Структура Крипке. Темпоральная логика CTL

Данилов И. Г.
к.т.н, ассистент каф. МОП ЭВМ

Институт компьютерных технологий и информационной безопасности ЮФУ

2 октября 2015 г.

Начнем с упражнений

Запишите следующие утверждения с помощью нотации LTL:

1. “q встретится в будущем точно один раз”.
2. “всегда если запрос p будет подан, реакция q на него обязательно будет получена, а до ее получения запрос p не сбросится”.
3. “если событие q наступит, то до его наступления событие p не наступит”.
4. “предикаты p и q выполняются попеременно (т.е. после p не встретится p, пока не встретится q)”.

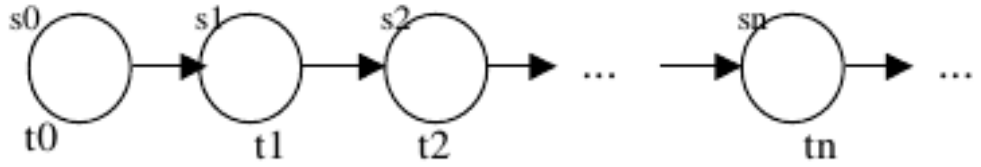
Материалы для занятия

Используемые материалы:

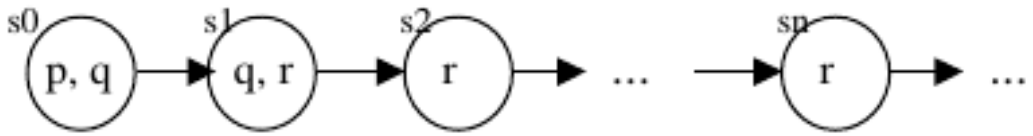
- Верификация параллельных и распределенных программных систем, лекция в Comp. Sci. Club 18.03.2012, Ю.Г. Карпов, И.В. Шошмина, А. Б. Беляев;
- Model Checking. Верификация параллельных и распределенных программных систем, Ю.Г. Карпов.

TL и анализ дискретных технических систем

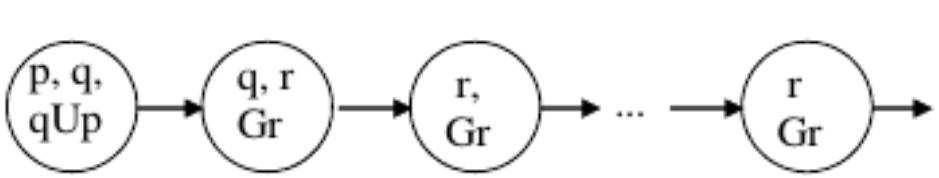
Последовательность "*миров*" в TL можно толковать как бесконечную последовательность состояний дискретной системы, а отношение достижимости – как дискретные переходы системы:



Атомарные предикаты - базисные свойства процесса в состояниях:



Производные темпоральные формулы в состояниях – это свойства вычисления в будущем, динамики процесса:



qUp – выполняется на вычислении

Gr – не выполняется на вычислении

Реагирующие системы

Реагирующие системы (reactive systems)

это класс информационных систем, основной функцией которых является поддержание взаимодействия с окружением, а не преобразование информации.

Вычисление и поведение реагирующей системы

Вычисление реагирующей системы — это бесконечная последовательность состояний, которые система проходит во времени. Поведение реагирующей системы — это все возможные ее вычисления.

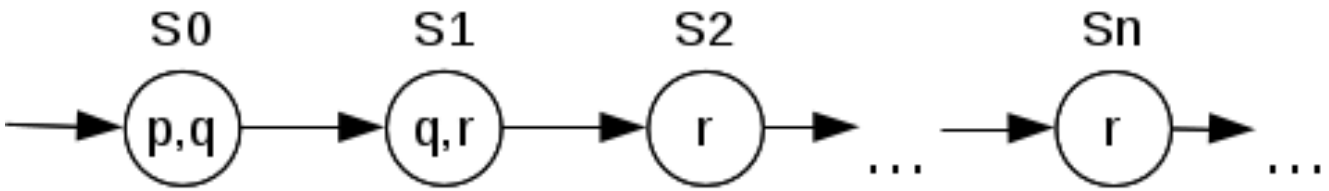
Математическая модель для исследования поведения реагирующих систем

По заданному вычислению — бесконечной цепочке состояний с определенным в каждом состоянии набором атомарных предикатов, истинных в этом состоянии, определяются значения истинности булевых и темпоральных формул.

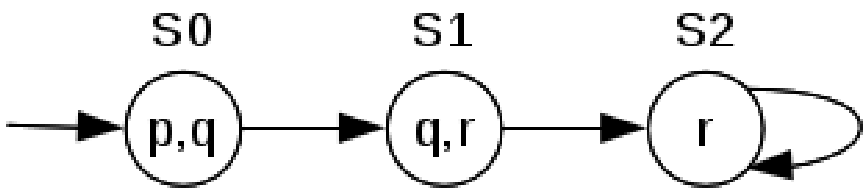
Как такие вычисления представить конечным образом?

Конечная система переходов

Даже если система переходов имеет конечное число состояний, в общем случае **количество возможных вычислений в ней бесконечно и каждое вычисление также бесконечно**.



Бесконечное вычисление.



Конечная система переходов.

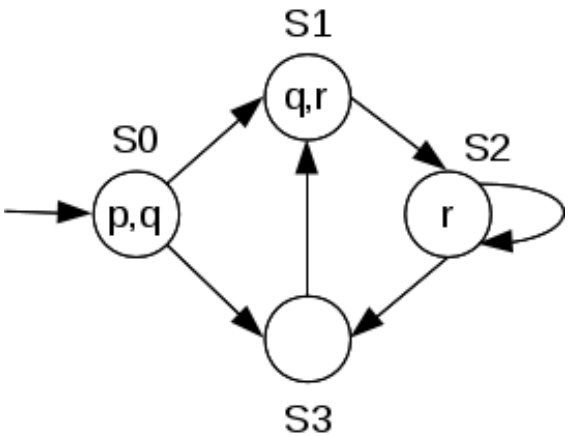
Альтернативные вычисления

Логика LTL: линейный мир — после сегодняшнего дня будет завтра, потом послезавтра ... линейная последовательность дискретных возрастающих значений.

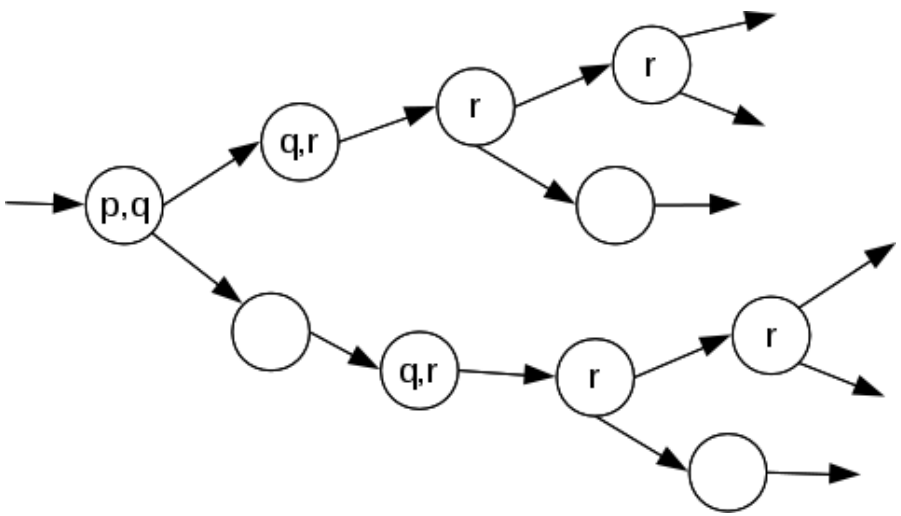
Поведения реальных систем (информационных в том числе) имеют альтернативы, выбор которых осуществляется на основе внешних событий, различий в содержании принятых сообщений и т.п.

