Computational Tree Logic* - это одна из возможных логик ветвящегося времени

Грамматика. Формула CTL* - это формула состояний ϕ:

- Формулы состояний $\phi ::= p \mid \neg \phi \mid \phi \vee \phi \mid E\, \alpha \mid A\alpha$
- Формулы путей $\alpha ::= \phi \mid \neg \alpha \mid \alpha \vee \alpha \mid \alpha U \alpha \mid X\alpha$

формула ϕ состояния s является формулой пути σ, если это состояние s является начальным состоянием пути σ

Формула пути имеет смысл только если зафиксирован путь!

В состояниях могут стоять только state formula!

Язык формул темпоральной логики CTL*

Возможные формулы CTL* : A [(pUr)∨(qUr)], A [Xp∨XXr], EGFp

Операции логики высказываний	Четыре темпоральных оператора	Два квантора пути
1. ¬ – Отрицание	1. X – “neXt time”	1. E – “Exists”
2. ∨ – Дизъюнкция	2. U – “Until”	2. A – “Always”
3. ∧ – Конъюнкция	3. F – “in the Future”	
4. ⇒ – Импликация	4. G – “Globally”	
…		

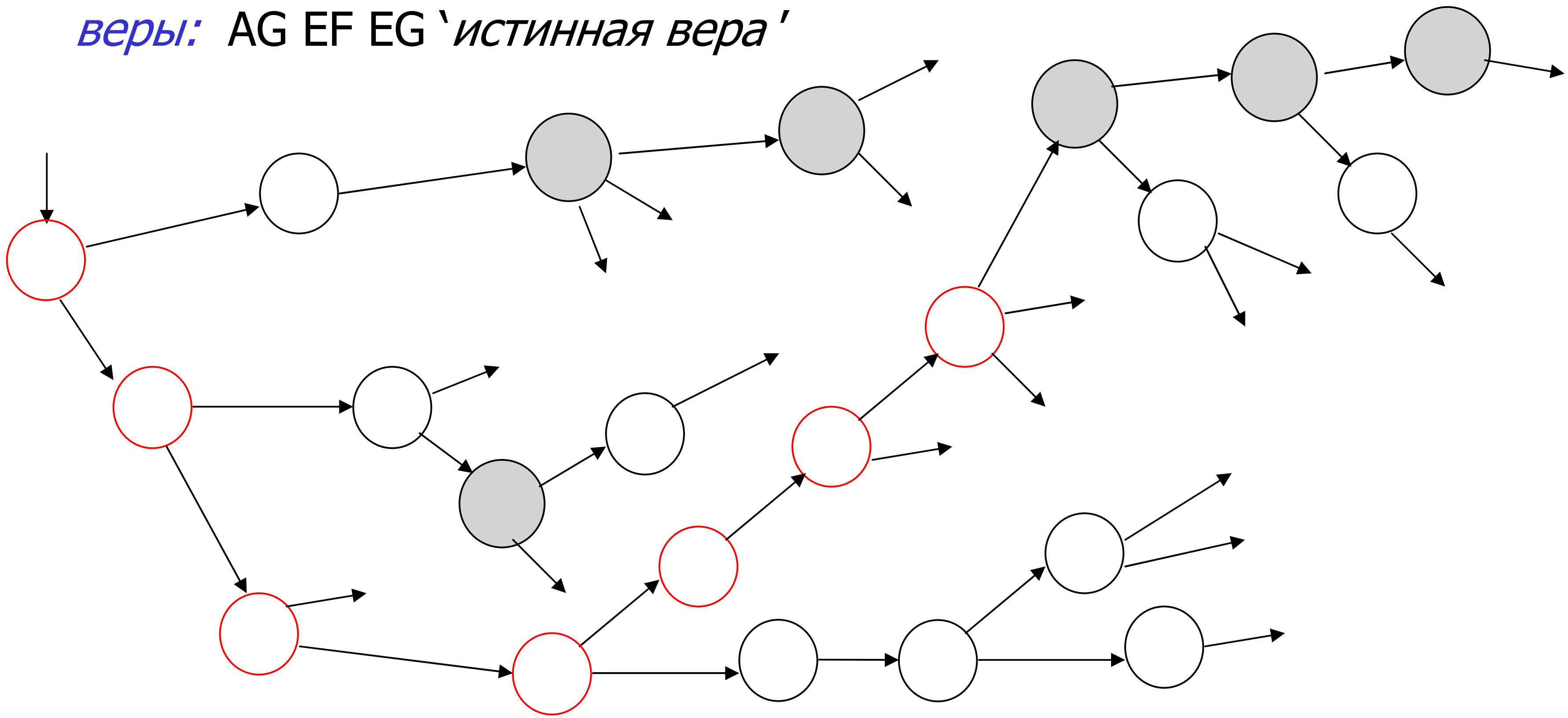
Базис CTL* = {¬, ∨, U, X, E}

Примеры

- Обозначение: p = “*Я люблю Машу*”
- AGp:
  - “*Я люблю Машу, и, что бы ни случилось, я буду любить ее всегда”*
- AFGp:
  - “Ч*то бы ни случилось, я в будущем полюблю Машу навсегда”*
- EFp:
  - *“Я не исключаю такого развития событий, что в будущем я полюблю Машу”*

Примеры

