

НАДЁЖНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

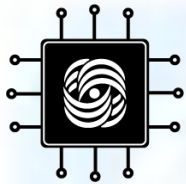
Лекция 5: *Логика линейного времени(LTL)*

ВМиК МГУ им. М.В. Ломоносова,
Кафедра АСВК, Лаборатория Вычислительных Комплексов
Ассистент Волканов Д.Ю.



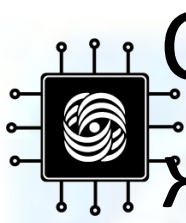
План лекции

- Свойства безопасности и живучести
- Логика линейного времени (LTL)
- Свойства, инвариантные к прореживанию
- Практические приёмы формулирования свойств на LTL



Рассуждения о правильности программы

- Требования правильности задаются как утверждения о возможных или невозможных вариантах выполнения модели,
 - рассуждения о вероятности не допускаются из соображений строгости доказательства;
- Мы утверждаем, что некоторые варианты выполнения модели
 - либо **невозможны**,
 - либо **неизбежны**;
- Двойственность утверждений:
 - если что-то неизбежно, то противоположное – невозможно,
 - если что-то невозможно, то противоположное – неизбежно,
 - используя логику, можно переходить от одного к другому при помощи логического отрицания.



Свойства безопасности и живучести

(автор – Leslie Lamport)

безопасность

• «ничего плохого никогда не произойдёт»;

- пример: инвариант системы
 - x всегда меньше y ;
- задача верификатора – найти вычисления, которые ведут к нарушению свойства безопасности (то «плохое», которое никогда не должно произойти)

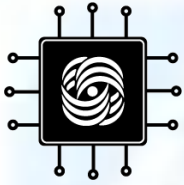


живучесть

• «рано или поздно произойдёт что-то хорошее»;

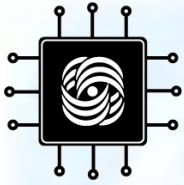
- пример: «отзывчивость»
 - если отправлен запрос, то рано или поздно будет сгенерирован ответ;
- задача верификатора – найти вычисления, в которых это «хорошее» может откладываться до бесконечности





Немного подробнее...

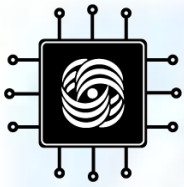
- Свойства безопасности – самые простые свойства («состояние, где истинно φ , недостижимо»).
[Достаточно исследовать всё (конечное)]
- Можно ли в рамках свойства безопасности сформулировать, что **φ неизбежно?**
- **«состояние, где истинно $!\varphi$, недостижимо»**
– не то, это слишком сильное свойство;
- **«состояние, где истинно φ , достижимо»**
– а это, наоборот, слишком слабое.



Немного подробнее...

- **« φ неизбежно»** – означает, что φ **обязательно достижимо**;
- для спецификации такой модальности и нужны свойства живучести.

Необходимо показать, что свойство достижимо в любом начальном вычислении системы. В частности, нужно исследовать циклы в системе переходов.

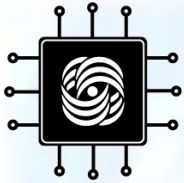


Классическая логика не может описать утверждения

- Классическая логика
 - Прimitивная 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 о событиях обычно истинны в один момент времени и ложны в другой!

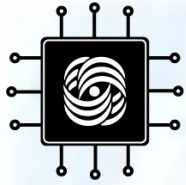
Утверждения о событиях, происходящих в разные моменты времени, нельзя корректно выразить в обычной логике



Темпоральная логика

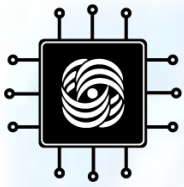
- **Определение** TL - это любая логическая система, которая позволяет, не вводя явно понятие времени, формализовать утверждения, истинность которых изменяется со временем. Используется для описания последовательностей явлений и их взаимосвязи и зависимости во времени
- **Применения TL**
 - **ФИЛОСОФИЯ:** 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);
 - **ТЕХНИКА:** для формализации утверждений о свойствах **будущего поведения** технических систем (программ, оборудования, систем управления, ...), для описания свойств их поведения во времени после того, как они будут запущены

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



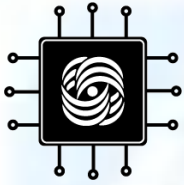
Примеры темпоральных свойств

- **p всегда** истинно;
- **p рано или поздно** станет всегда ложным;
- **p всегда рано или поздно** станет ложным хотя бы ещё один раз;
- **p всегда** ведёт к $\neg q$;
- **p всегда** ведёт к тому, что **рано или поздно** станет истинным q .



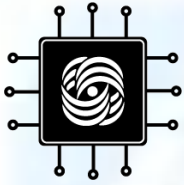
Темпоральные свойства на естественном языке?

- нет строгой семантики => возможно множество трактовок
 - в части области проверки: «индикатор не горит» - в начальном состоянии? или это инвариант?
 - в темпоральной части:
 - попробуйте объяснить разницу: «от события А до события Б» и «между событиями А и Б»
 - «деньги выплачиваются, как только работа будет выполнена» -- требуется ли выполнение работы?
- значение зависит от контекста:
 - если нажата кнопка, рано или поздно будет выпущено шасси
 - от взлёта и до посадки, если нажата кнопка, то рано или поздно будет выпущено шасси (область проверки вложенного свойства изменилась)
 - после взлёта, если была нажата кнопка, то до посадки будет выпущено шасси (более строгая формулировка)
- зависит от знания и понимания естественного языка, который сложнее LTL.



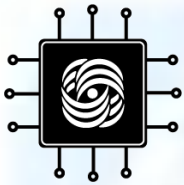
Темпоральная логика LTL

- Ясный, лаконичный и непротиворечивый способ описания требований к программам;
- В явном виде время не присутствует, однако рассуждения ведутся в терминах «никогда», «всегда», «рано или поздно», которые представлены в виде темпоральных операторов.
- Мы рассматриваем темпоральную логику линейного времени – LTL. С её помощью можно описывать свойства, которым должны удовлетворять линейные последовательности наблюдаемых состояний – трассы.
- LTL предложена Амиром Пнуэли (Amir Pnueli) в конце 70-х.