Модель системы переходов с конечным числом состояний

Структура Крипке.



Развертка структуры Крипке.

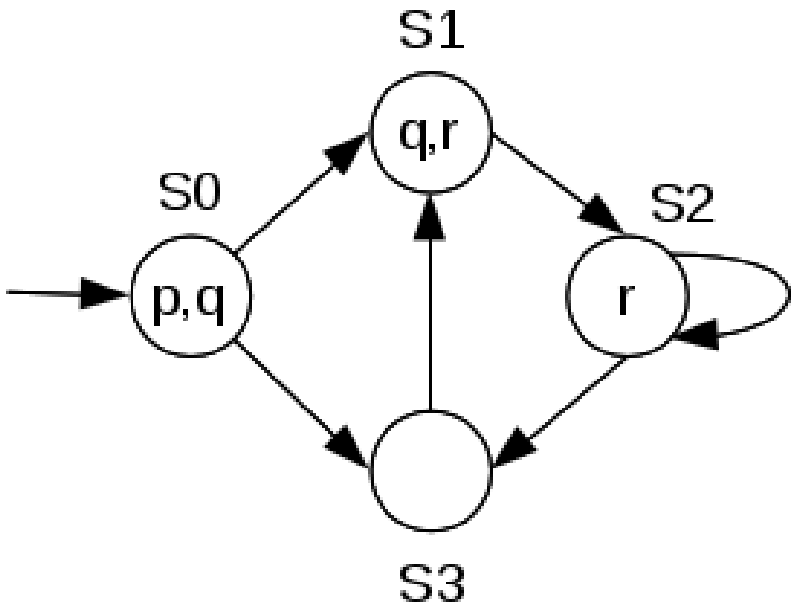


Структура Крипке

Структура Крипке M — это пятерка $M = (S, S_0, R, AP, L)$, где:

- S — конечное непустое множестов состояний;
- $S_0 \subseteq S$ — непустое множество начальных состояний;
- $R \subseteq S \times S$ — тотальное отношение на S, т.е. множество переходов, удовлетворяющих требованию: $(\forall s \in S)(\exists s' \in S)(s, s') \in R$;
- AP — конечное множество атомарных предикатов;
- $L : S \to 2^{AP}$ — функция пометок (каждому состоянию отображение L сопоставляет множество истинных в нем атомарных предикатов).

Структура Крипке. Пример



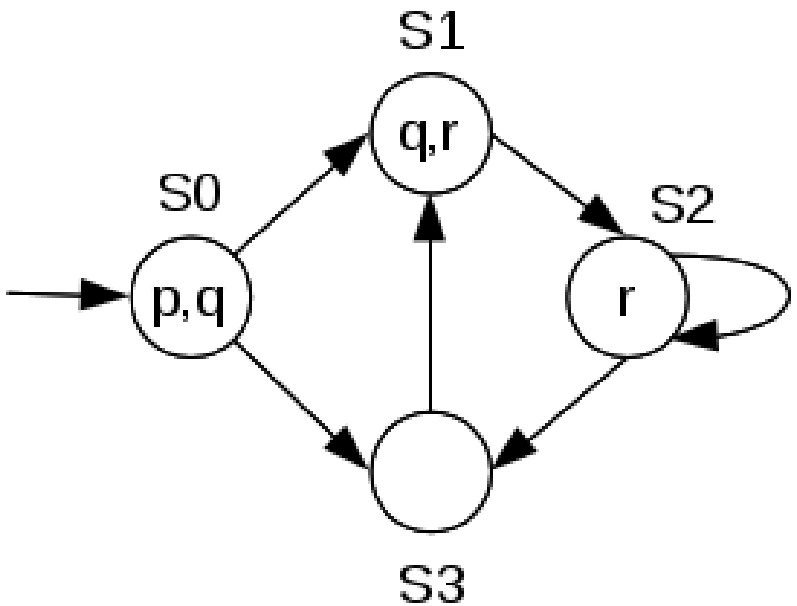
Здесь задана структура Крипке с четырьмя состояниями $\{s0, s1, s2, s3\}$, одним начальным состоянием $\{s0\}$, множеством атомарных предикатов $\{p, q, r\}$ и функцией пометок L : $L(s0) = \{p, q\}, L(s1) = \{q, r\}, L(s2) = \{r\}, L(s3) = \{\}$.

Вычисление структуры Крипке

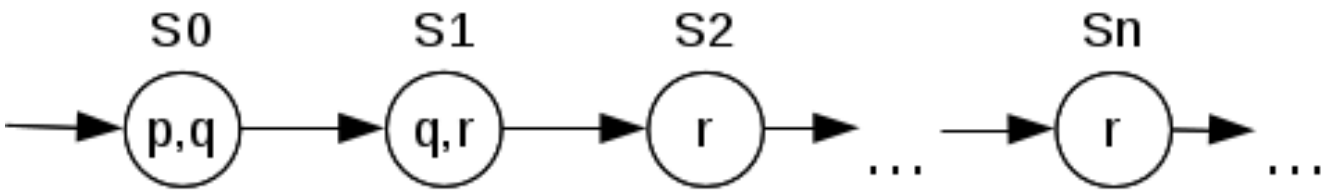
Вычисление структуры Крипке

Вычислением структуры Крипке M называется любая бесконечная цепочка $\sigma = q_0 q_1 q_2 q_3 \ldots$, такая что $q_0 \in S_0$ и $(q_i, q_{i+1}) \in R$. Формально вычисление структуры Крипке M можно представить как отображение $\sigma : N \to S$, где N — множество натуральных чисел, т.е. $\sigma(i)$ — это некоторое состояние структуры Крипке.

Вычисление структуры Крипке. Пример



Вычисление: $s0, s1, s2, s2...$



Траектория (трасса) структуры Крипке

Траектория структуры Крипке

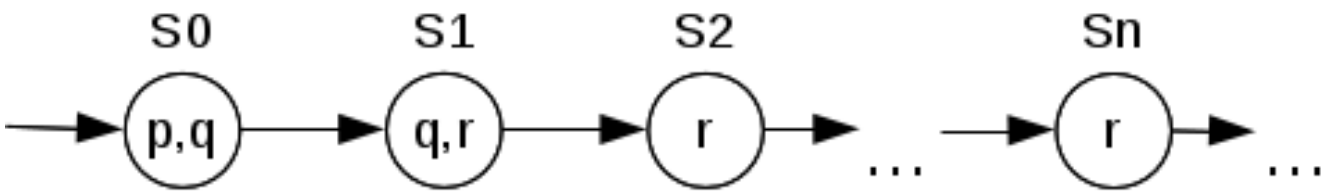
Траекторией структуры Крипке M, индуцированной вычислением $\sigma = q_0 q_1 q_2 q_3 ...$, называется бесконечная цепочка $L(\sigma) = L(q_0)L(q_1)L(q_2)L(q_3)...$, т.е. бесконечная цепочка подмножеств атомарных предикатов, истинных в соответствующих состояниях вычисления σ. Формально траектория — это отображение натурального ряда в множество 2^{AP}.

Траектория: **наблюдаемые свойства**, которые могут выполниться в анализируемых системах.

Формулы темпоральной логики описывают **свойства траекторий**.

