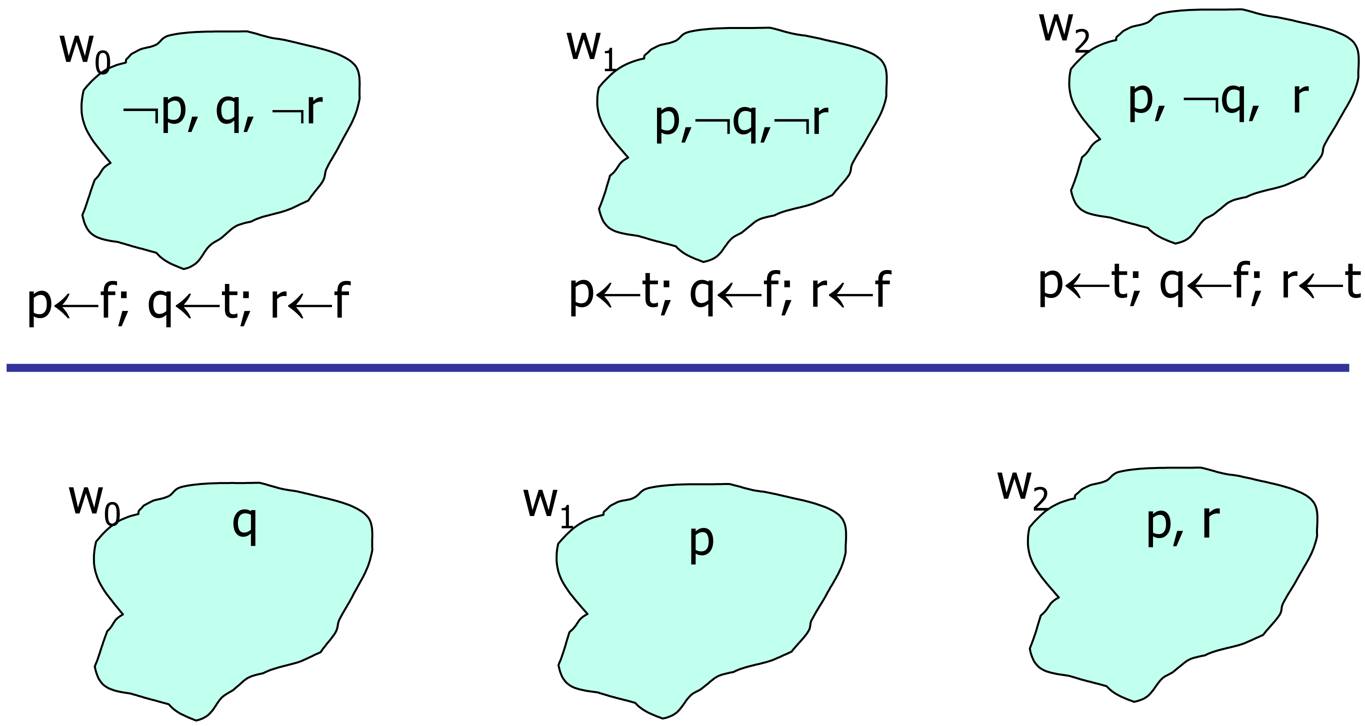
Темпоральные логики

Интерпретации формул логики высказываний – возможные миры

- Определение. Моделью логической формулы F называется такая интерпретация, на которой F истинна
- Определение. Интерпретациями формул логики высказываний являются наборы истинностных значений атомарных (неделимых) утверждений

p q r	F	R
0 0 0	0	0
0 0 1	0	1
0 1 0	1	1
0 1 1	0	0
1 0 0	0	0
1 0 1	0	1
1 1 0	1	1
1 1 1	0	1



- В каждом мире указаны те атомарные утверждения, которые в этом мире истинны

Нужна логика, задающая свойства динамики, поведения систем

- Классическая логика - примитивная модель истины: “черно-белая” модель (нет градаций “серого”), не существует степени уверенности-неуверенности, высказывания статичны, неизменны во времени $\Rightarrow$ неадекватна для высказываний о времени
- Пример - (некоммутативность конъюнкции, $A \wedge B \neq B \wedge A$):
  - “Джону стало страшно и он убил” $\neq$ “Джон убил и ему стало страшно”
  - “Джон умер и его похоронили” $\neq$ “Джона похоронили и он умер”
  - “Джейн вышла замуж и родила ребенка” $\neq$ “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! Сложные утверждения характеризуют динамические свойства процесса, развивающегося во времени

Темпоральная логика

- Определение
  Темпоральная логика - это 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, различных времен языка;
  - 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 развивающихся процессов, в т.ч. **вычислений** - выполнения программ
  - ТЕХНИКА: для формализации утверждений о свойствах **поведения** технических систем и свойствах процессов

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

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 $R_n(t)$ – в момент t лифт находится на этаже n

$$(\forall t\geq 0)\ (\forall\ t' \geq t)\ [\mathit{Запрос}_n(t)\ \&\ (\forall\ t'': t\leq t''< t')\ \neg \mathit{Обслужен}_n(t'') \Rightarrow \neg R_n(t'')]$$

“Мы не друзья, пока ты не извинишься”

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

В предикатной логике громоздкая нотация, тяжелый формальный аппарат

Введение модальностей

- Определение Модальной логики
  - Модальность (от лат. modus – вид, способ, наклонение) – это категория, как-то характеризующая следующее за ней утверждение
  - Модальная логика - любая формальная логическая система, в которой присутствуют модальные операторы
- Примеры модальных операторов возможности/необходимости :
  - M - “возможно, что” (Mp – “возможно, что p”)
  - L - “необходимо, что” (Lq – “q обязательно выполняется”)
  - Соотношения между модальностями: $Lp \equiv \neg M \neg p$
  - Примеры: $\neg Mp \neq M\neg p$ Не может писать $\neq$ может не писать !!
    $Lp \equiv \neg M\neg p$: должен писать тогда и только тогда, когда не может не писать
- Примеры темпоральных модальных операторов:
  - Fp – “когда-нибудь в будущем p выполнится обязательно” $F\neg p \neq \neg Fp$
  - Gp – “всегда в будущем будет выполняться p” $G\neg p \neq \neg Gp$

  Соотношения между модальностями:

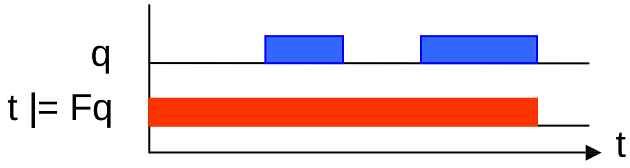
  $Fp \equiv \neg G \neg p$ – “нечто когда-то будет $\equiv$ неверно, что это никогда не будет”

  $Gp \Rightarrow Fp$ – “если нечто будет всегда, то оно когда-нибудь случится”

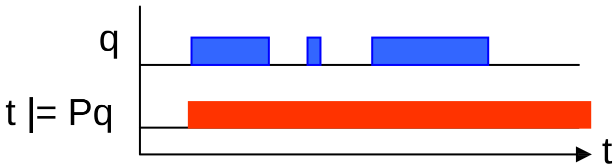
Tense Logic (временная логика) – вариант модальной логики

Впервые - философ Diodorus Cronus. В 20 веке – Артур Прайор (Arthur Prior)

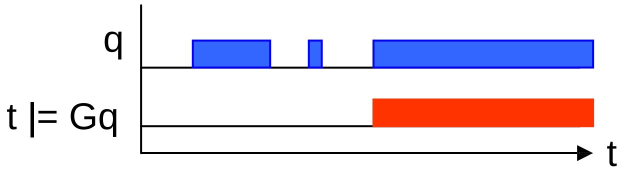
Fq – q обязательно случится когда-нибудь в будущем:



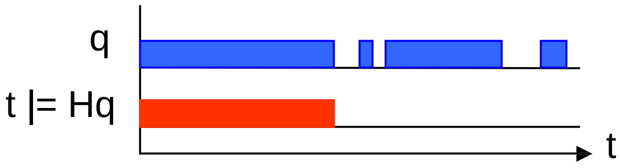
Pq – q случилось когда-то в прошлом:



Gq – q в будущем всегда будет истинно:

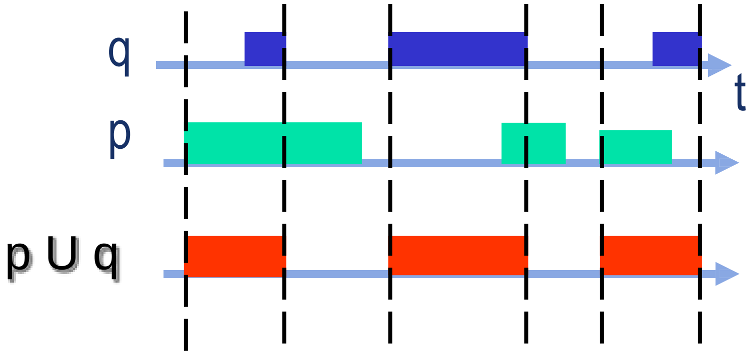


Hq – q всегда было в прошлом:



Дополнительные модальности TL: Until, X

U (Until)



pUq - q когда-то в будущем обязательно выполнится, а до этого все время будет выполняться p

X (Next time)