Формулы LTL

- Могут использоваться для описания как свойств живучести, так и безопасности
- LTL = пропозициональная логика
(свойства состояний)
- + темпоральные операторы
(последовательности состояний)
- Формула LTL $f ::=$
 - $p, q, \dots$ - свойства состояний, включая true и false,
 - (f) – группировка при помощи скобок,
 - αf – унарные операторы,
 - $f_1 \beta f_2$ – бинарные операторы.



Операторы LTL

- Унарные:

- $\Box([])$ всегда (в будущем),
- $\Diamond(<>)$ рано или поздно,
- (X) в следующем состоянии,
- $\neg (!)$ логическое отрицание;

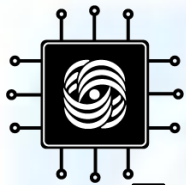
*когда-нибудь,
в конце концов*

- Бинарные:

- $U (U)$ пока, *до тех пор*
- $\wedge (\&\&)$ логическое И,
- $\vee (||)$ логическое ИЛИ,
- $\rightarrow (->)$ логическая импликация,
- $\leftrightarrow (<->)$ логическая эквивалентность.

$$\equiv (\neg p \vee q)$$

$$\equiv (p \rightarrow q) \wedge (q \rightarrow p)$$



Выполнимость формул

- Последовательность состояний прохода σ

$S_0, S_1, S_2, S_3, \dots$

- Набор пропозициональных формул p, q :

$\forall i, i \geq 0$, и $\forall p$, определено $s_i \models p$

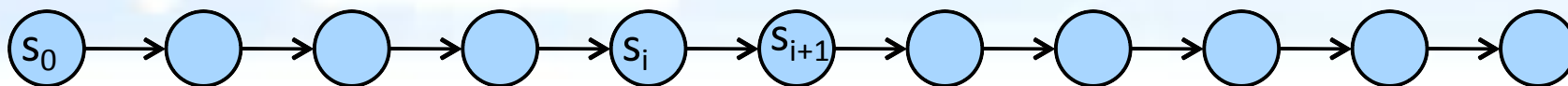
- Семантика LTL:

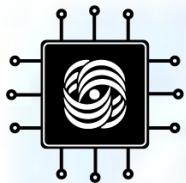
$$\sigma \models f \iff s_0 \models f$$

$$s_i \models []f \iff \forall j, j \geq i : s_j \models f$$

$$s_i \models \Diamond f \iff \exists j, j \geq i : s_j \models f$$

$$s_i \models Xf \iff s_{i+1} \models f$$





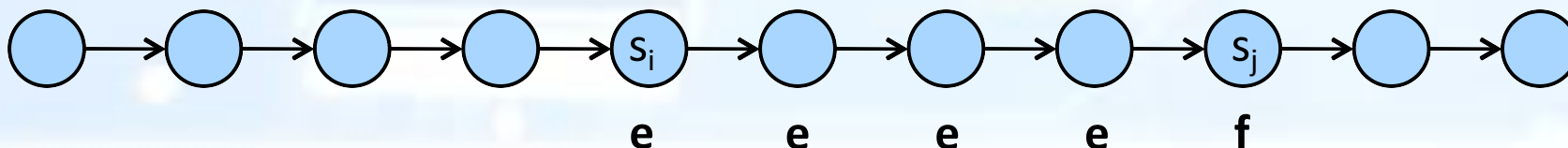
Слабый и сильный until

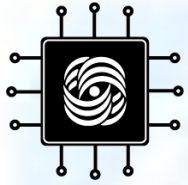
слабый:

$$s_i \models eWf \iff s_i \models f \vee (s_i \models e \wedge s_{i+1} \models (eWf))$$

сильный:
(Spin)

$$s_i \models eUf \iff \begin{array}{l} \exists j, j \geq i : s_j \models f \quad u \\ \forall k, i \leq k < j : s_k \models e \end{array}$$





Практически важные следствия

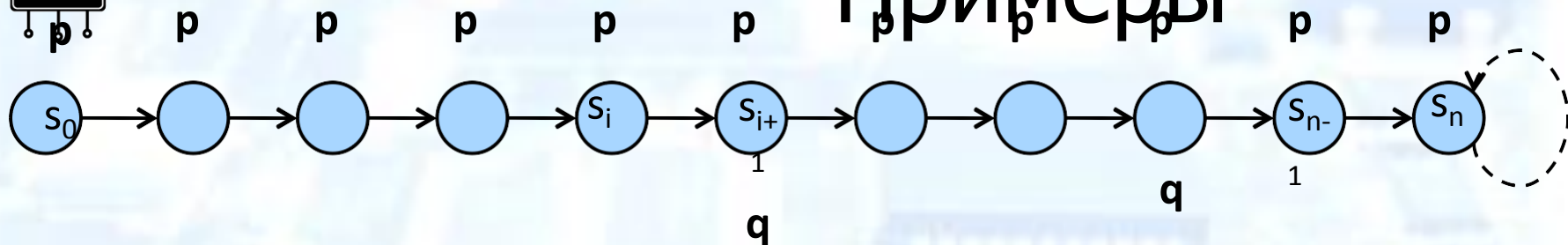
$$\sigma \models e \text{ } W \text{ } false \iff \sigma \models []e$$