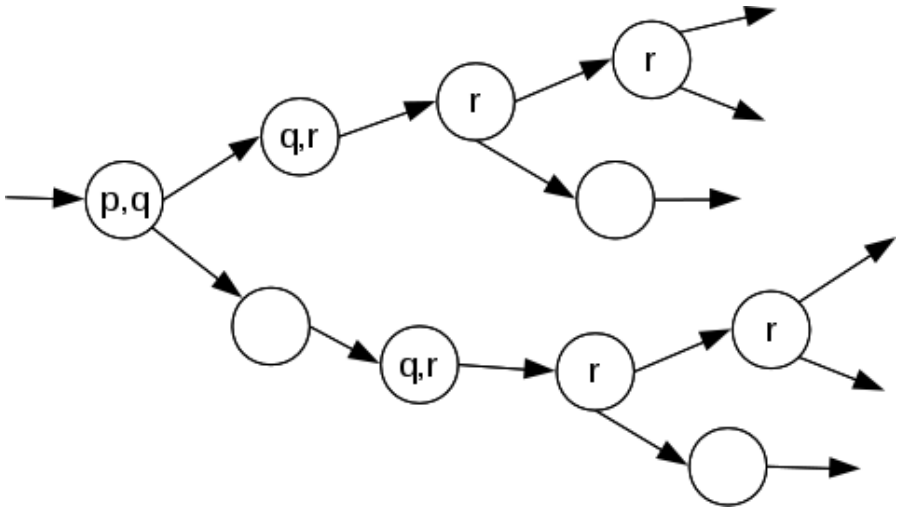
Траектория структуры Крипке. Пример

Вычисление: $s0, s1, s2, s2...$



Траектория: $\{p, q\}, \{q, r\}, \{r\}, \{r\}...$

ω-слова и языки



Вычисления индуцируют следующие траектории (ω-слова):

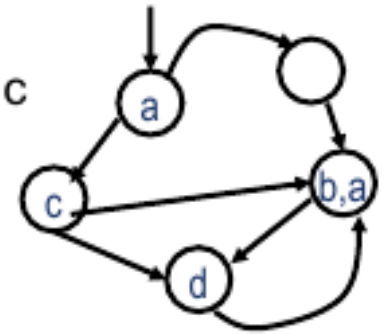
- $\pi_1 = \{p, q\}\{q, r\}\{r\}\{\}...$;
- $\pi_2 = \{p, q\}\{q, r\}\{r\}\{r\}...$;
- $\pi_3 = \{p, q\}\{\}\{q, r\}\{r\}\{\}...$;
- $\pi_4 = \{p, q\}\{\}\{q, r\}\{r\}\{r\}...$

Множество всех ω-слов структуры Крипке M называется ω-языком, допускаемым M.

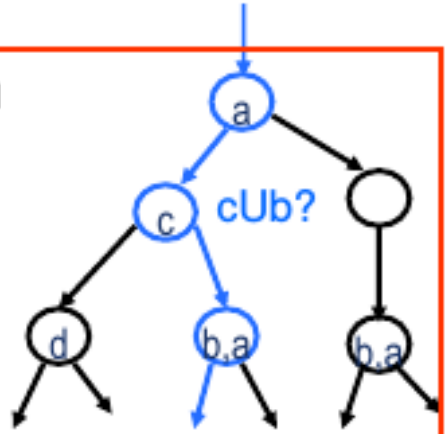
Линейное и ветвящееся время

Как бесконечные вычисления задать конечным образом?

Структура Крипке – система переходов с помеченными состояниями и непомеченными переходами



Развертка структуры Крипке определяет бесконечные цепочки состояний - возможные ВЫЧИСЛЕНИЯ



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

Но как понимать формулы LTL: Gp, pUq, … в состоянии? Для какого пути?

Ввести "*кванторы пути*"

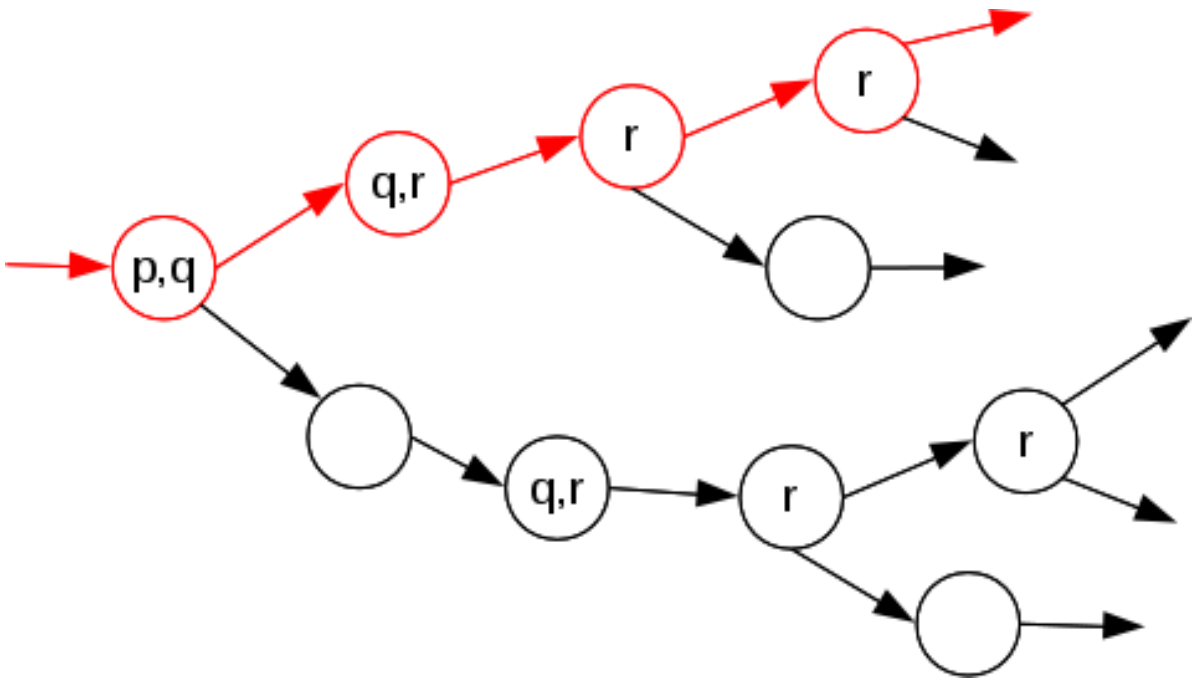
E φ ≡ "*существует путь из данного состояния, на котором формула пути* ϕ *истинна*" (**Exists**)

A φ ≡ "*для всех путей из данного состояния формула пути* ϕ *истинна*" (**All**)

Очевидно, **A** φ ≡ ¬**E** ¬φ

LTL — формулы пути

XGr будет истинна не для всех путей, но если ввести *E*, то *EXGr* — будет выполняться.