Xp - p истинно в следующий момент времени
если мы рассматриваем в ДИСКРЕТНОМ времени

F и G выражаются через U:

$Fp \equiv true\ U\ p$,

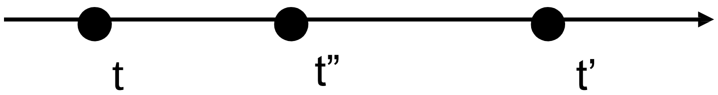
$Gp \equiv \neg F \neg p$

Примеры формализаций высказываний

- Джейн вышла замуж и родила ребенка
  P(Джейн_выходит_замуж $\wedge$ F Джейн_рожает_ребенка)
- Джейн родила ребенка и вышла замуж
  P(Джейн_рожает_ребенка $\wedge$ F Джейн_выходит_замуж)
- Джон умер и его похоронили
  P(Джон_умирает $\wedge$ XF Джона_хоронят)
- Если я видел ее раньше, то я ее узнаю при встрече
  G(P Увидел $\Rightarrow$ G(Встретил $\Rightarrow$ X Узнал))
- Ленин – жил, Ленин – жив, Ленин – будет жить (В. Маяковский)
  PG Ленин_жив
- Любое посланное сообщение будет получено
  G(Послано(m) $\Rightarrow$ F Получено(m))
- Вчера он сказал, что придет завтра, значит, он придет сегодня
  X^{-1}X Прихожу $\equiv$ Прихожу
- Сократ умер
  PG $\neg$Сократ_жив

Логика предикатов и Tense Logic

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



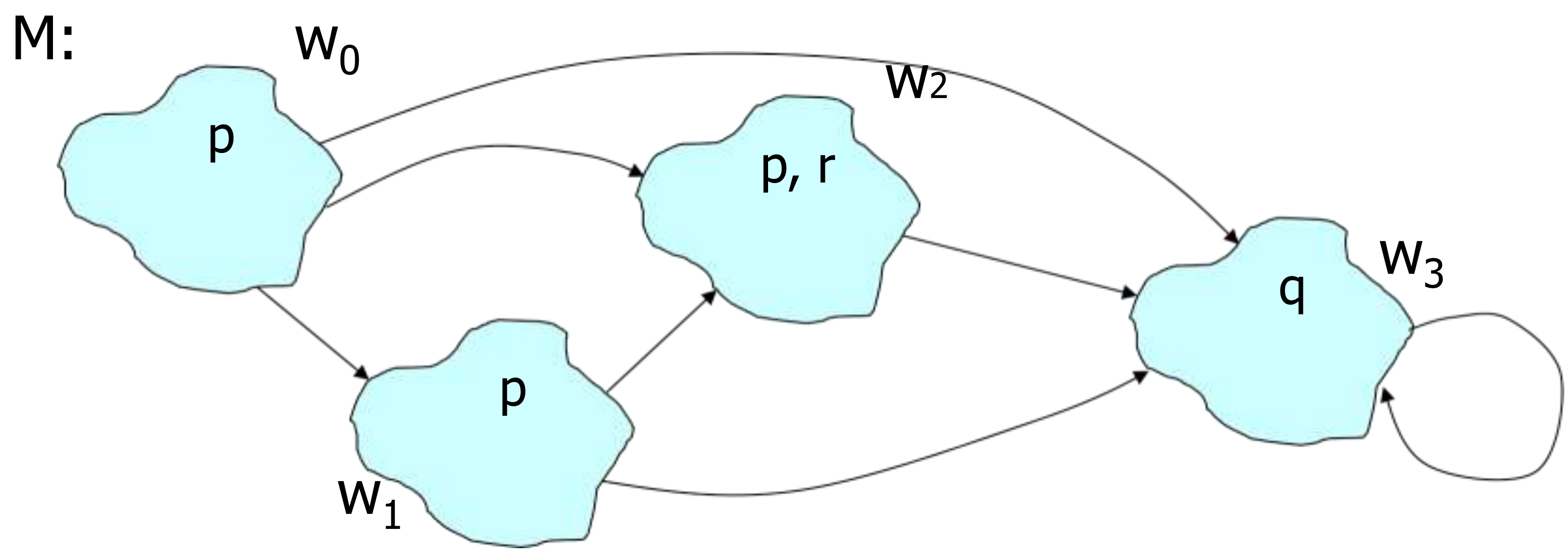
Пусть R_n (t) – в момент t лифт находится на этаже n

$(\forall t \geq 0)\ (\forall\ t' \geq t)\ [\mathit{Запрос}_n\ (\ t\ )\ \&\ (\forall\ t'': t \leq t'' < t')\ \neg \mathit{Обслужен}_n\ (t'') \Rightarrow \neg R_n(t'')]$

$G\ [\mathit{Запрос}_n \Rightarrow \neg R_n\ U\ \mathit{Обслужен}_n\ ]$

Спецификация свойств в TL – ясная, четкая, компактная

Семантика Крипке – множество возможных миров, связанных отношением достижимости



Семантика Крипке для утверждений, истинность которых зависит от времени:

- В мире w_1 возможно в будущем выполнится r
- В мире w_2 обязательно в будущем выполнится q
- В мире w_3 всегда в прошлом было истинным p
- В мире w_0 обязательно когда-то выполнится q, а до этого будет истинно p

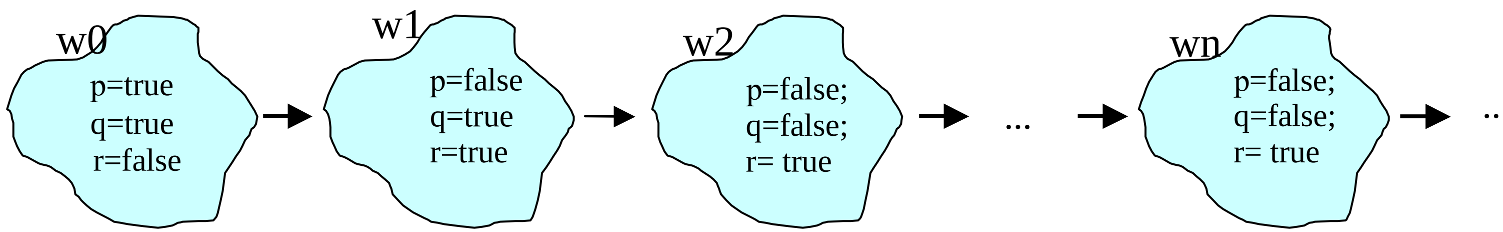
Миры Крипке

- Saul Kripke, Sep. 1958: “In an indetermined system, we perhaps should not regard time as a linear series, as you have done. Given the present moment, there are several possibilities for what the next moment may be like – and for each possible next moment, there are several possibilities for the moment after that. Thus the situation takes the form, not of a linear sequence, but of a ’tree’ ”. (в это время Крипке был студентом 18 лет в high-school, в г. Omaha, Nebraska.)
- **Семантика Крипке.** Рассматривается множество миров. Модальное высказывание $F\varphi$ считается истинным, если φ истинно в некоторых из возможных миров. Истинность обычных формул логики измеряется по отношению к текущему миру. (Идея принадлежит Лейбницу и была разработана Солом Крипке).
- Возьмём произвольное множество W; его элементы будем называть **мирами** или **состояниями**. Рассмотрим произвольное бинарное отношение R на W. Если значение предиката R(t, w) равно 1, то w называется **возможным** или **доступным** миром для t.
- **Определение.** Пара множеств (W, R), где W – непустое множество, а $R \subseteq W \times W$ – бинарное отношение на W, называется **шкалой Крипке**. Отношение R называется отношением **доступности**.

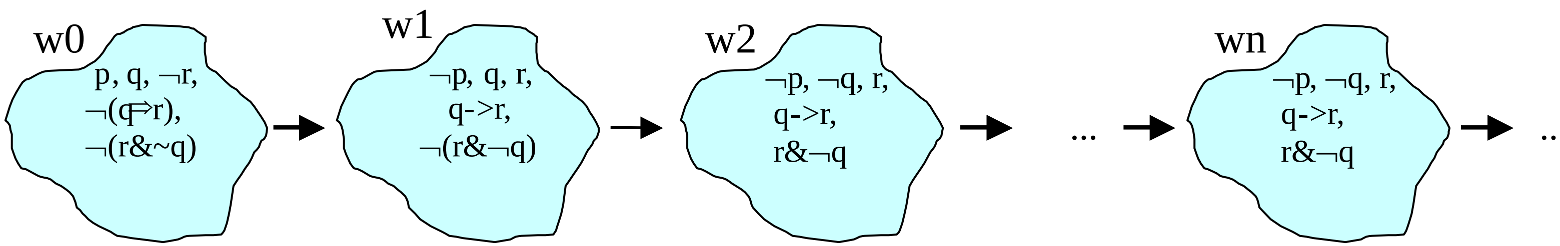
Амир Пнуэли (1977)