$$\sigma \models true \text{ } U \text{ } f \iff \sigma \models \Diamond f$$

$$\sigma \models e \text{ } W \text{ } f \iff \sigma \models []e \vee e U f$$



Примеры



$$\sigma \models []p$$

$$\sigma \models \Diamond p$$

$$\sigma \models []\Diamond p$$

$$\sigma \models []q$$

$$\sigma \models \Diamond q$$

$$\sigma \models []\Diamond q$$

$$\sigma \models pUq$$

$$\sigma \models [](pUq)$$

$$\sigma \models [](pWq)$$

$$\sigma \models qUp$$

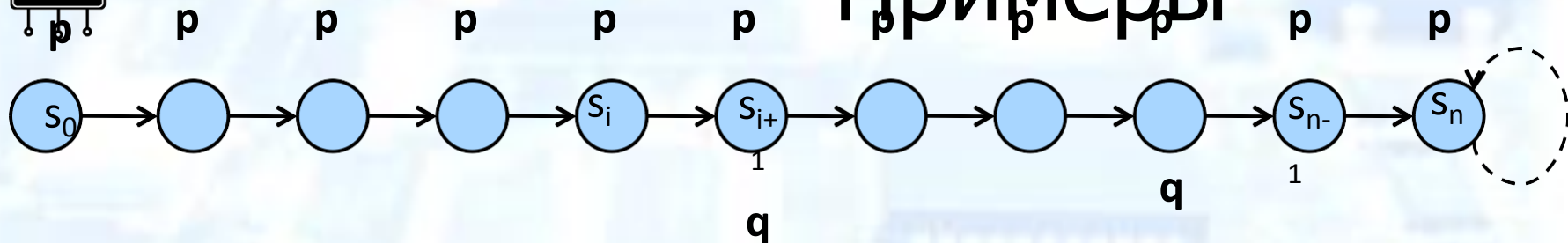
$$\sigma \models [](qUp)$$

$$\sigma \models qWp$$

$$\sigma \models []qWp$$



Примеры



$$\sigma \models []p$$

$$\sigma \models \Diamond p$$

$$\sigma \models []\Diamond p$$

$$\sigma \models []q$$

$$\sigma \models \Diamond q$$

$$\sigma \models []\Diamond q$$

$$\sigma \models pUq$$

$$\sigma \models [](pUq)$$

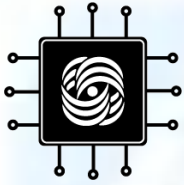
$$\sigma \models [](pWq)$$

$$\sigma \models qUp$$

$$\sigma \models [](qUp)$$

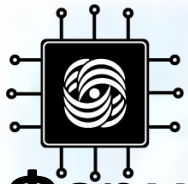
$$\sigma \models qWp$$

$$\sigma \models []qWp$$



Цикличность и стабильность

- Свойством цикличности называется любая темпоральная формула, которая представима в виде $\Box\Diamond p$, где p – формула на состоянии;
- Свойством стабильности называется любая темпоральная формула, которая представима в виде $\Diamond\Box p$, где p – формула на состоянии.



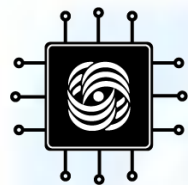
Распространённые LTL-формулы

Формула

Описание

Тип

$\Box p$	всегда p	<i>инвариант</i>
$\langle \rangle p$	рано или поздно p	<i>гарантия</i>
$p \rightarrow \langle \rangle q$	если p , то рано или поздно q	<i>отклик</i>
$p \rightarrow qUr$	если p , то q , затем r	<i>приоритет</i>
$\Box \langle \rangle p$	всегда рано или поздно будет p	<i>цикличность (прогресс)</i>
$\langle \rangle \Box p$	рано или поздно всегда будет p	<i>стабильность (бездействие)</i>
$\langle \rangle p \rightarrow \langle \rangle q$	если рано или поздно p , то рано или поздно q	<i>корреляция</i>



Эквивалентные преобразования

$$![]p \Leftrightarrow \Diamond!p$$

$$!\Diamond p \Leftrightarrow []!p$$

$$!(pUq) \Leftrightarrow !qW(!p \wedge !q)$$

$$!(pWq) \Leftrightarrow !qU(!p \wedge !q)$$

$$[](p \wedge q) \Leftrightarrow []p \wedge []q$$

$$\Diamond(p \wedge q) \Leftrightarrow \Diamond p \wedge \Diamond q$$

$$pW(q \vee r) \Leftrightarrow (pWq) \vee (pWr)$$

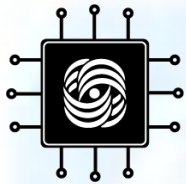
$$(p \wedge q)Wr \Leftrightarrow (pWr) \wedge (qWr)$$

$$pU(q \vee r) \Leftrightarrow (pUq) \vee (pUr)$$

$$(p \wedge q)Ur \Leftrightarrow (pUr) \wedge (qUr)$$

$$[]\Diamond(p \vee q) \Leftrightarrow []\Diamond p \vee []\Diamond q$$

$$\Diamond[](p \wedge q) \Leftrightarrow \Diamond[]p \wedge \Diamond[]q$$



Примеры темпоральных свойств

- **p** всегда истинно;
- **p** рано или поздно станет всегда ложным;
- **p** всегда рано или поздно станет ложным хотя бы ещё один раз;
- **p** всегда ведёт к $\neg q$;
- **p** всегда ведёт к тому, что рано или поздно станет истинным **q**.