- *Любой грешник всегда имеет шанс вернуться на путь истинной веры:* AG EF EG *`истинная вера'*



В любом состоянии вашей жизни (AG) существует такой путь (E), что на нем в конце концов (F) попадем в состояние, с которого идет "истинный путь" (EG)

LTL и CTL – подклассы CTL*

В LTL - формулы пути, которые должны выполняться для всех вычислений, т.е. предваряются квантором пути A

В CTL каждый темпоральный оператор предваряется квантором пути A или E

Формулы LTL:

AG(p ⇒**F** q)

A (¬a∨ **G**b & (a**U** ¬c))

A (a **U** ¬b)

Формулы CTL:

AG(p&¬**EF**(q⇒r))

EF(a & **E**(a**U** ¬c))

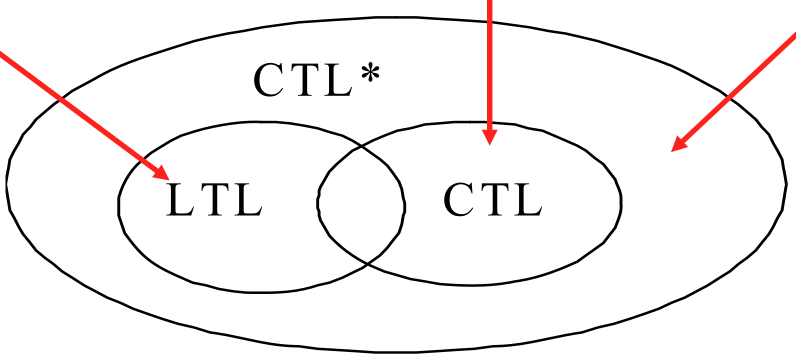
A (a **U** ¬ b)

Формулы CTL*:

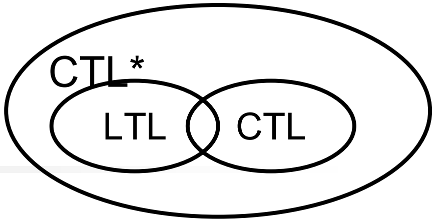
E(¬p & **X A F** q)

EX (a & **AX**(b**U**c)]

A (a **U** ¬ (**F** b))



Сравнение логик LTL и CTL

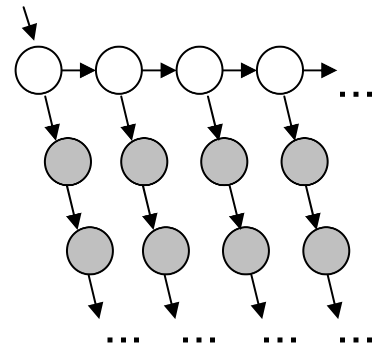


- Формулы этих двух логик характеризуют свойства разных объектов
  - LTL – формулы пути, CTL – формулы состояний
- Выражают свойства вычислений, которые представлены по-разному
  - LTL – множество поведений, CTL – деревья поведений
- Интерпретируются по-разному
  - формулы LTL - на бесконечном множестве поведений
  - формулы CTL – на конечном множестве состояний
- Методы анализа - алгоритмы model checking - совершенно разные
- Выразительная мощь несравнима
  - есть формулы CTL, невыразимые в LTL, и наоборот