Темпоральная логика в дискретном времени

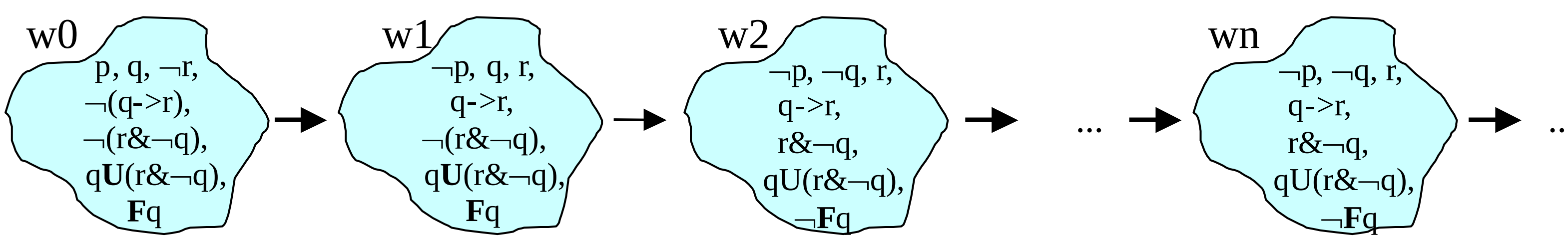
Последовательность “миров” Лейбница, в каждом свое понимание истинности:



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), qU(r&¬q), Fq, ... потому что они истинны в w0

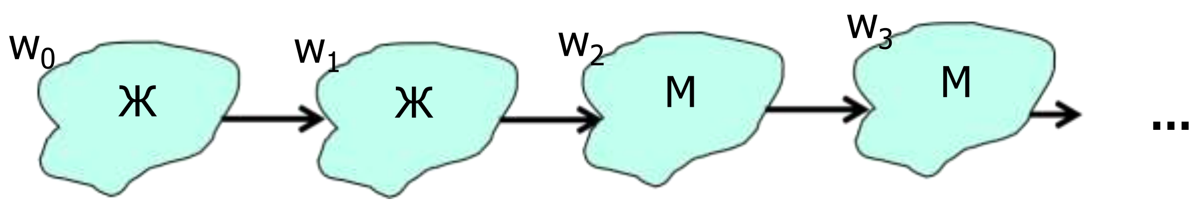
Логический парадокс - софизм

- Секст Эмпирик, т.2, с.289, М., 1976:
  - Если умер Сократ, то он умер, когда он был живой, или когда был мертвый.
  - Если когда он был живой, то он не умер, так как один и тот же человек и жил бы, и был бы мертв
  - И не тогда, когда он был мертвый, ибо он был бы дважды мертвым.

  Стало быть, Сократ не умер

- Схема рассуждения:
- F1: $У \Rightarrow Ж \vee М$
  F2: $Ж \Rightarrow \neg У$
  F3: $М \Rightarrow \neg У$

  R: $\neg У$



- Схема рассуждения правильна. Ошибка в том, что классическая логика не имеет понятия времени, в рамках классической логики нельзя формализовать рассуждения о событиях, происходящих в разные моменты времени
- Здесь: утверждение У относится к событию перехода между мирами, а такой переход мы считаем мгновенным, неделимым

Линейная темпоральная логика
Linear Temporal Logic (LTL)

- Язык формальной логики имеет:
  синтаксис (правила построения формул) и
  семантику (правила, определяющие истинностное значение формул)
- Определение синтаксиса LTL задается всего тремя правилами:
  - Формула LTL: это :
    - атомарное утверждение p, q, ...,
    - или формулы, связанные логическими операциями □, □
    - или формулы, связанные темпоральными операторами U, X

Другие (выводимые) темпоральные операторы:

$$Fp \equiv \text{true } U\ p$$

$$Gp \equiv \neg F\ \neg p$$

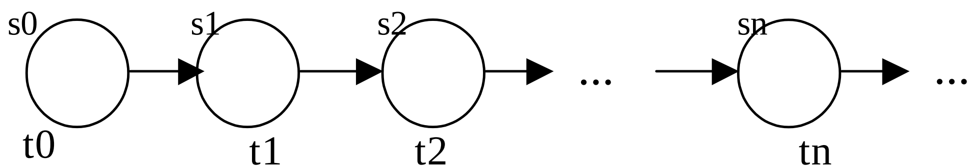
Операторы прошлого не используются
Прошлое при анализе технических систем менее важно

Формализация высказываний в LTL

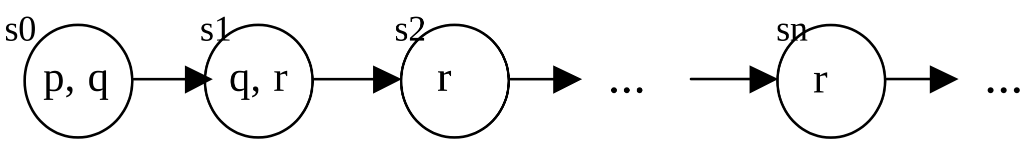
- “Dum spiro, spero” - пока живу – надеюсь
  - **G**(я_живу $\Rightarrow$ я_надеюсь)
- “Мы придем к победе коммунистического труда!”
  - **F** коммунистический_труд_победил!
- “Сегодня он играет джаз, а завтра Родину продаст!” (В.Бахнов)
  - он_играет_джаз $\Rightarrow$ **X**он_продает_Родину – слишком буквально
  - **G**(он_играет_джаз $\Rightarrow$ **FX**он_продает_Родину)
- p = “я люблю Машу”, q =“я люблю Дашу”
  - **F**p – “я когда-нибудь обязательно полюблю Машу”
  - q**U**p – “я полюблю Машу, а до этого буду любить Дашу”
  - **FG**p – “когда-нибудь в будущем я полюблю Машу навечно”
  - **GF**q – “я буду бесконечно влюбляться в Дашу ”
- “Раз Персил – всегда Персил”
  - **G**(Персил $\Rightarrow$ **G**Персил) – раз попробовав, будешь использовать всегда
- “Я твоя навеки!”
  - **G** Я_твоя!

LTL и анализ дискретных технических систем

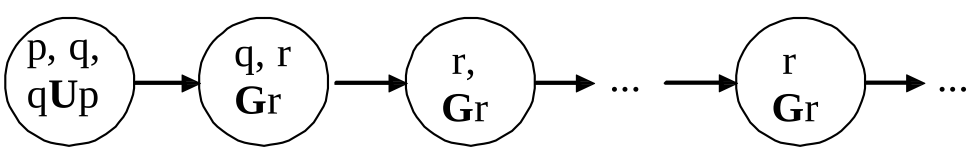
Последовательность “миров” в LTL можно трактовать как бесконечную последовательность состояний дискретной системы, а отношение достижимости – как дискретные переходы системы:



Атомарные формулы - базисные свойства процесса в состояниях:

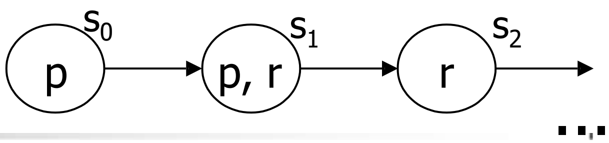


Производные темпоральные формулы в состояниях – это свойства динамического процесса, характеризующие вычисление в будущем:



qUp – выполняется на вычислении
Gr – не выполняется на вычислении

Семантика операторов LTL



Обозначения: $\sigma_i \models \varphi \equiv$ в состоянии s_i вычисления $\sigma = s_0 s_1 s_2 s_3 \ldots$ формула φ истинна

$\sigma_i \models p$	iff	в состоянии s_i истинен атомарный предикат p
$\sigma_i \models \neg\varphi$	iff	$\sigma_i \not\models \varphi$
$\sigma_i \models \varphi \vee \psi$	iff	$\sigma_i \models \varphi$ или $\sigma_i \models \psi$
$\sigma_i \models X\,\varphi$	iff	$\sigma_{i+1} \models \varphi$
$\sigma_i \models \varphi\, U\, \psi$	iff	$(\exists j: j \geq i)\,[\,\sigma_j \models \psi \wedge (\forall k: i \leq k < j)\ \sigma_k \models \varphi\,]$
$\sigma_i \models F\,\varphi$	iff	$(\exists j: j \geq i)\ \sigma_j \models \varphi$
$\sigma_i \models G\,\varphi$	iff	$(\forall j: j \geq i)\ \sigma_j \models \varphi$

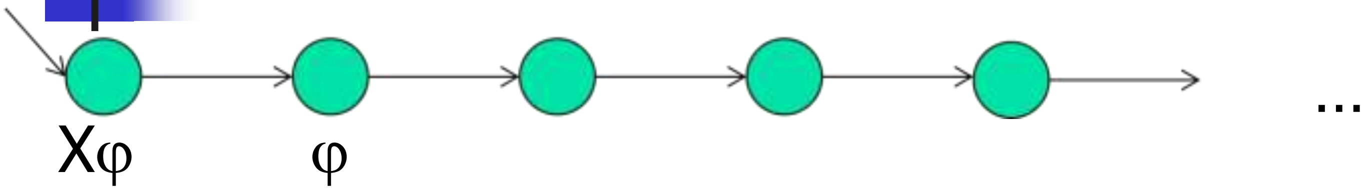
Выводимые операторы: **F**p $\equiv$ true **U** p, **G**p $\equiv \neg$**F** $\neg$p