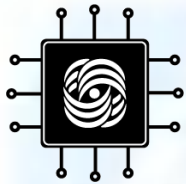
$[]p$

$\langle \rangle [] !p$

$[] (p \rightarrow !q)$

$[] \langle \rangle !p$

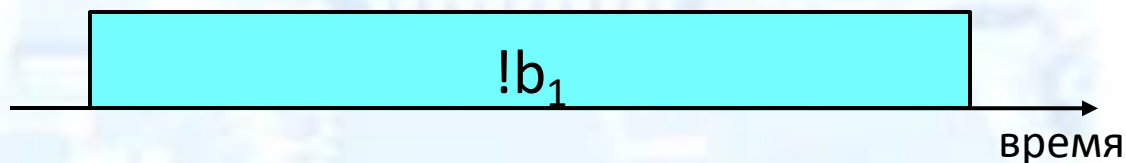
$[] (p \rightarrow \langle \rangle q)$



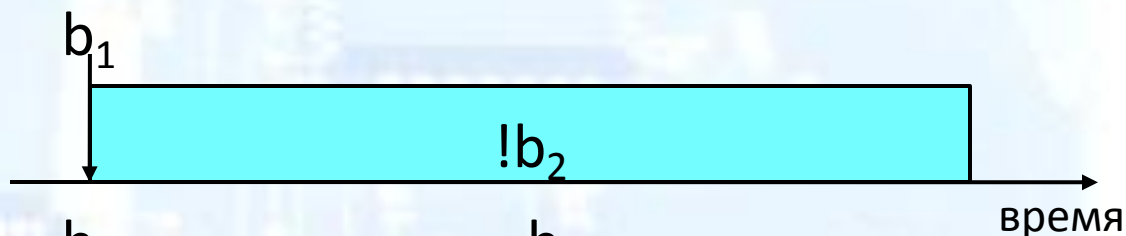
Правильная интерпретация формул LTL

$$LTL: (\langle \rangle (b_1 \wedge (!b_2 U b_2)) \rightarrow [] !a_3)$$

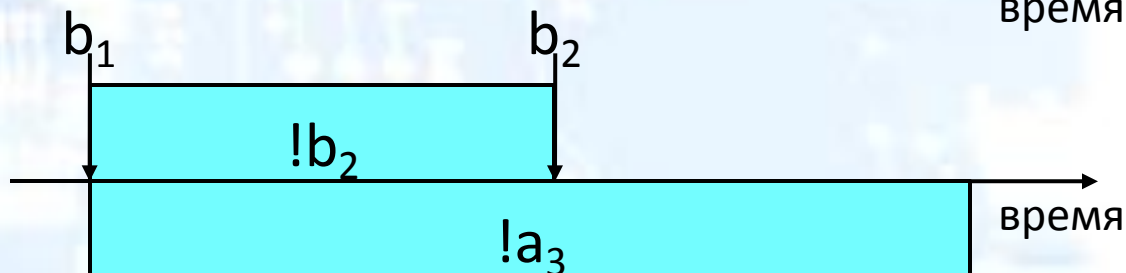
1. Пусть b_1 всегда ложно, $p \rightarrow q$ означает, что $!p \vee q$; **формула выполняется.**



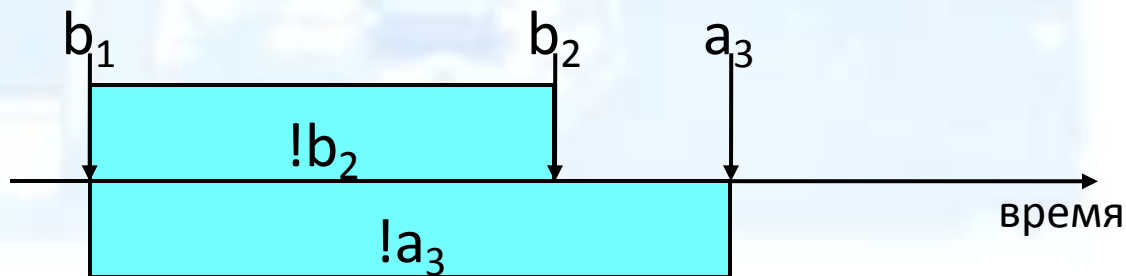
2. Пусть b_1 стало истинно, **формула выполняется.**

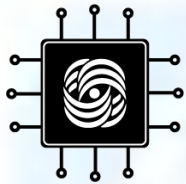


3. Пусть b_1 стало истинно, затем — b_2 , однако a_3 всегда ложно; **формула выполняется.**



4. Пусть b_1 стало истинно, затем — b_2 , затем — a_3 ; **формула не выполняется.**

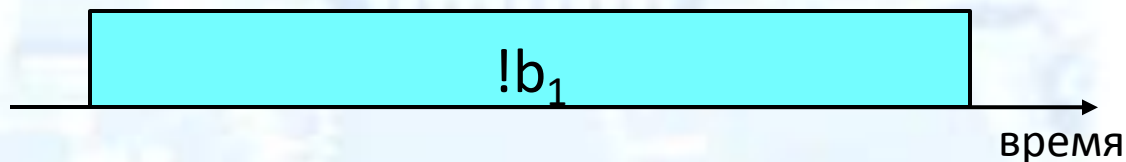




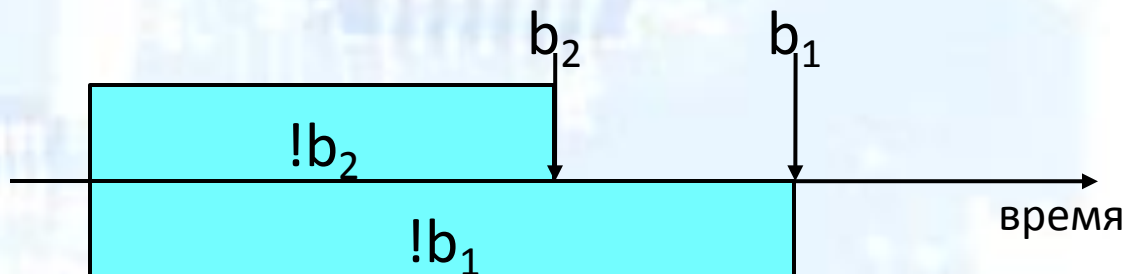
Правильная интерпретация формул LTL

$$LTL: (\langle \rangle b_1) \rightarrow (\langle \rangle b_2)$$

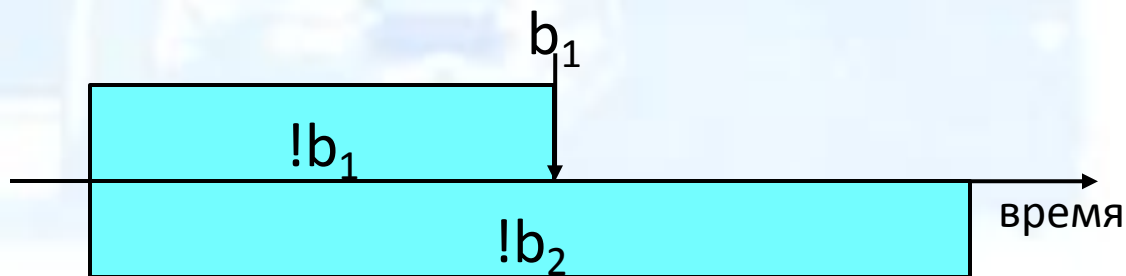
1. Пусть b_1 и b_2 всегда ложно; **формула выполняется.**

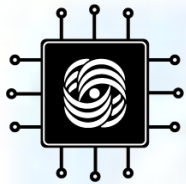


2. Пусть и b_1 , и b_2 становятся истинными; **формула выполняется.**



3. Пусть b_1 становится истинным, но b_2 всегда ложно; **формула не выполняется.**