Пример формализации в CTL*

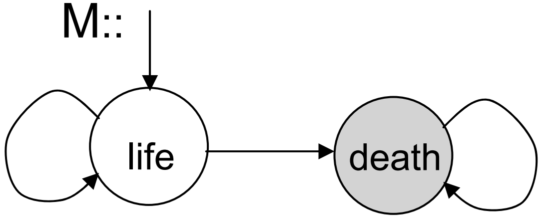
“Летят за днями дни, и каждый час уносит
Частичку бытия, а мы с тобой вдвоем
Предполагаем жить, и глядь — как раз - умрем”

А.С.Пушкин



φ_{br} = **A** [(**G** *life*) $\Rightarrow$ (**GEX** *death*)]

M::

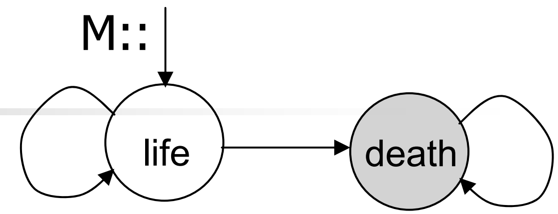


M |= φ_{br}

На любой истории (**A**) с вечной жизнью (**G** *life*) всегда (**G**) возможно (**E**) помереть в следующий момент (**X** *death*)

Возможность переключения нашего бытия на другую ветвь, на которой нас ожидает смерть

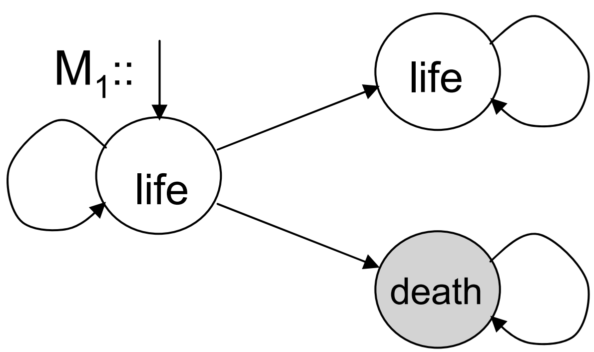
Другая формализация в CTL*



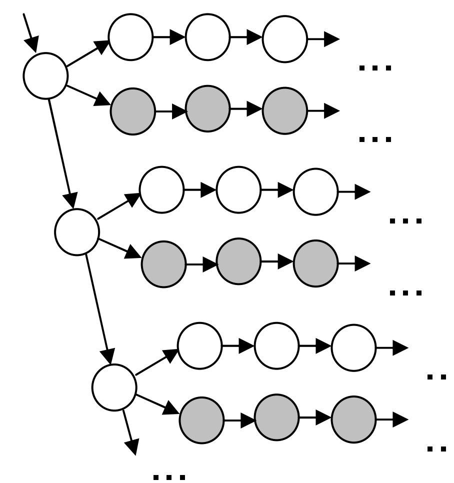
“Летят за днями дни, и каждый час уносит
Частичку бытия, а мы с тобой вдвоем
Предполагаем жить, и глядь — как раз - умрем”

А.С.Пушкин

$$\varphi_{br} = \mathbf{A}\ [(\ \mathbf{G}\ \mathit{life}) \Rightarrow (\mathbf{GEX}\ \mathit{death})]$$