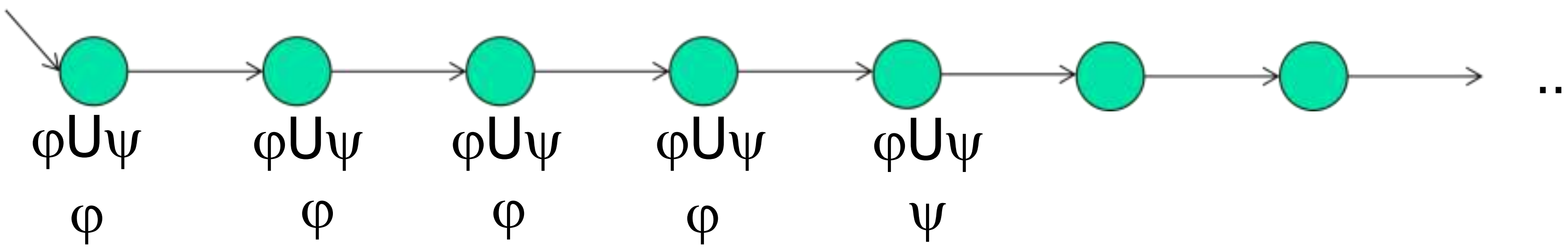
Задача: выведите семантику операторов Fp и Gp из семантического определения pUq

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

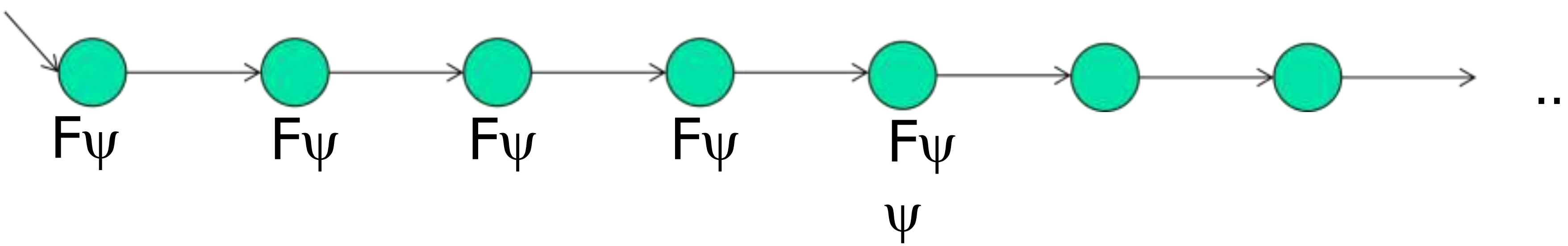
Φ выполняется на σ, если Φ выполняется в начальном состоянии σ

Семантика основных темпоральных операторов





$$\varphi U\psi = \psi \vee \varphi \wedge X\varphi U\psi$$

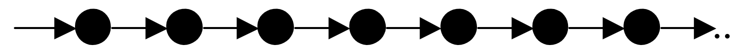


$$F\psi = \psi \vee XF\psi$$

$$F\psi = True U\psi$$

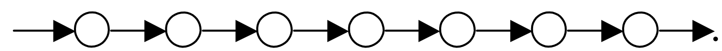
Примеры формул LTL для дискретных систем

G q - *всегда в будущем*

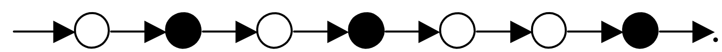


F q - *хотя бы раз в будущем*

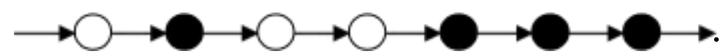
$\neg$F q – *никогда в будущем*



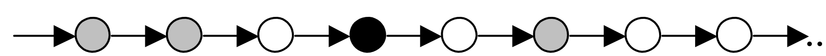
GFq – *бесконечно много раз в будущем*



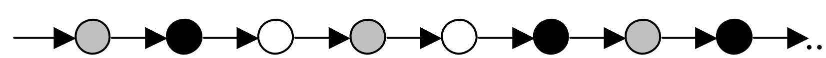
FGq – *с какого-то момента постоянно*



$p \Rightarrow Fq$ – *на p в s_0 будет реакция q когда-нибудь в будущем*



$G[p \Rightarrow Fq]$ – *всегда на p будет реакция q*



Пример свойства LTL

- LTL: Жизнь конечна

  Жизнь – это непрерывный интервал времени, когда мы живем. После начального момента времени t_0 мы когда-то начинаем жить, живем непрерывно, и когда-то кончим жить, но уж если кончили, то не воскреснем! Атомарный предикат q: “Я жив”

$q \wedge XF\neg q \wedge G(\neg q \Rightarrow G\neg q)$

Сейчас я жив, но когда-нибудь умру, и если умру, то уж навсегда

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?
  - строим формальную запись высказывания

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

Пример спецификации свойства протокола

- Формализуем свойство P причинно-следственной связи событий "посылка сообщения send всегда, в конце концов, приведет к получению подтверждения ack"

- send $\Rightarrow$ ack

  это неправильная формализация свойства P. Построенная формула имеет следующий смысл: "если в начальном состоянии send истинно, то в том же состоянии истинно и ack".

- **G**send $\Rightarrow$ ack

  поскольку темпоральные операторы имеют более высокий приоритет, чем логические связки, эта формула будет пониматься как (**G**send)$\Rightarrow$ack, т.е. она имеет следующий смысл: "если во всех состояниях вычисления выполняется send, то в начальном состоянии должно выполняться ack"

Пример спецификации свойства протокола (2)

- Формализуем свойство P причинно-следственной связи событий “посылка сообщения send всегда, в конце концов, приведет к получению подтверждения ack”

- **G**(send $\Rightarrow$ ack)

Эта формула тоже неверна: она говорит, что в любом состоянии вычисления, если в нем выполняется send, то в этом же состоянии должно выполняться и ack

- **G**(send $\Rightarrow$ **F**ack)

Эту формулу часто называют “темпоральным следствием”. Смысл ее в том, что в любом состоянии вычисления если выполняется send, то когда-нибудь в будущем (включая настоящее) должно выполняться и ack. Она часто используется для спецификации свойств параллельных систем. Под названием “leads-to” эта формула была введена Овицки и Лэмпортом в 1982 г. и получила там свой знак “~>”: send ~> ack.

Но она будет истинна и в тех случаях, когда в том же состоянии, в котором истинно send, будет истинно и ack

Пример спецификации свойства протокола (3)

- Формализуем свойство причинно-следственной связи событий "посылка сообщения send всегда, в конце концов, приведет к получению подтверждения ack"

- **G**(send $\Rightarrow$ **XF**ack)

Эта формула более точно отражает причинно-следственную связь между событиями send и ack, но она выполняется и в том случае, если событие send вообще никогда не наступит!

- **G(F**send $\wedge$ (send $\Rightarrow$ **XF**ack))

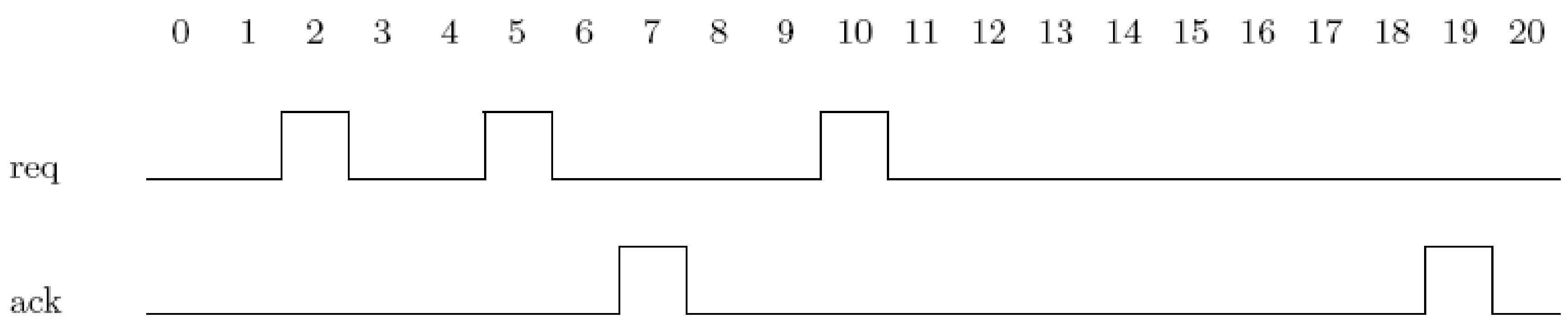
Эта формула наиболее полно отражает идею причинно-следственной связи событий send и ack в протоколе передачи информации: она утверждает, что send в вычислении будет наступать неопределенно часто, и всегда после посылки сообщения send, затем, когда-нибудь в будущем процесс получит подтверждение ack.