Темпоральные логики LTL и CTL

В LTL (Linear Temporal Logic) – темпоральной логике линейного времени, формулы пути предваряются квантором пути A: они должны выполняться для всех вычислений структуры Крипке

В CTL (Computational Tree Logic), в темпоральной логике ветвящегося времени, каждый темпоральный оператор предваряется квантором пути A или E

Формулы LTL:

$\mathbf{AG}(p \Rightarrow \mathbf{F}\, q)$

$\mathbf{A}\,(\neg a \vee \mathbf{G} b \;\&\; (a \mathbf{U} \neg c))$

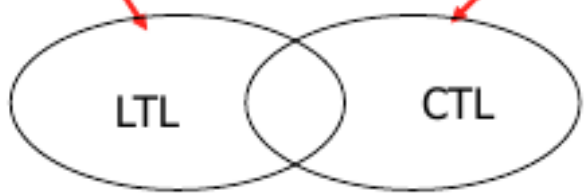
$\mathbf{A}\,(a \mathbf{U} \neg b)$

Формулы CTL:

$\mathbf{AG}(p \& \neg \mathbf{EF}(q \Rightarrow r))$

$\mathbf{EF}(a \;\&\; \mathbf{E}(a \mathbf{U} \neg c))$

$\mathbf{A}\,(a \mathbf{U} \neg b)$

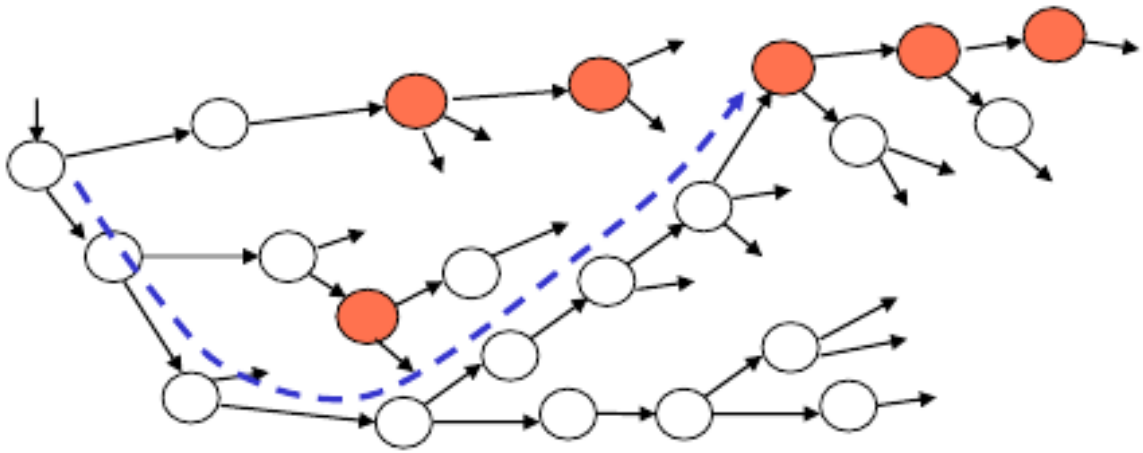


Примеры спецификации требований в CTL

- AGAF Restart
  - из любого состояния при любом функционировании системы обязательно вернемся в состояние рестарта
- ¬EF(int >0.01)
  - не существует такого режима работы, при котором интенсивность облучения пациента превысит 0.01 радиан в сек
- AG (req ⇒ A (req U ack))
  - во всех режимах после того, как запрос req установится, он никогда не будет снят, пока на него не придет подтверждение
- E[p U A [q U r]]
  - существует режим, в котором условие p будет истинным с начала вычисления до тех пор, пока q не станет непрерывно активно до выполнения r

Пример свойства, выраженного формулой логики CTL

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

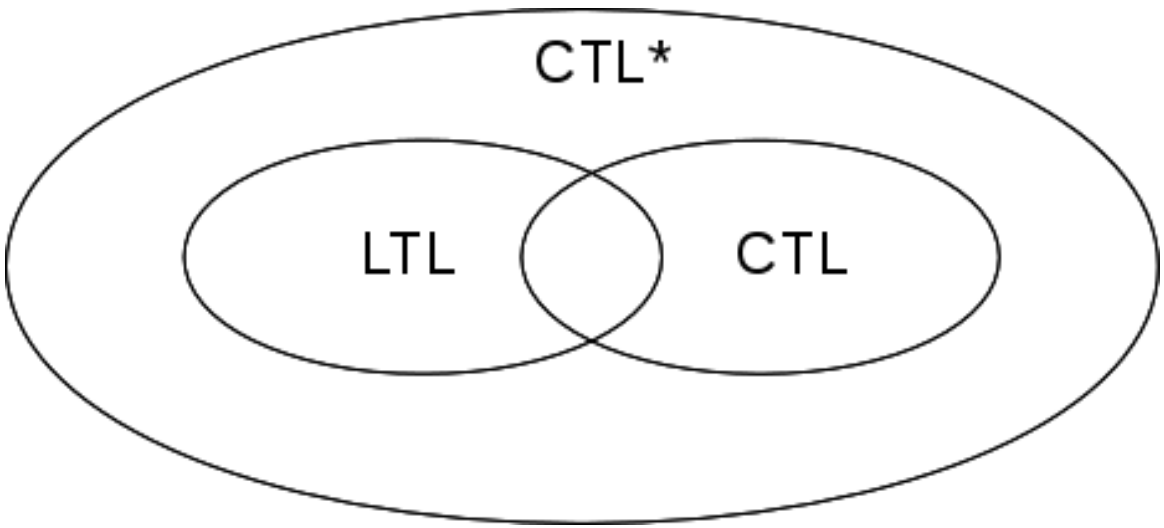


AG EF EG '*истинная вера* '

Всегда, куда бы мы ни попали в нашей жизни (AG), существует такой путь, что на нем в конце концов обязательно попадем (EF) в состояние, с которого идет путь (EG) 'истинной веры'

LTL, CTL, CTL*

CTL* — расширенная логика темпорального времени. Вводится квантор E, который применяется без ограничений.



Набор упражнений 3

Какие из следующих формул являются CTL-формулами? Какие из них LTL-формулы?

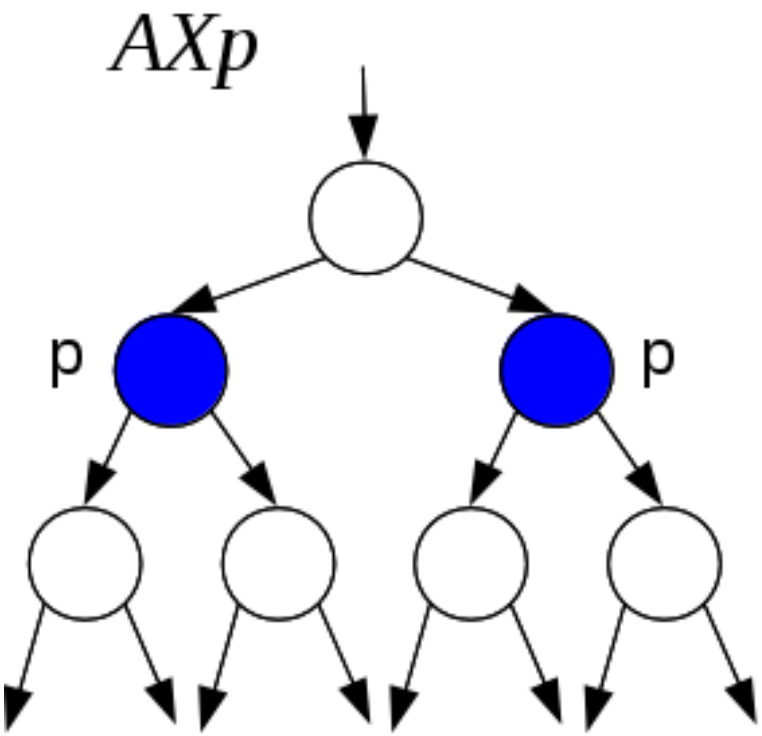
1. $p \vee \neg q$
2. $AF(p \vee AGX\neg q)$
3. $A(p \cup (AXq))$
4. $EGAF(p \cup q)$
5. $EF(p \wedge \neg AGq)$
6. $A(p \vee Xq \vee XX(p \cup \neg q))$