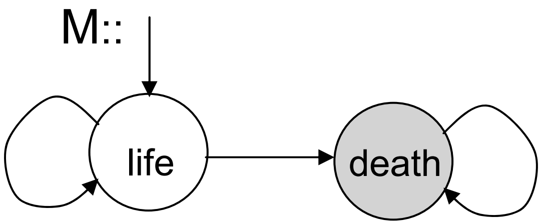


$$M_1 \not\models \varphi_{br}$$

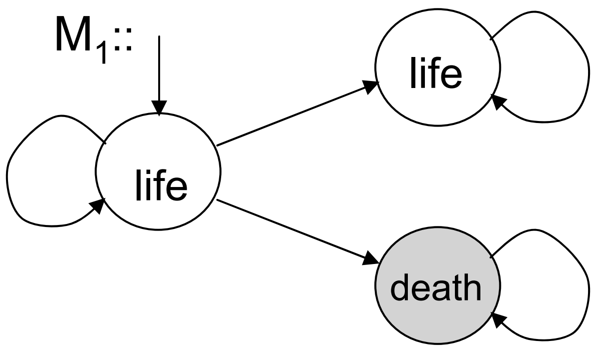


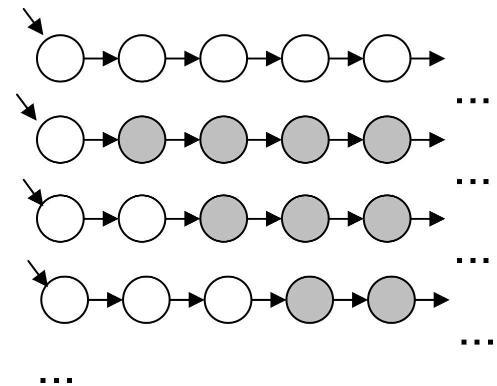
На структуре Крипке M1 формула φ_{br} не выполняется: ее развертка имеет траектории “вечной жизни” без возможного ответвления на состояния, помеченные *death*

LTL рассматривает только цепочки состояний (поведения)



Множества поведений у M и M_1 совпадают





$$\mathit{life}^{\omega} \cup \mathit{life}^{+}\,\mathit{death}^{\omega}$$

$M \models \varphi_{br}$ $\quad$ $M_1 \not\models \varphi_{br}$

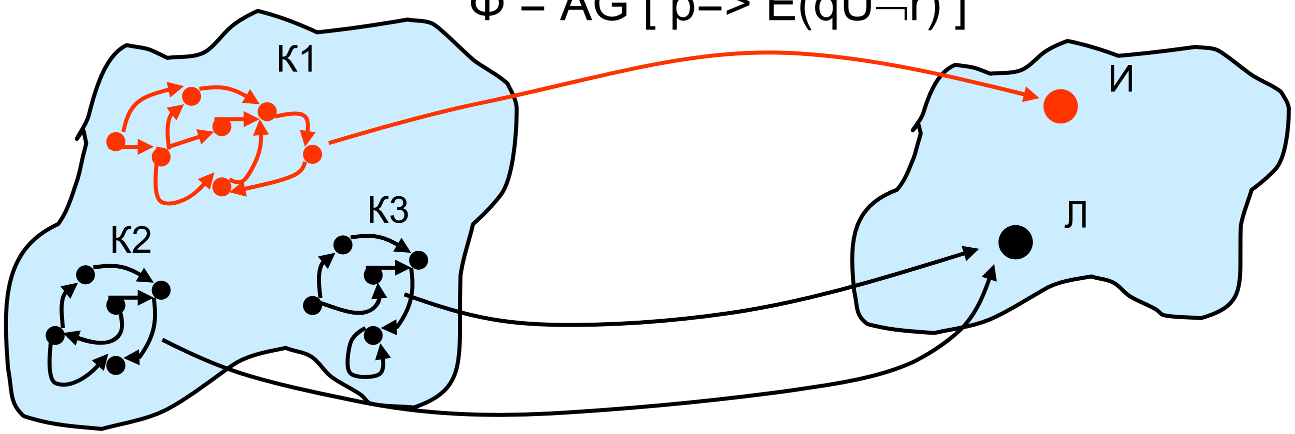
Следствие. Ни одна LTL формула не может различить структуры Крипке M и M_1, а CTL* - может.