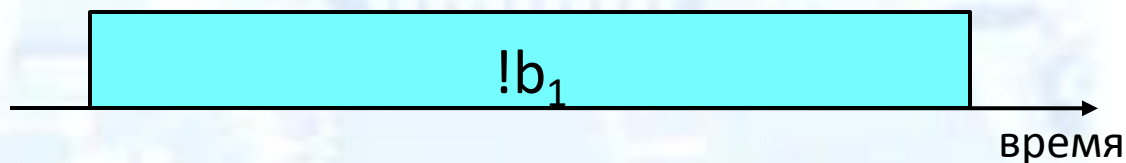




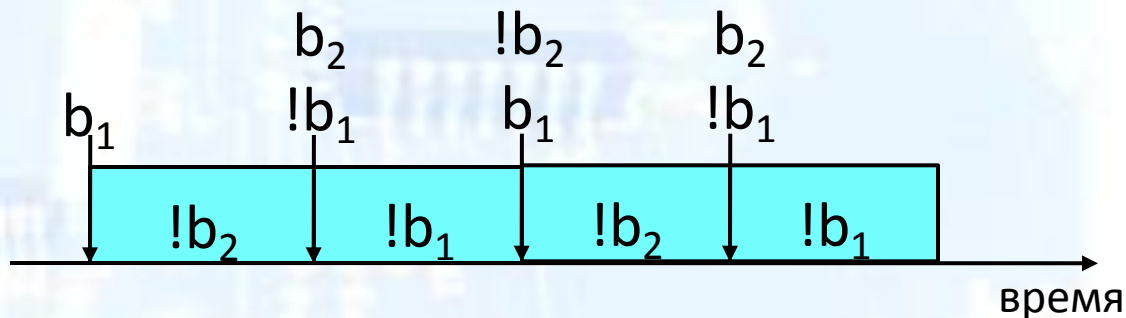
Правильная интерпретация формул LTL

$$LTL: [] (<>b_1) \rightarrow (<>b_2)$$

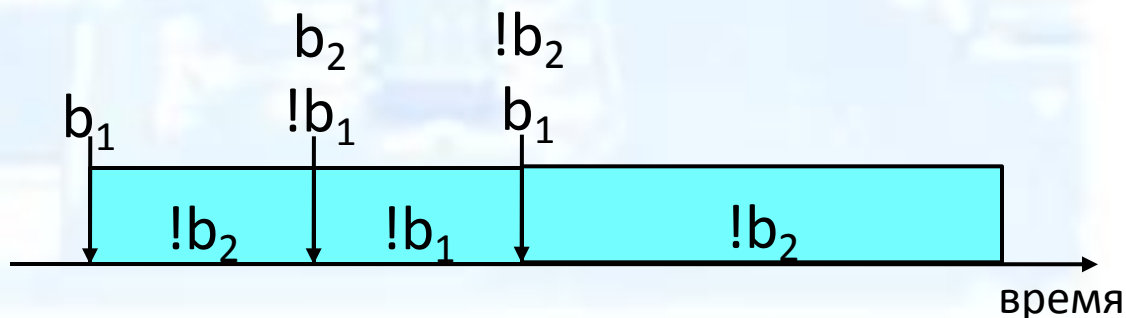
1. Пусть b_1 и b_2 всегда ложно; **формула выполняется.**

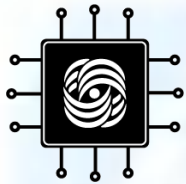


2. Пусть и b_1 , и b_2 бесконечно чередуются; **формула выполняется.**



3. Пусть b_2 становится истинным только один раз; **формула не выполняется.**

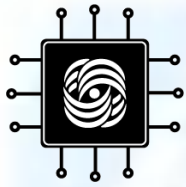




Описание требований при помощи LTL

“ p приведёт к q ”

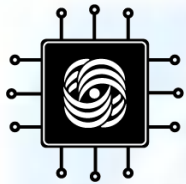
- $p \rightarrow q$
 - нет темпоральных операторов, т.е. применяется только к первому состоянию;
 - выполняется только если $\exists p \vee q$ в первом состоянии, остальная трасса не рассматривается;
 - **не подходит;**
 - **нужно использовать темпоральные операторы.**



Описание требований при помощи LTL

“ p приведёт к q ”

- $[] p \rightarrow q$
 - **правила приоритета!** $[]$ применяется только к p ;
 - означает $([]p) \rightarrow q$;
 - не подходит;
 - **нужно расставить скобки.**

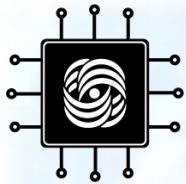


Описание требований при помощи LTL

“ p приведёт к q ”

- **[] ($p \rightarrow q$)**

- проверяем условие во всех состояниях, но причинно-следственная связь между **p** и **q** отсутствует;
- выполняется, только если **$\neg p \vee q$** во всех состояниях;
- не подходит;
- нужно описать, что **p** является причиной **q** .



Описание требований при помощи LTL

“ p приведёт к q ”

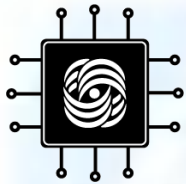
- **[] ($p \rightarrow \langle \rangle q$)**

—уже лучше;

—тем не менее, формула выполнима, если q становится истинным в том же состоянии, что и p — причинно-следственная связь отсутствует;

—не подходит;

—нужно описать, что q не может произойти раньше следующего шага после p .



Описание требований при помощи LTL

“ p приведёт к q ”

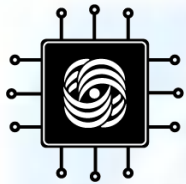
- $[] (p \rightarrow X(<>q))$

- практически то, что нужно;

- формула выполнима, если p всегда ложно;

- возможно, не подходит;

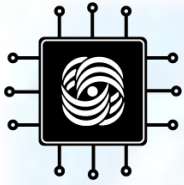
- нужно описать, что p обязательно произойдёт и приведёт к q .