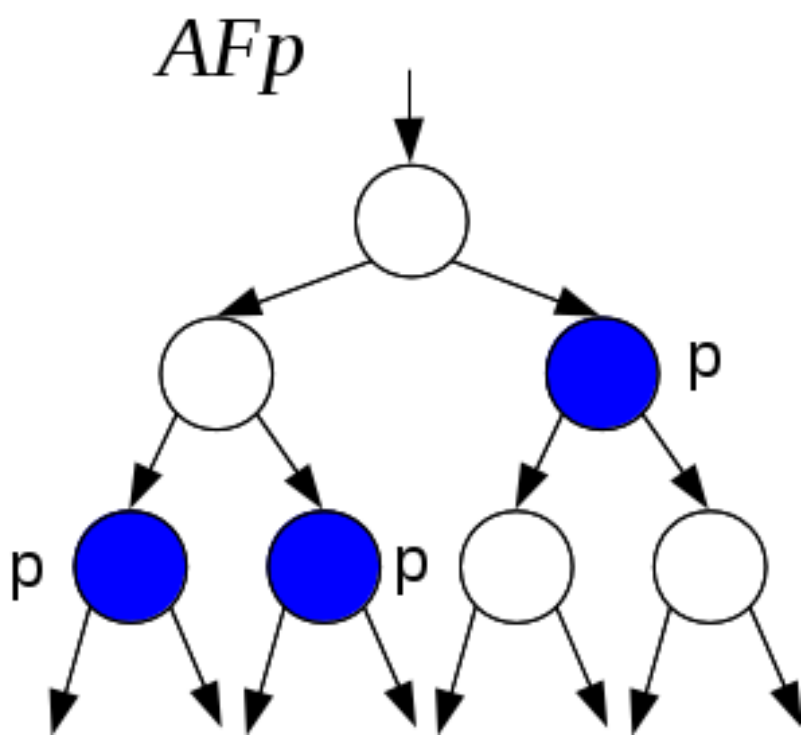
Пример

Пусть p означает “Я люблю Машу”, q означает “Я люблю Дашу”. Тогда:

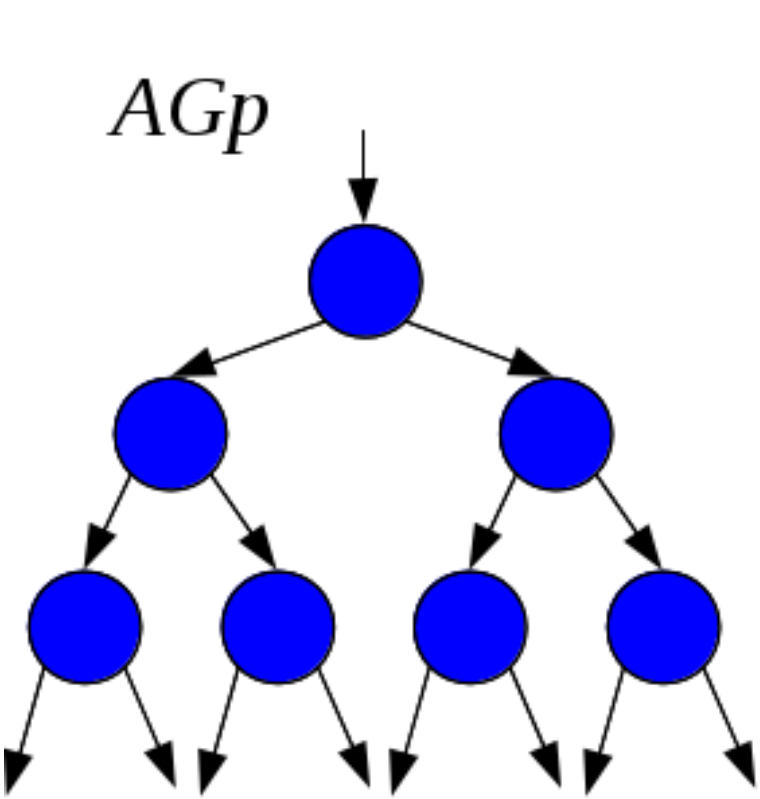
1. формула AGp является формулой CTL, она формализует следующее высказывание: “Я люблю Машу, и, что бы ни случилось, я буду любить ее всегда”;
2. формула $AFG(p \wedge \neg q)$ является формулой LTL, она формализует следующее высказывание “Что бы ни случилось, в будущем я полюблю Машу навсегда, а Дашу не буду любить!”;
3. формула $E(p \cup (AGq))$ является формулой CTL, она формализует следующее высказывание “Я не исключаю такого развития событий (E), что я буду любить Машу до тех пор (U), пока, наконец я не полюблю Дашу навечно (AG)”.