Спецификация свойства поведения дискретной синхронной схемы

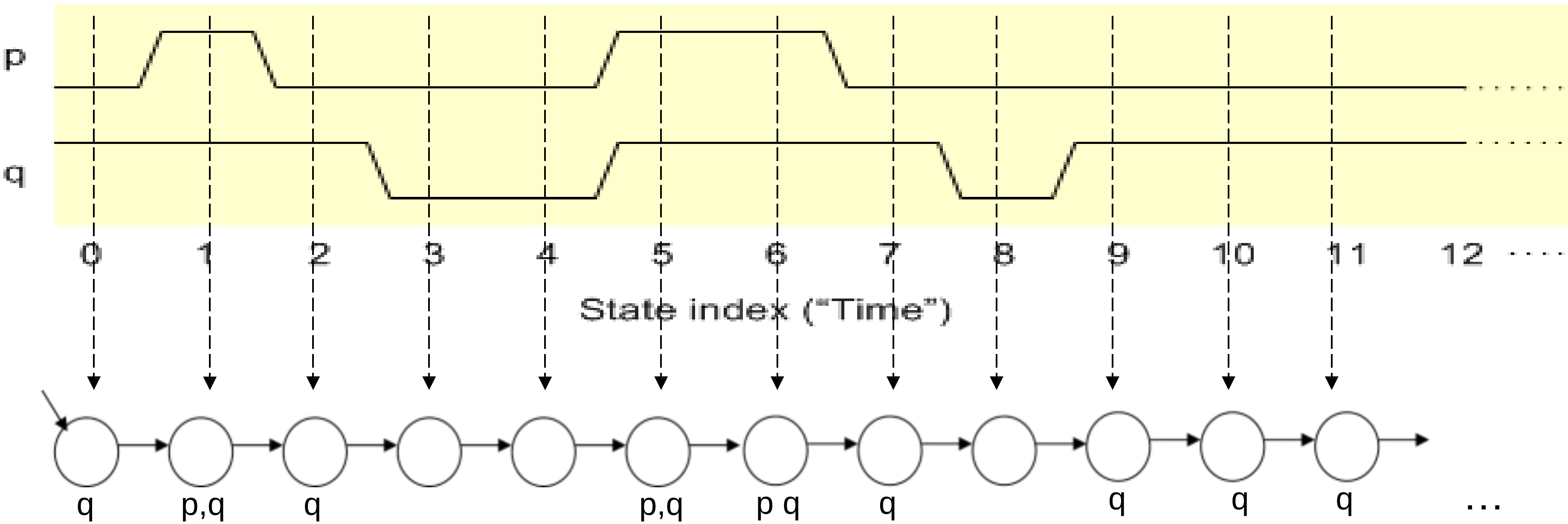
Пример поведения, удовлетворяющего формуле G(req ⇒ F ack)

- G(req ⇒ F ack) – всегда запрос req когда-нибудь в будущем будет подтвержден сигналом ack

  (req – request, запрос ack – acknowledge, подтверждение)



Спецификация свойств дискретных синхронных схем (LTL)



$\sigma_0 \models \neg p$

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

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

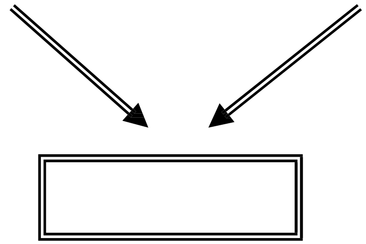
$\sigma_0 \models \mathrm{q\ U\ p}$

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

Требования к ПО в системе параллельных процессов

P1::

```
k0: noncritical1;
k1: x1 := busy;
k2: await x2 = free;
k3: critical1;
k4: x1 := free;
k5: goto k0;
```



P2::

```
m0: noncritical2;
m1: x2 := busy;
m2: await x1 = free;
m3: critical2;
m4: x2 := free;
m5: goto m0;
```

x1=free

x2=free

процесс P находится в состоянии с меткой k

- Каждый раз, когда какой-нибудь процесс запросил ресурс, то, рано или поздно, он этот ресурс получит

  $\mathbf{G}$ [(P1@k2 $\Rightarrow$ $\mathbf{F}$ P1@k3) & (P2@m2 $\Rightarrow$ $\mathbf{F}$ P2@m3)]
- Всегда, если процесс занял ресурс, он, в конце концов, его освободит

  $\mathbf{G}$ [(P1@k3 $\Rightarrow$ $\mathbf{F}$ P1@k5) & (P2@m3 $\Rightarrow$ $\mathbf{F}$ P2@m5)]
- Ресурс может быть использован в один и тот же момент не более, чем одним процессом

  $\mathbf{G}$ [$\neg$(P1@k3 & P2@m3)]
- Если процесс P1 хочет использовать ресурс, который в это время используется процессом P2, то процесс P2 повторно не сможет получить доступ к ресурсу до того, как процесс P1 получит этот ресурс

  $\mathbf{G}$ [(P1@k2 & P2@m3) $\Rightarrow$ P2@m3 $\mathbf{U}$ ($\neg$P2@m3 $\mathbf{U}$ P1@k3)]

Формулы для выражения свойств программ

- **GF** enabled
  - “Свойство enabled будет истинным бесконечное число раз на всех траекториях системы”
- **GF** true
  - Свобода от дедлоков (блокировок): “для каждого достижимого состояния существует возможность продолжения функционирования”
- $\mathbf{G}(send \Rightarrow \mathbf{X}(\neg send\,\mathbf{U}\,receive))$
  - На вычислениях системы выполняется следующее свойство: “если выполнится send, то со следующего состояния в будущем обязательно выполнится receive, а до этого момента send не будет выполняться”
- $\mathbf{G}(input \Rightarrow \mathbf{X}\ (output \vee \mathbf{X}output))$
  - “Как только установится сигнал input, по крайней мере через два следующих шага вычисления будет установлен output”
- $\mathbf{G}(\ crint \Rightarrow \mathbf{F}\ \neg crint\ )$
  - “Если процесс вошел в критическую секцию, когда-нибудь в будущем он из нее выйдет”. Эта формула может отражать требование конечности времени нахождения процесса в критической секции

Линейное и ветвящееся время

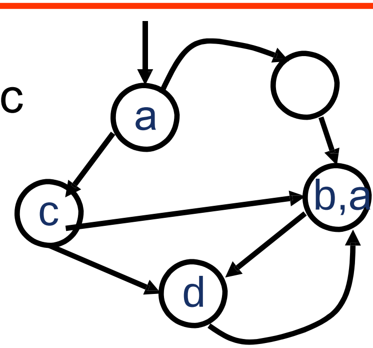
По заданной бесконечной цепочке состояний с определенным в каждом состоянии набором истинных атомарных предикатов нужно вычислить значение булевых и темпоральных формул. Как вычисления представить конечным образом?

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

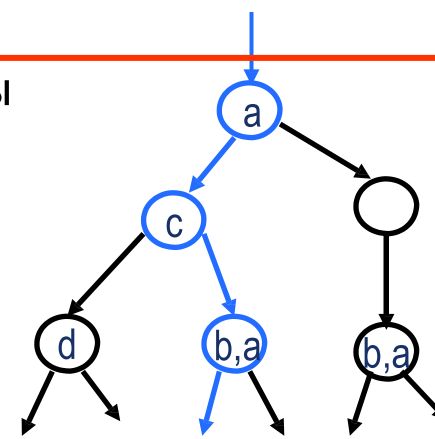
Для формализации этого введена структура Крипке

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

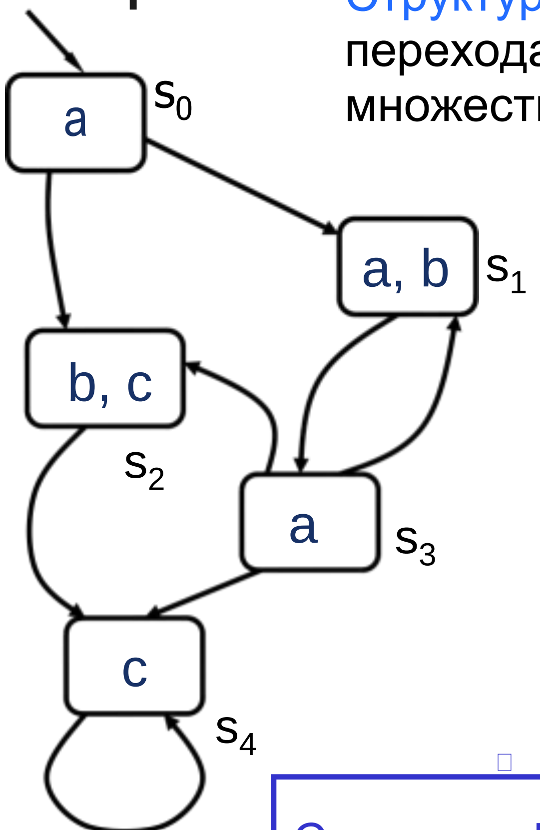
Структура Крипке – система переходов с помеченными состояниями и непомеченными переходами



Развертка структуры Крипке определяет бесконечные цепочки состояний - возможные ВЫЧИСЛЕНИЯ



Структура Крипке – интерпретация формул TL



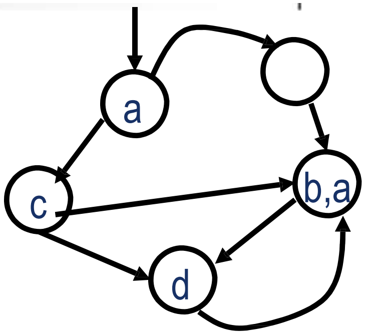
Структура Крипке – это конечный автомат с непомеченными переходами, с каждым состоянием которого связано некоторое множество простых утверждений, истинных в этом состоянии

- Формально: $M = (S, S_0, R, L)$, где:
  - S – конечное множество состояний
  - S_0 - множество начальных состояний
  - R – подмножество SxS – множество переходов; $(\forall s)(\exists s'): (s,s') \in R$
  - AP – множество атомных утверждений
  - $L: S \to 2^{AP}$ – функция пометок: каждому состоянию s – некоторое подмножество атомарных утверждений, истинных в s
- 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

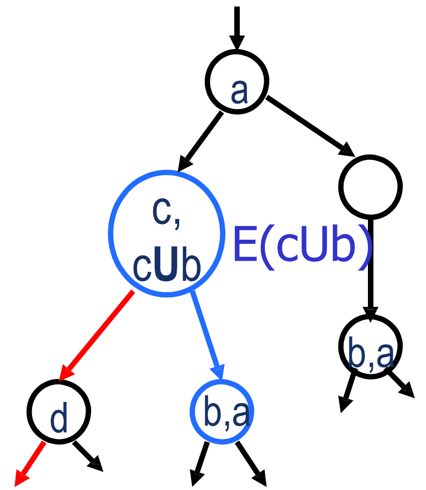
Как идеи TL применить к ветвящемуся времени?