Описание требований при помощи LTL

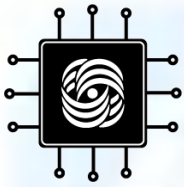
“ p приведёт к q ”

- $[] (p \rightarrow X(<>q)) \ \&\& \ (<>p)$
 - скорее всего, мы имели ввиду именно это;
 - несколько отличается от первоначального $p \rightarrow q$;
 - LTL позволяет выразить множество различных оттенков свойства;
 - подойдёт ли такое свойство для модели параллельной программы?**



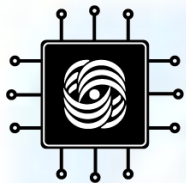
Оператор neXt

- Оператор **X** нужно использовать аккуратно:
 - с его помощью делается утверждение о выполнимости формулы на непосредственных потомках текущего состояния;
 - в распределённых системах значение оператора **X** неочевидно;
 - поскольку алгоритм планирования процессов неизвестен, не стоит формулировать спецификацию в предположении о том, какое состояние будет следующим;
 - стоит ограничиться предположением о справедливости планирования.

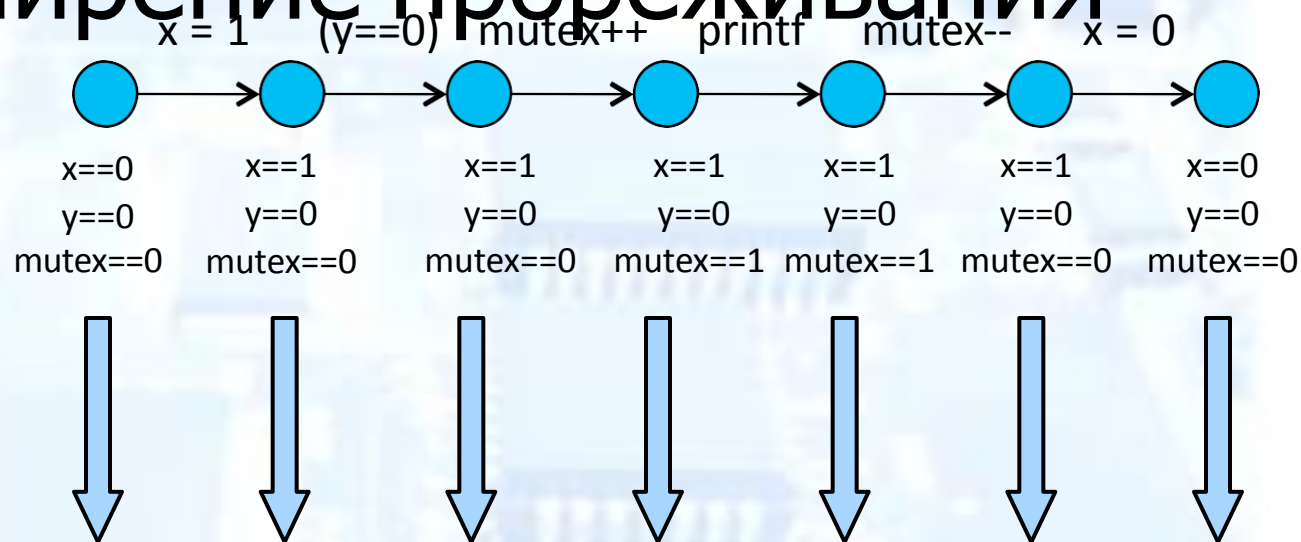


Свойства, инвариантные к прореживанию

- Пусть φ – трасса некоторого вычисления над пропозициональными формулами P ,
 - по трассе можно определить, выполняется ли на ней темпоральная формула,
 - трассу можно записать в виде $\varphi = \varphi_1^{n1} \varphi_2^{n2} \varphi_3^{n3} \dots$, где значения пропозициональных формул на каждом интервале совпадают.
 - Обозначим $E(\varphi)$ набор всех трасс, отличающихся лишь значениями $n1, n2, n3$ (т.е. длиной интервалов)
 - $E(\varphi)$ называется **расширением прореживания** φ .