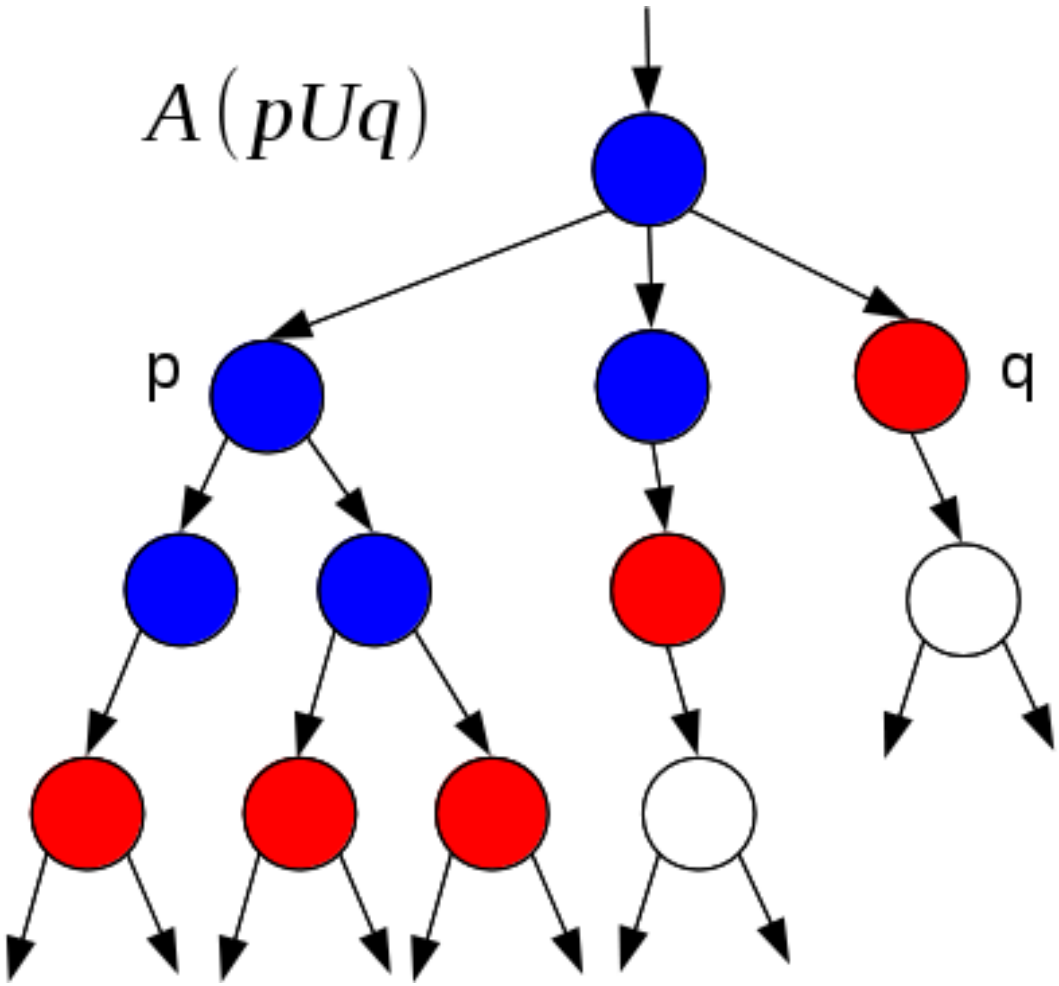
Визуальная семантика формул CTL



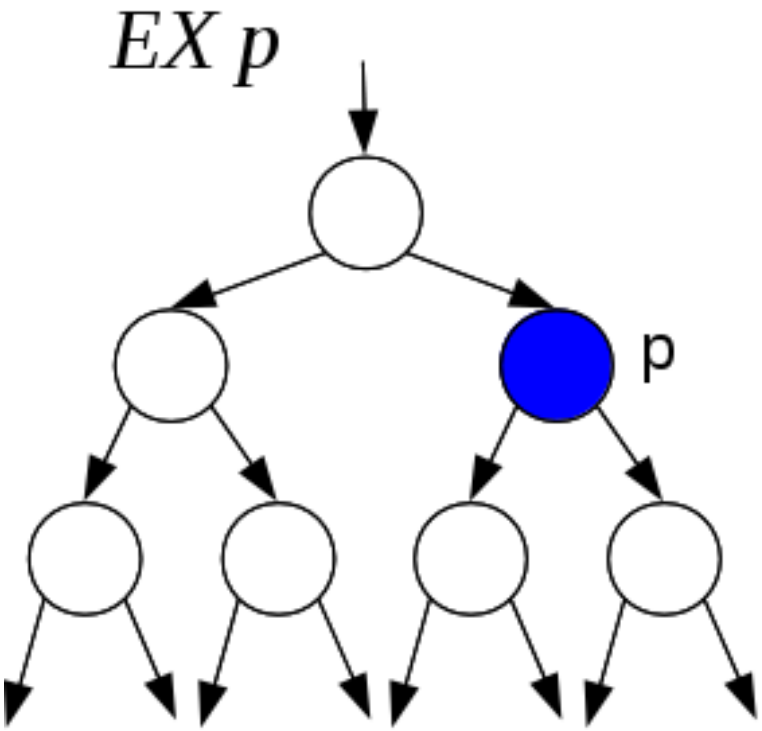


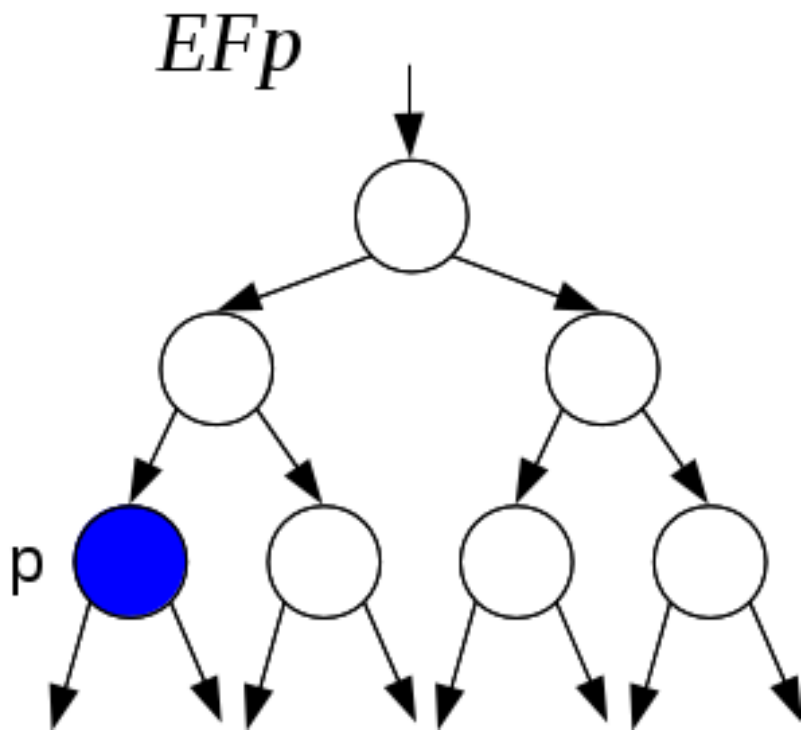
Визуальная семантика формул CTL



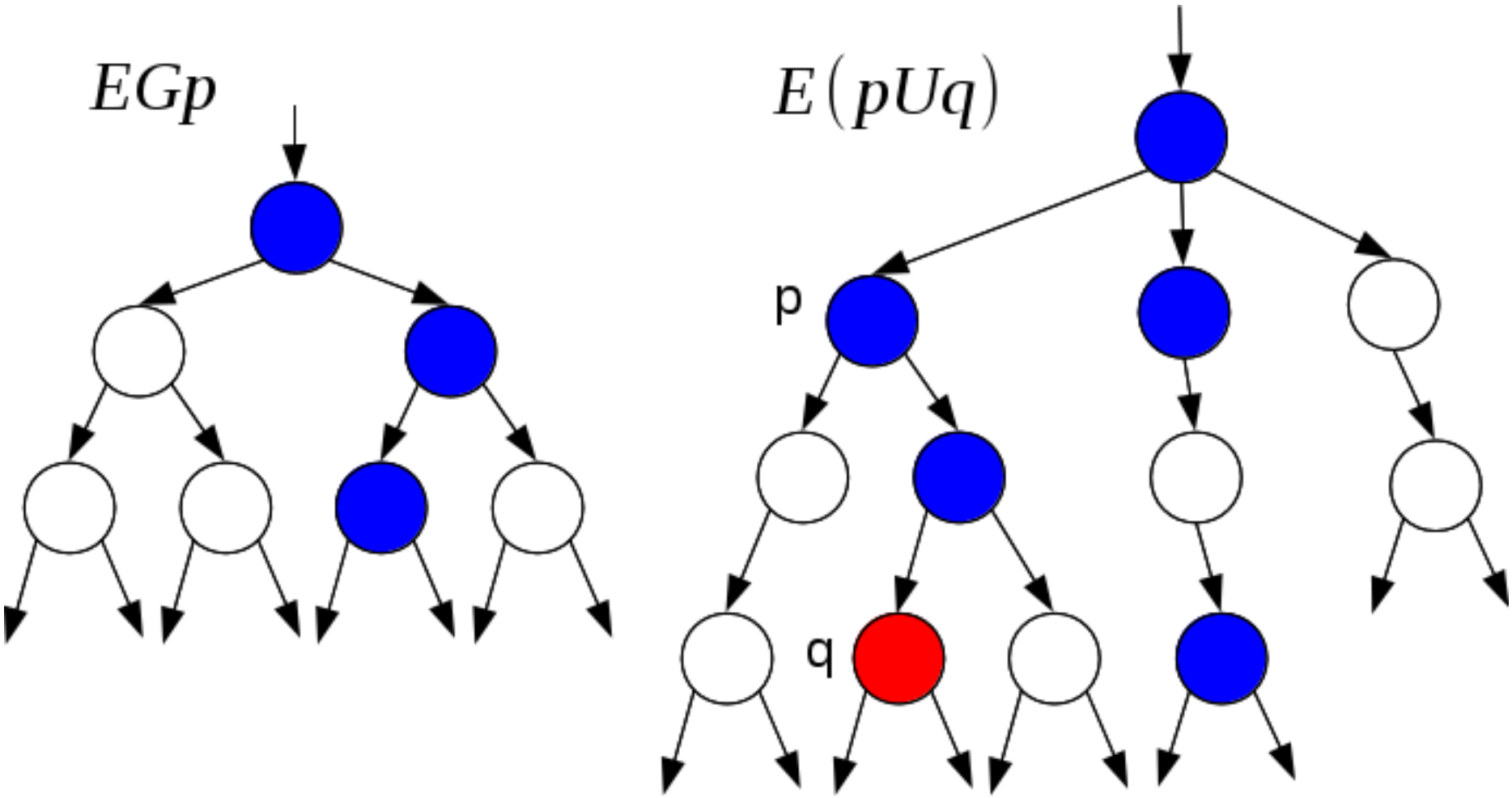


Визуальная семантика формул CTL





Визуальная семантика формул CTL



Набор упражнений 4

Пусть p означает “Я люблю Машу”, q означает “Я люблю Дашу”. Каким высказываниям соответствуют следующие формулы CTL:

1. $AFEGp$
2. $EFAGp$
3. $A(p \cup q)$
4. $E((EXp) \cup (AGq))$

Сравнение логик CTL и LTL

Формулы этих логик характеризуют свойства разных типов:

- формулы LTL являются формулами пути; в логике LTL вообще нет кванторов пути;
- формулы CTL являются формулами состояний; в логике CTL любой темпоральный оператор предварен квантором пути.