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

Но как понимать формулы LTL: **F**p, p**U**q, … в состоянии s?
Это формулы пути, но какого пути? Из состояния – много путей

Ввести квантор "*пути*" (path quantifier)



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

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

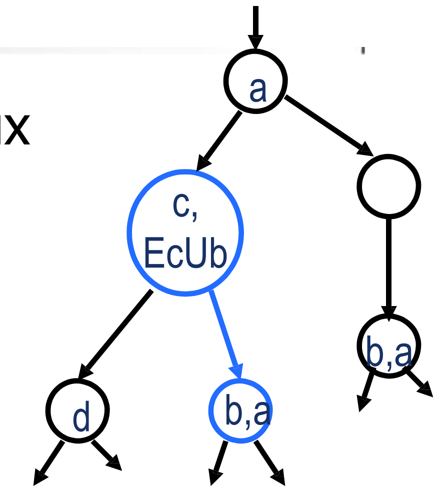
A $\phi \equiv$ "*для всех путей из данного состояния LTL формула* ϕ *истинна*"

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!)

Логика ветвящегося времени – CTL



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

Темпоральная логика ветвящегося времени рассматривает траектории на развертке структуры Крипке

- Формула φ логики ветвящегося времени CTL - это формула, характеризующая состояние вычисления:
  - атомарное утверждение p, q, ...,
  - или формулы CTL, связанные логическими операциями $\vee$, $\neg$
  - или формулы CTL, связанные темпоральными операторами, перед каждым из которых стоит квантор пути E или A:
    - EX φ, AX φ,
    - EF φ, AF φ,
    - EG φ, AG φ,
    - E (φ_1 U φ_2), A (φ_1 U φ_2)

Примеры формул логики ветвящегося времени

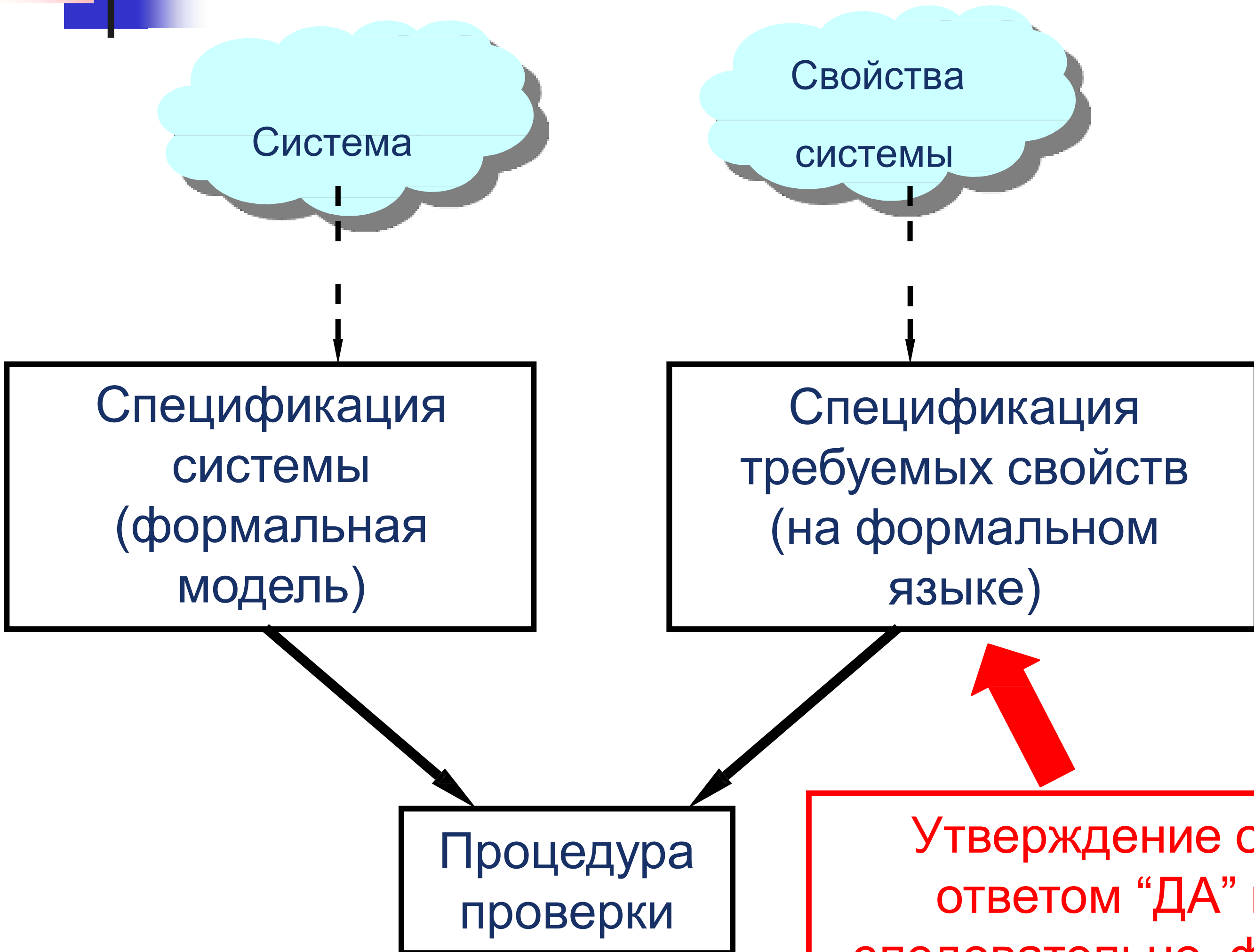
- Обозначение: p = “Я люблю Машу”
- AG p:
  - “Я люблю Машу, и, что бы ни случилось, я буду любить ее всегда”
- AF p:
  - “Что бы ни случилось, я в будущем полюблю Машу”
- EF p:
  - “Я не исключаю такого развития событий, что в будущем я полюблю Машу”
- $\neg$E ($\neg$Captain_AntiFire_Permission U AntiFire)
  - Не существует режима, в котором включение противопожарного устройства производится без предварительной санкции

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 вычислений (цепочек состояний)
  - формулы CTL – на конечном множестве состояний, связанных отношениями перехода (фактически, на деревьях вычислений)
- Методы анализа (проверки выполнения формулы) совершенно разные
- Выразительная мощь несравнима
  - некоторые формулы LTL выражают свойства, которые выражают и формулы CTL, но есть формулы CTL, невыразимые в LTL, и наоборот

Проверка свойств поведения систем

Формально доказать можно только нечто формально определенное

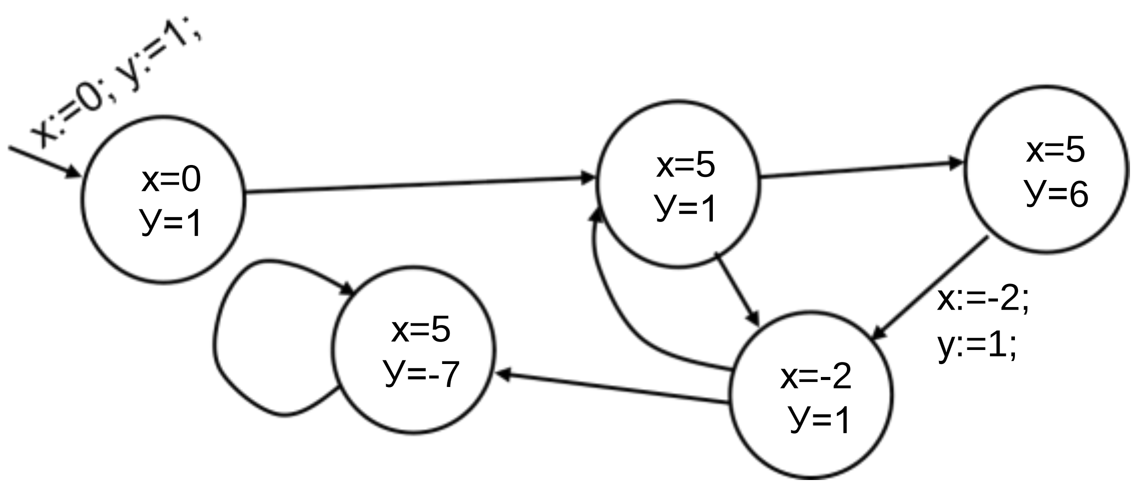


Утверждение о свойствах с ответом “ДА” или “НЕТ” – следовательно, формула логики (обычно – Темпоральной логики)