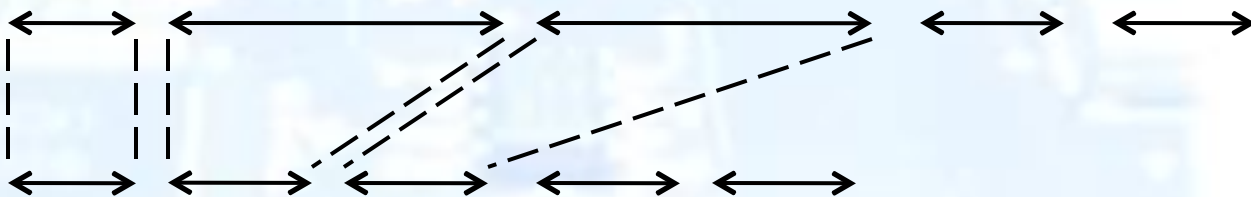


Расширение прореживания



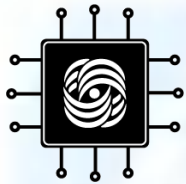
трасса φ

p	$!p$	$!p$	p	p	$!p$	p
$!q$	q	q	q	q	q	$!q$



трасса $\varphi_1 \in E(\varphi)$

p	$!p$	p	$!p$	p
$!q$	q	q	q	$!q$

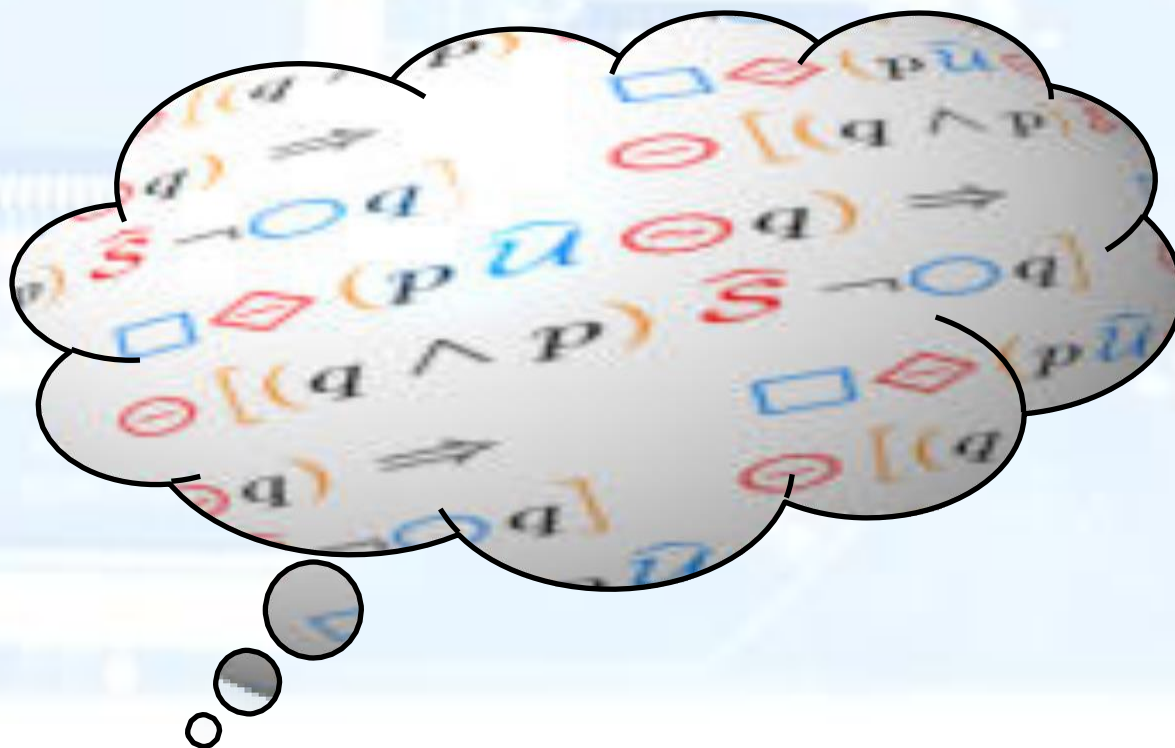
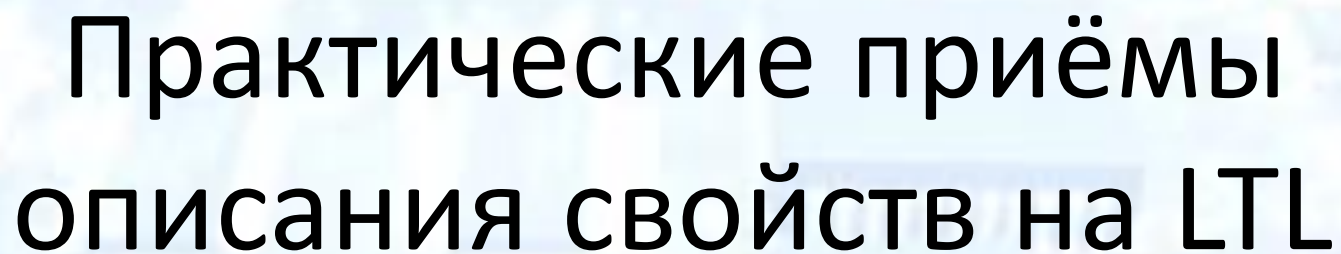


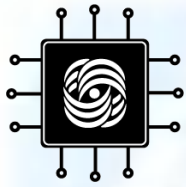
Свойства, инвариантные к прореживанию

- Свойство φ , инвариантное к прореживанию, либо истинно для всех трасс из $E(\varphi)$, либо ни для одной из них:

$$\varphi \models f \rightarrow \forall v \in E(\varphi), v \models f$$

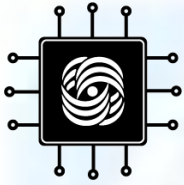
- истинность такого свойства зависит от порядка, в котором пропозициональные формулы меняют свои значения, и не зависит от длины трассы;
- Теорема:** все формулы LTL без оператора X инвариантны к прореживанию.
- Более того, в рамках LTL без X можно описать **все** свойства, инвариантные к прореживанию.





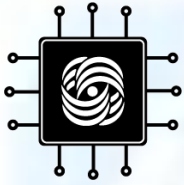
Практические приёмы описания свойств на LTL

- Выполнимость формулы LTL проверяется только для первого состояния в трассе
- Темпоральные операторы управляют проверкой выполнимости своих аргументов
- Сложное свойство можно (и нужно!) строить как суперпозицию простых
- Суперпозиция темпоральных операторов не ограничивает диапазон действия «вложенного» оператора.



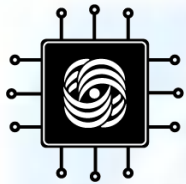
Выполнимость формул LTL

- Выполнимость формулы LTL определена и проверяется для одного (первого) состояния трассы $\left(\begin{array}{l} \text{выполнимость подформул} \\ \text{— уже нет} \end{array} \right)$
- Распространение свойств на другие состояния управляется темпоральными операторами $\left(\begin{array}{l} \text{для этого они и нужны} \end{array} \right)$
- Единственный оператор, который может ограничить сверху проверку выполнимости формулы — **Until** $\left(\begin{array}{l} \text{описать свойство, выполняющееся} \\ \text{для участка программы (одной} \\ \text{итерации, одного вызова функции)} \\ \text{без Until не получится} \end{array} \right)$



Суперпозиция формул LTL

- Составлять сложные формулы LTL нужно методом суперпозиции простых формул
- Внешняя формула задаёт, на каких участках вычислений будет проверяться подформула
- Подформула задаёт свойства, проверяемые для участков вычислений
- Суперпозиция темпоральных операторов не ограничивает диапазон действия «вложенного» оператора ■



База шаблонов темпоральной логики

<http://patterns.projects.cis.ksu.edu>

Логические паттерны (LTL/CTL/GIL)

встречаемость
(occurrence)

отсутствие
(absense)

универсальность
(universality)

существование
(existence)

bounded
existence

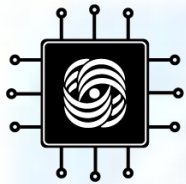
порядок
(order, sequence)

приоритет
(precedence)

chain
precedence

отклик
(response)





База шаблонов темпоральной логики

<http://patterns.projects.cis.ksu.edu>

- Для каждого шаблона – пять вариантов формул:

имя

пример для “absense” и LTL

всегда

$[] (!p)$

перед r

$\langle \rangle r \rightarrow (!p \cup r)$

после q

$[] (q \rightarrow []) (!p))$

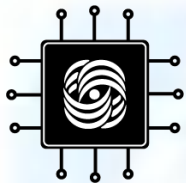
между r и q

$[] ((r \ \&\& \ !q \ \&\& \ \langle \rangle q) \rightarrow (!p \cup q))$

после r до q

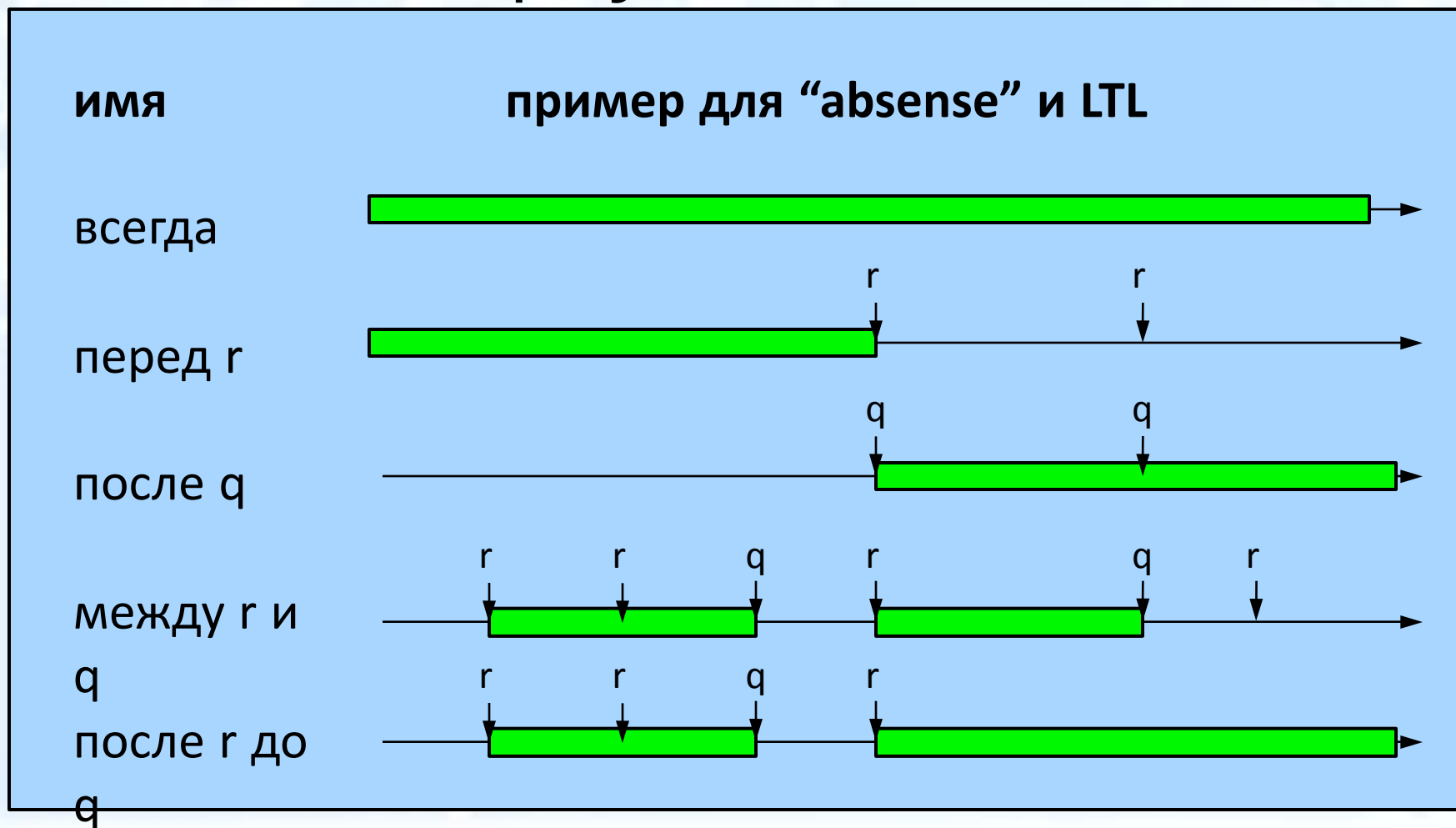
$[] ((r \ \&\& \ !q) \rightarrow ((!p \cup q) \ || \ [] !p))$

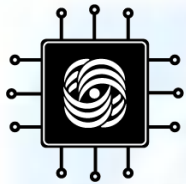
В чём разница?



База шаблонов темпоральной логики <http://patterns.projects.cis.ksu.edu>

- Для каждого шаблона пять вариантов формул:

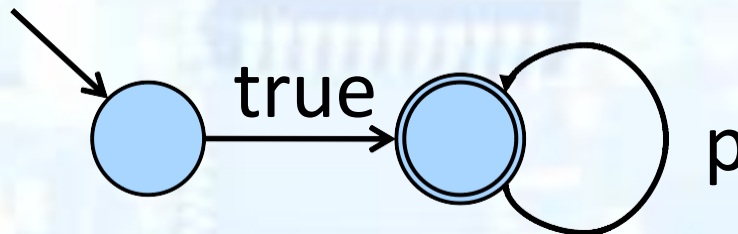




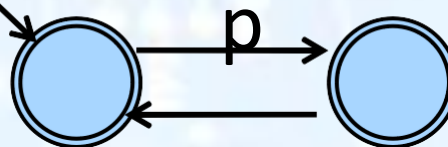
Пример свойства, не выражимого на LTL

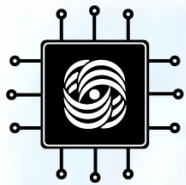
- (p) **может** быть истинным после выполнения системой чётного числа шагов, но **никогда не истинно** после **нечётного**.

- $[]X(p)$ не подходит



- $p \ \&\& \ [](p \rightarrow X!p) \ \&\& \ [](!p \rightarrow Xp)$ – также не подходит (здесь p **всегда** истинно после чётных шагов)





Спасибо за внимание