=> CTL* мощнее LTL

Задачи верификации методом Model checking

Алгоритм Model Checking – это алгоритм проверки того, выполняется ли произвольная формула темпоральной логики Φ на произвольной структуре Крипке, модели технической системы

$\Phi = AG [\, p \Rightarrow E(q U \neg r) \,]$



Интерпретации Φ – это структуры Крипке, в каждом состоянии которых свой набор значений переменных p, q, r

Интерпретация К1 - модель формулы Φ

Для расширенной логики CTL* этот алгоритм очень сложен.
Мы рассмотрим такие алгоритмы для CTL и LTL.

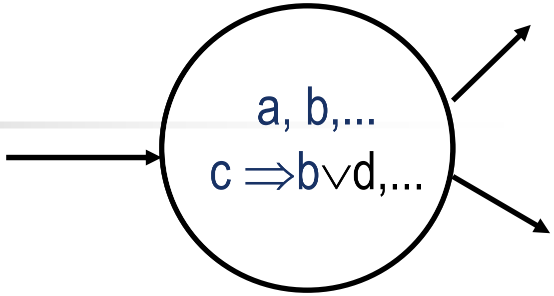
Лекция 5 – алгоритм model checking для CTL

Лекция 6 – алгоритм model checking для LTL

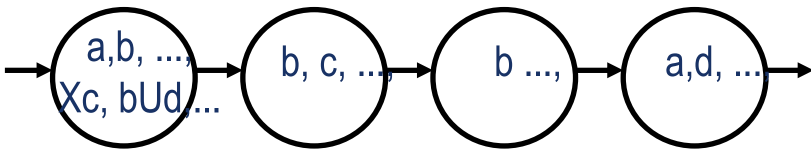
Персоналии

- Создатель современной теории линейной темпоральной логики и ее применений - Амир Пнуэли, профессор The Weizmann Institute of Science, Rehovot, Израиль
- В 1996 г. А.Пнуэли получил ACM премию Тьюринга
  - за выдающиеся результаты, которые ввели темпоральную логику в вычислительную науку;
  - за выдающийся вклад в верификацию программ и систем;
  - за идентификацию класса «*Реактивных систем (reactive systems)*» как систем, спецификация, анализ и верификация которых требуют специального подхода;
  - за разработку детальной методологии, основанной на темпоральной логике, для формального рассмотрения реактивных систем
- Логика ветвящегося времени – Э.Кларк, А. Эмерсон и многие другие

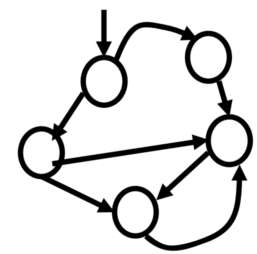
Заключение: TL – общие идеи



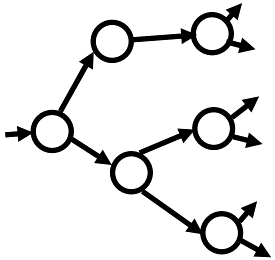
- Логика высказываний строится введением атомных утверждений и базисных операторов {∨, ¬}. По значениям истинности каждого атомарного утверждения можем вычислить истинность любой логической формулы

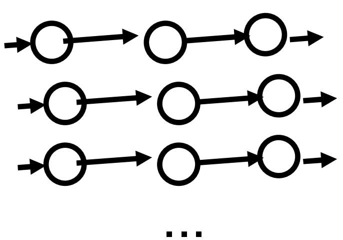


- В логике линейного времени LTL кроме атомарных утверждений и операций логики высказываний вводятся темпоральные операторы {U, X} (кроме них удобно использовать еще F и G)
- По конкретной цепочке состояний (миров) в каждом состоянии можем вычислить истинностные значения любой формулы темпоральной логики LTL



- В логике CTL* добавляются кванторы пути, позволяющие различать свойства различных путей
- Формула CTL* определена для конкретной интерпретации (структуры Крипке) и всех возможных ее вычислений





- CTL является подмножеством CTL* - в формулах CTL каждый темпоральный оператор предваряется квантором пути

Спасибо за внимание