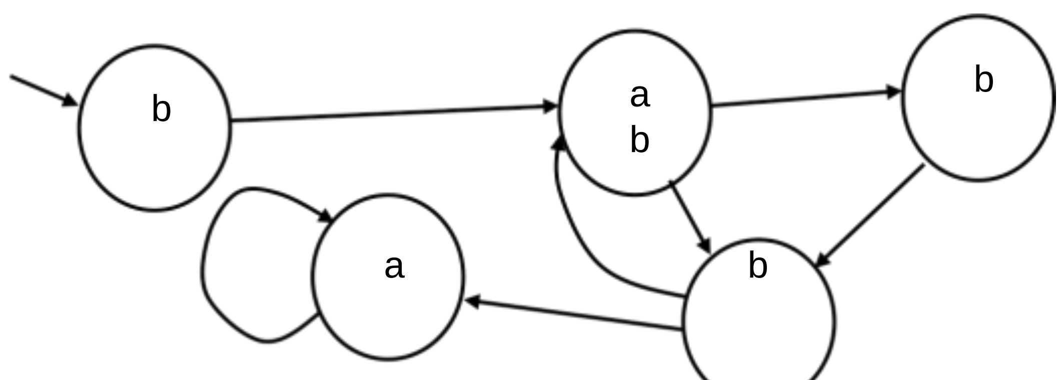
Структура Крипке как модель программы

```
begin
x:=0; y:=1;
while x+y < 5 do
{ x:=5;
if y=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. Число состояний должно быть конечным:



Пусть атомарные утверждения, *ИНТЕРЕСУЮЩИЕ НАС:*
$a=x>y$; $b=|x+y|<3$

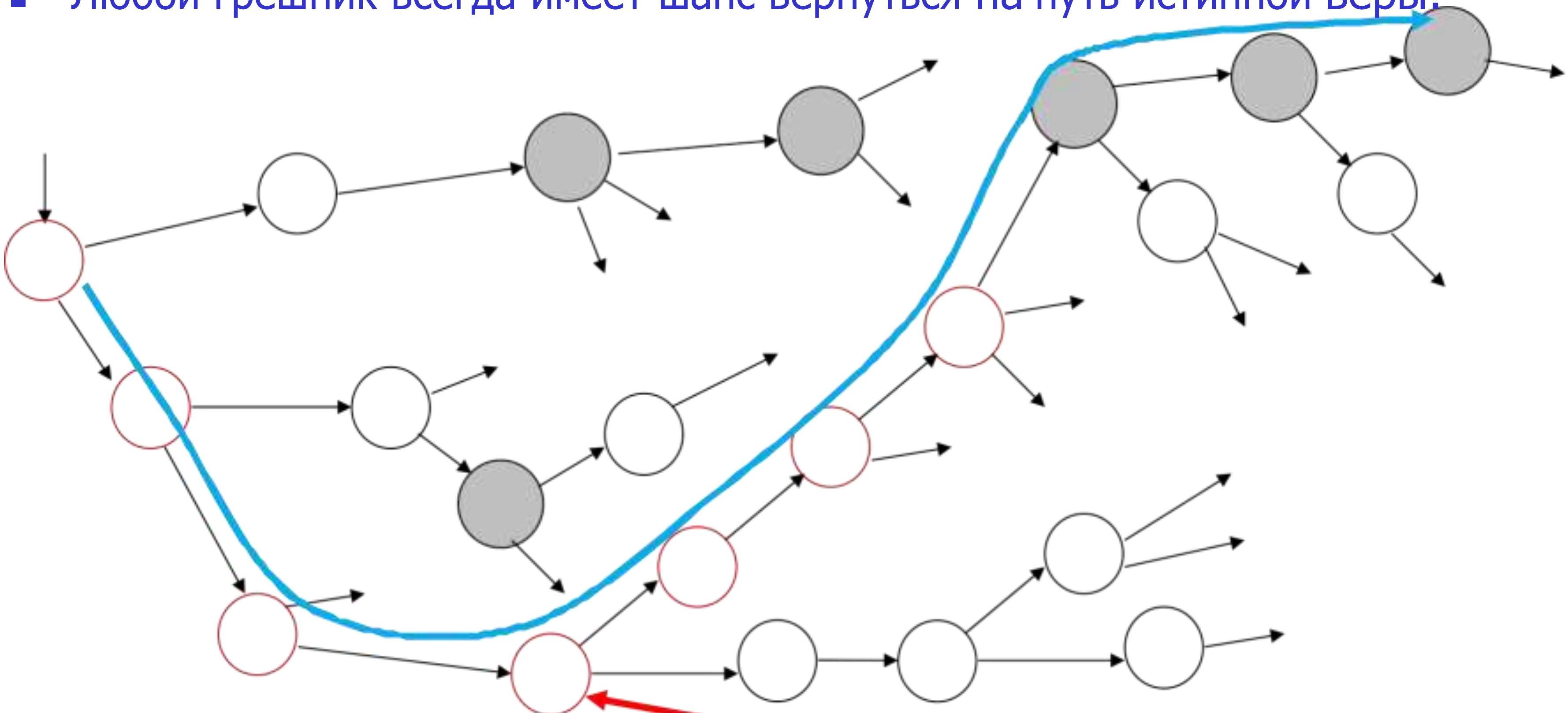


Примеры проверяемых свойств (CTL)

- EF(Start & ¬Ready)
  - в программе возможно достичь состояния, в котором свойство Start выполняется, а Ready – нет
- AGAF Restart
  - при любом функционировании системы (на любом пути) из любого состояния системы всегда обязательно вернемся в состояние рестарта
- AG EF Restart
  - при любом функционировании системы (на любом пути) из любого состояния системы существует путь, по которому можно перейти в состояние рестарта
- E[p U A [q U r]]
  - существует путь, на котором p выполняется до тех пор, пока в будущем q будет всегда выполняться до выполнения r

Пример свойства, выраженного в CTL

- Любой грешник всегда имеет шанс вернуться на путь истинной веры:



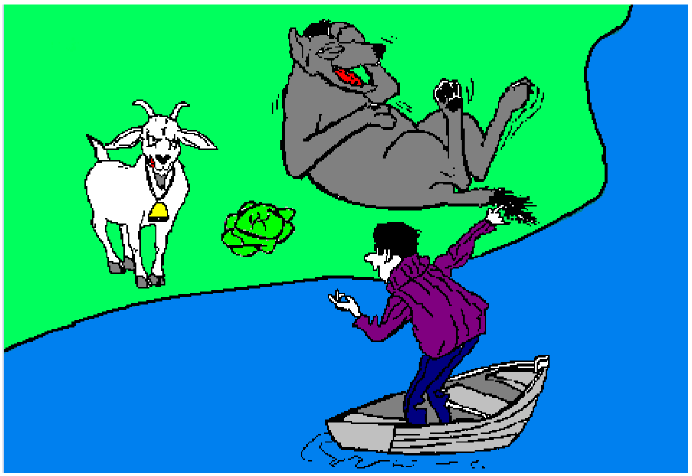
AG EF EG ‘истинная вера ’

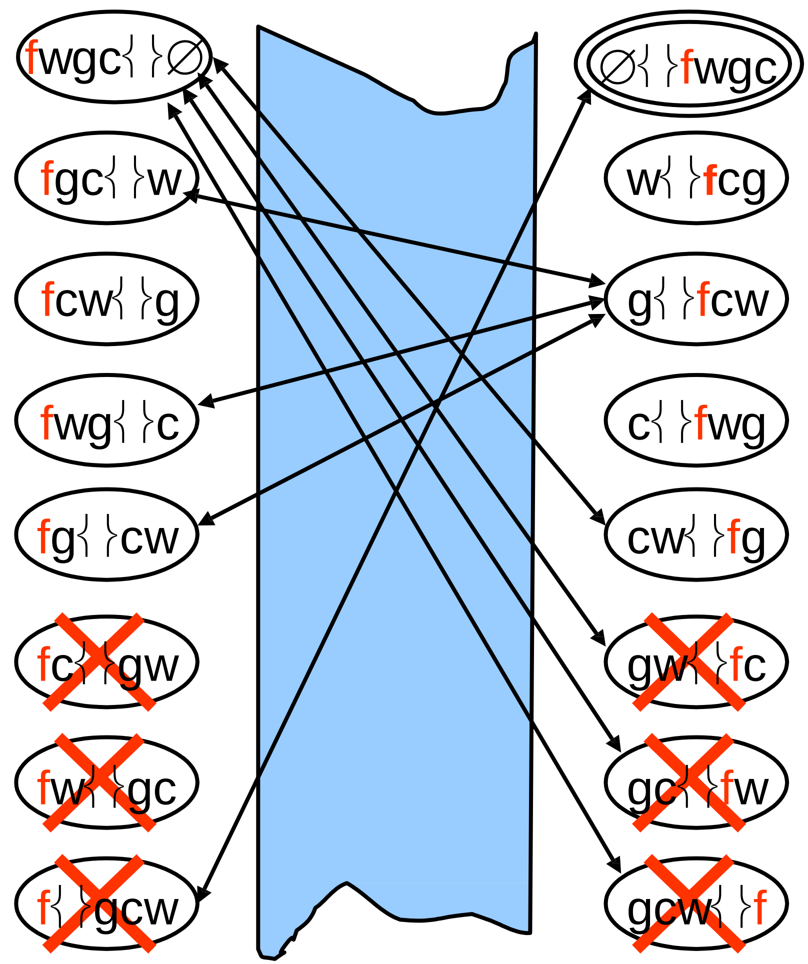
Сюда попал грешник

В любом состоянии нашей жизни (AG) существует такой путь (E), что на нем в конце концов (F) попадем в состояние, с которого существует непрерывный “истинный” путь (EG “истинная вера”)

Логические задачи: фермер, волк, коза и капуста

Старинная задача: Фермеру нужно перевезти через реку волка, козу и капусту. Лодка вмещает кроме фермера еще только что-то одно. Но без фермера нельзя волка оставлять с козой, а козу – с капустой





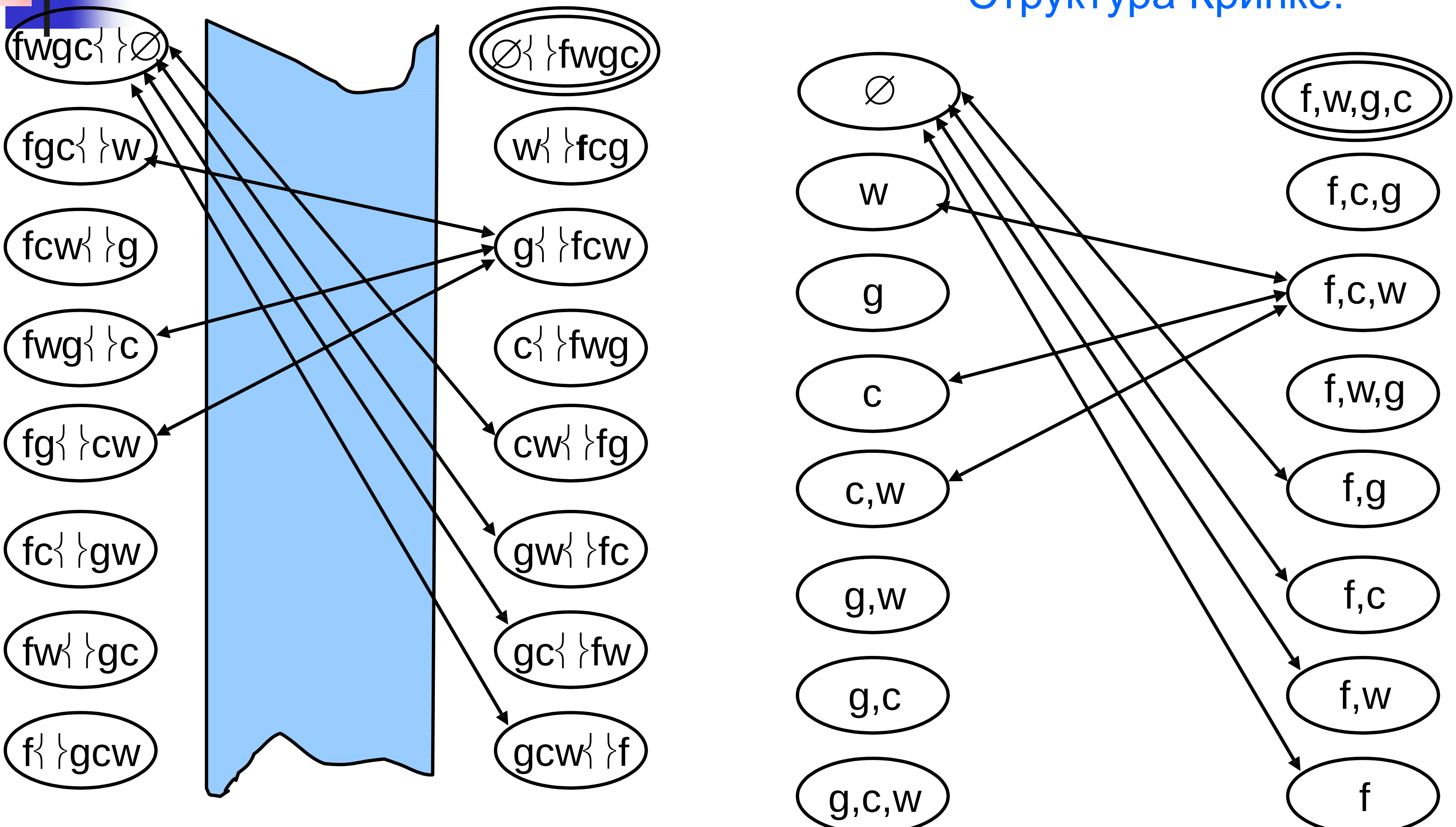
Какую абстракцию построить для решения?

Хотя система не дискретная, для решения задачи можно построить абстрактную модель - систему переходов, и исследовать возможные пути

Состояние: w{ }cgf

Model checking для этой проблемы

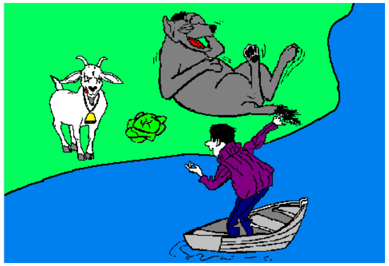
Структура Крипке:



Атомарные предикаты:

f=1 – фермер на правом берегу, w=1 – волк на правом берегу,