Сравнение логик CTL и LTL

Формулы обеих логик интерпретируются (принимают истину или ложь) на структурах Крипке, но:

- формулы логики LTL интерпретируются на всех вычислениях структуры Крипке, начинающихся в начальном состоянии (большие трудности при проверке);
- формулы логики CTL интерпретируются на деревьях вычислений структуры Крипке (конечное число состояний).

Сравнение логик CTL и LTL

Техника верификации для LTL и CTL совершенно различна как по применяемым алгоритмам, так и по их сложности:

- для LTL — Spin. Сложность алгоритмов верификации линейна относительно числа состояний стр-ры Крипке и экспоненциальна относительно сложности (числа подформул) формулы LTL (но на практике они малы);
- для CTL — SMV. Сложность алгоритмов верификации пропорциональна числу состояний структуры Крипке и сложности (числу подформул) формулы CTL.

Сравнение логик CTL и LTL

Многие свойства технических систем могут быть выражены как формулами логики CTL, так и формулами логики LTL. Например, “каждое сообщение когда-нибудь в будущем будет подтверждено”:

- для LTL — $G(send \Rightarrow Freq)$;
- для CTL — $AG(send \Rightarrow AFreq)$.

Однако **неверно** утверждение, что если в формуле LTL каждый темпоральный оператор будет предварен квантором пути A, то построенная так формула CTL будет эквивалентна исходной формуле.

Сравнение логик CTL и LTL

Выразительная мощность этих двух логик несравнима, т.е. некоторые свойства могут быть выражены в CTL, но не могут быть выражены в LTL и наоборот:

- например, формула $AGEF\phi$ логики CTL, выражающая "достижимость свойства ϕ", не выразима в LTL;
- например, формула $FG\phi$ логики LTL, выражающая "стабилизацию свойства ϕ когда-нибудь в будущем", не выразима в CTL.

Набор упражнений 5

Известно, что если в формуле Φ LTL каждый темпоральный оператор будет предварен квантором пути A, то такая формула CTL не всегда будет эквивалентна исходной формуле Φ.

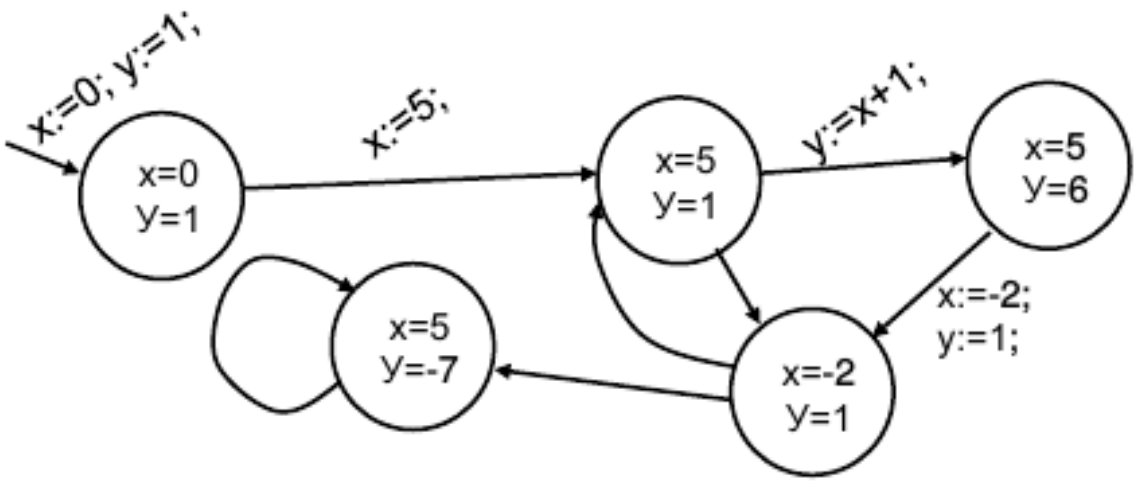
1. Проверьте, выполняются ли LTL-формула FGp и CTL-формула $AFAGp$ на структуре Крипке $M1 = (S, S_0, R, AP, L)$, где: $S = \{s_0, s_1, s_2\}$, $S_0 = \{s_0\}$, $R = \{(s_0, s_0), (s_0, s_1), (s_1, s_2), (s_2, s_2)\}$, $AP = \{p\}$, $L(s_0) = L(s_2) = \{p\}$, $L(s_1) = \emptyset$.
2. Проверьте, выполняются ли LTL-формула GFp и CTL-формула $AGAFp$ на структуре Крипке $M2 = (S, S_0, R, AP, L)$, где: $S = \{s_0, s_1\}$, $S_0 = \{s_0\}$, $R = \{(s_0, s_0), (s_0, s_1), (s_1, s_1)\}$, $AP = \{p\}$, $L(s_0) = \emptyset$, $L(s_1) = \{p\}$.

Структура Крипке как модель программы

```
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
```

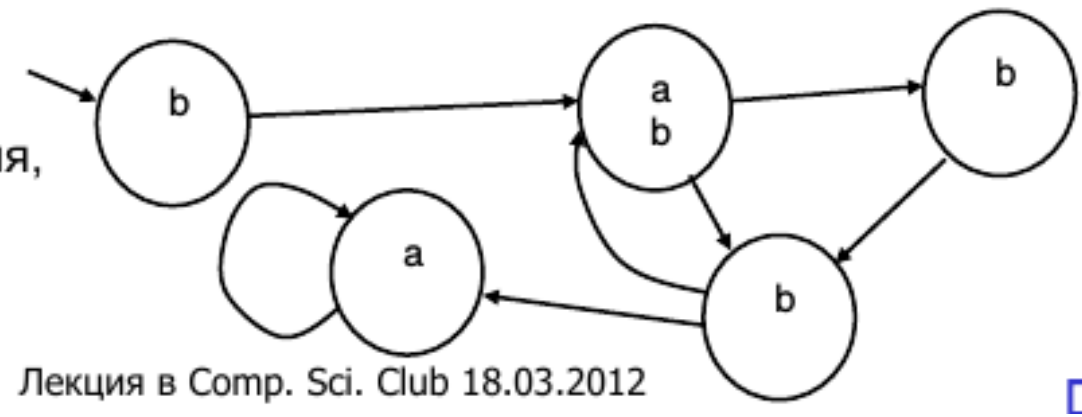
Состояние программы – вектор значений ее переменных И метки (pc)