c=1 – капуста на правом берегу, g = 1 – коза на правом берегу

Свойства системы фермер-волк-коза-капуста (LTL)



1. В заключительное состояние можно попасть

$F(fwgc)$

2. Не существует такого решения задачи, при котором, если коза переедет на ту сторону, она там навсегда и останется

$\neg\ [\ F\ (fwgc)\quad \&\ ((\neg g)\ U\ Gg)\ ]$

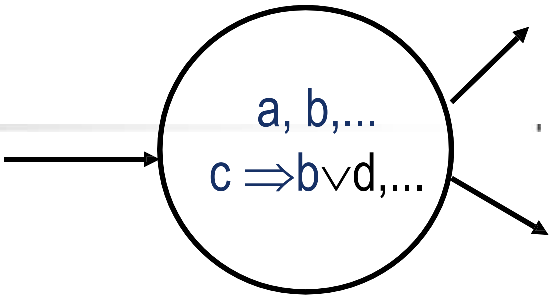
3. Путь, удовлетворяющий решению задачи, существует

$((g \equiv c) \vee (g \equiv w) \Rightarrow (g \equiv f))\ U\ (fwgc\ )$

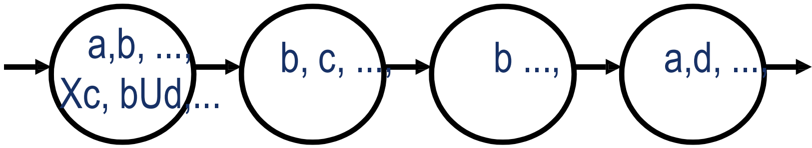
4. При любом решении задачи фермер когда-нибудь возвращается через реку один

$[\ F(fwgc) \Rightarrow F[(f\ \&X\neg f) \wedge (w\&Xw\ \vee \neg w\&X\neg w) \wedge (g\&Xg \vee \neg g\&X\neg g) \wedge (c\&Xc\ \vee \neg c\&X\neg c)]\ ]$

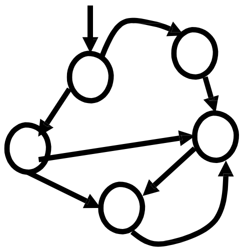
Заключение: TL – общие идеи



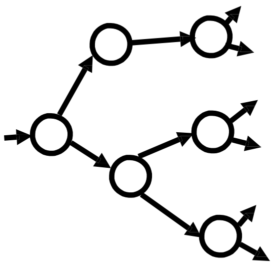
- Логика высказываний строится введением атомарных утверждений и базисных операторов $\{\vee, \neg\}$.
  По значениям истинности каждого атомарного утверждения можем вычислить истинность любой логической формулы
- В логике линейного времени LTL кроме атомарных утверждений и операций логики высказываний вводятся темпоральные операторы {U, X} (кроме них удобно использовать еще F и G)

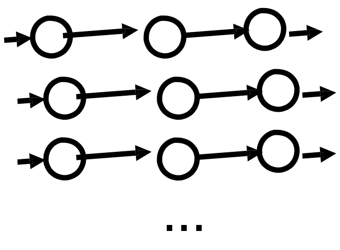


- По конкретной цепочке состояний (миров) в каждом состоянии можем вычислить истинностные значения любой формулы темпоральной логики LTL



- В логике ветвящегося времени CTL добавляются кванторы пути, позволяющие различать свойства различных путей
- В формулах CTL каждый темпоральный оператор предваряется квантором пути. Анализируются деревья поведений





- В логике LTL формула должна выполняться на всех путях. Анализируются все линейные поведения

Персоналии

- Создатель современной теории линейной темпоральной логики (LTL) 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 (reactive systems)
- Логика ветвящегося времени – Э.Кларк, А. Эмерсон (Университет Карнеги-Меллон, США) и многие другие

Спасибо за внимание