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

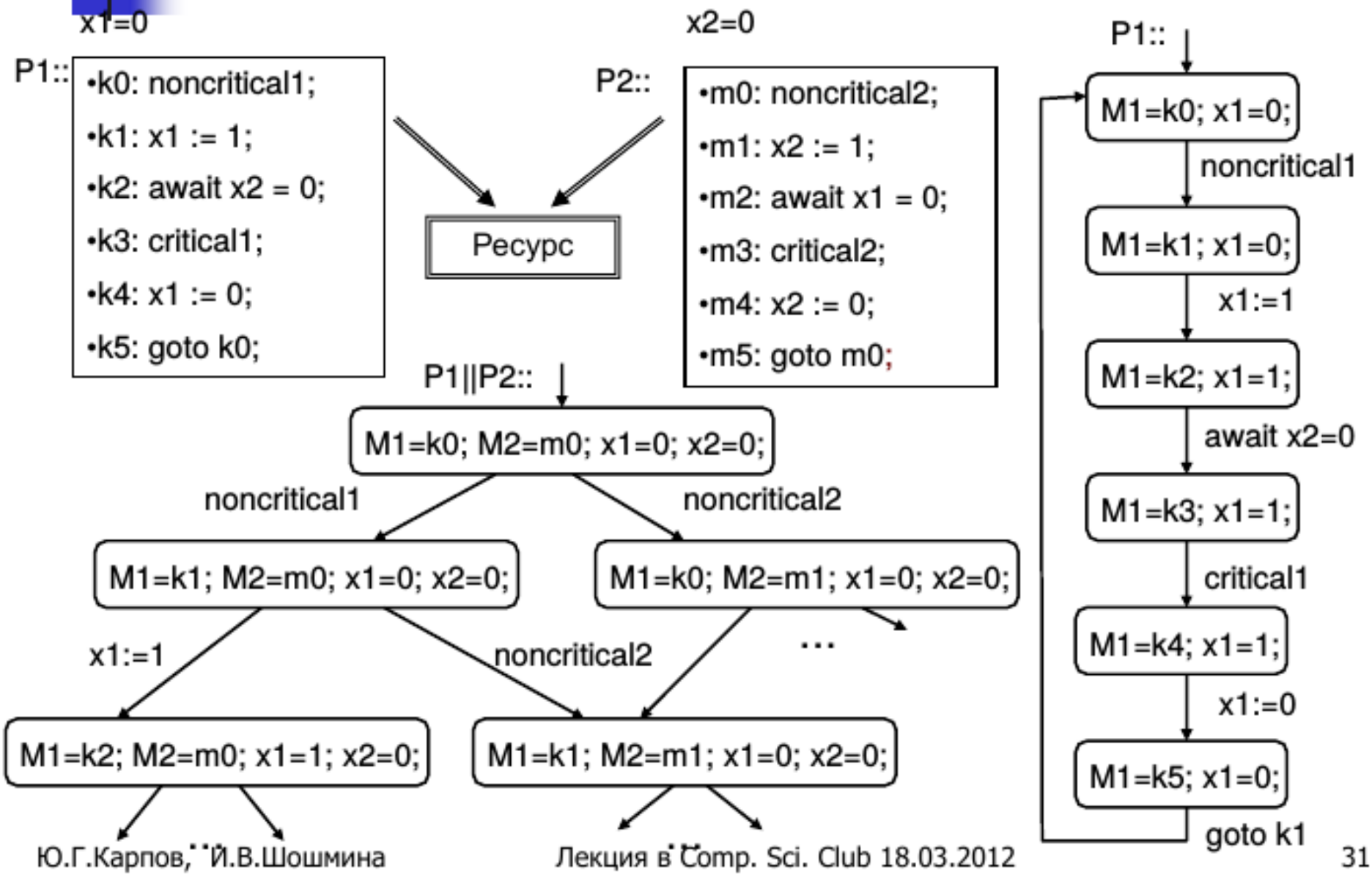


Пусть атомарные утверждения, *ИНТЕРЕСУЮЩИЕ НАС:*

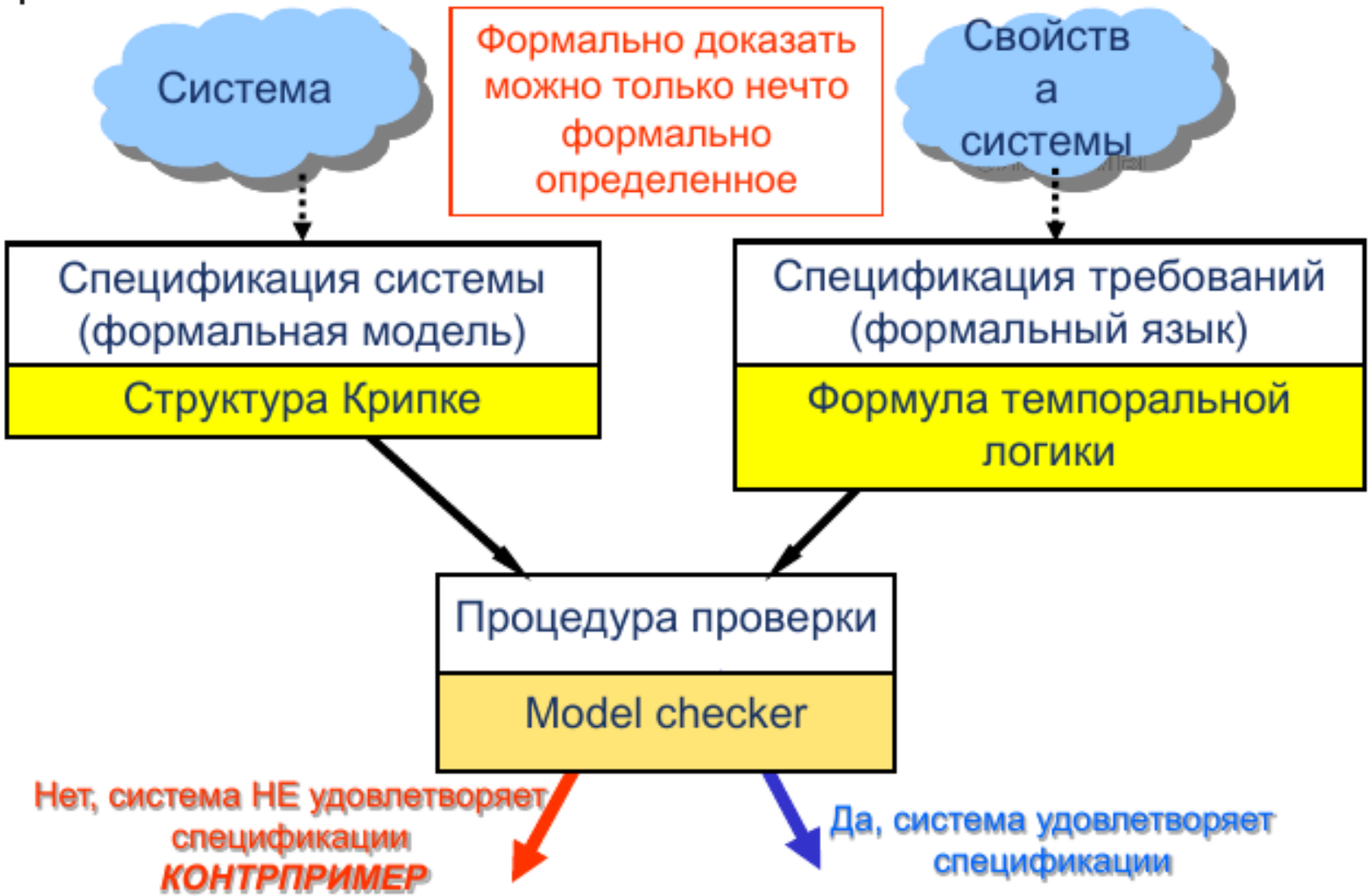
a=x>y; b=|x+y|<3



Структура Крипке - модель параллельной программы



Общая схема верификации Model checking



Q&A

Вопросы?