Математические методы верификации схем и программ

Лекторы:

Захаров Владимир Анатольевич

Подымов Владислав Васильевич

e-mail рассказчика:

valdus@yandex.ru

Осень 2017

Лекция 9

Системы реального времени

Временные автоматы

Неправдоподобные вычисления

Timed CTL

Задача model checking для Timed CTL

Системы реального времени

Попробуем формализовать такой “реальный” пример:

- птица летает по полю и охотится на рой комаров
- птица может быть сытой, голодной и мёртвой
- птица рождается сытой далеко от роя
- если птица долго не ест, то она начинает голодать
- если птица долго голодает, то она умирает
- если птица голодна, то она летит ближе к рою
- если голодная птица подлетела к рою, то она пытается наесться, схватив комара
- рой, заметив подлетевшую птицу, улетает

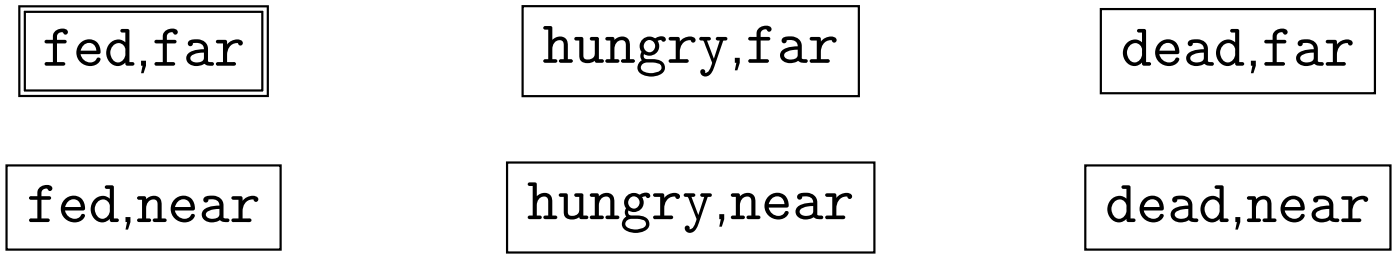
Модель Крипке, которую можно получить для этой системы, действуя “в лоб”, *скорее всего* будет содержать 6 состояний: $\{\mathtt{fed}, \mathtt{hungry}, \mathtt{dead}\} \times \{\mathtt{far}, \mathtt{near}\}$, где

- $\mathtt{fed}$, $\mathtt{hungry}$, $\mathtt{dead}$ — состояния птицы: накормлена, голодна и мертва
- $\mathtt{far}$, $\mathtt{near}$ — состояние (*положение*) роя

Системы реального времени

Попробуем формализовать такой “реальный” пример:

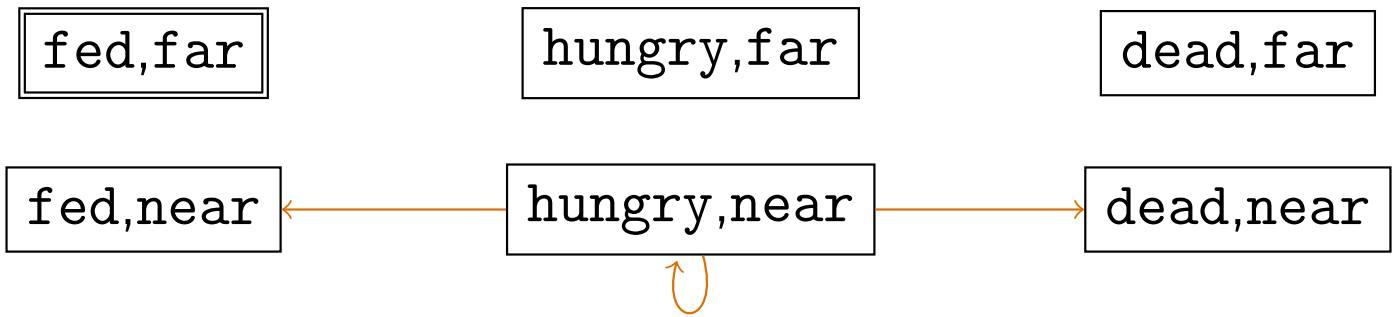
- состояния: $\{\texttt{fed}, \texttt{hungry}, \texttt{dead}\} \times \{\texttt{far}, \texttt{near}\}$
- начальное состояние: (fed,far)
- если долго fed, то наступает hungry
- если долго hungry, то наступает dead
- если hungry, то птица летит ближе к рою
- если hungry и near, то птица пытается наесться, схватив комара
- рой, заметив подлетевшую птицу, становится far



Системы реального времени

Попробуем формализовать такой “реальный” пример:

- состояния: $\{\texttt{fed}, \texttt{hungry}, \texttt{dead}\} \times \{\texttt{far}, \texttt{near}\}$
- начальное состояние: (fed,far)
- если долго fed, то наступает hungry
- если долго hungry, то наступает dead
- если hungry, то птица летит ближе к рою
- если hungry и near, то птица пытается наесться, схватив комара
- рой, заметив подлетевшую птицу, становится far



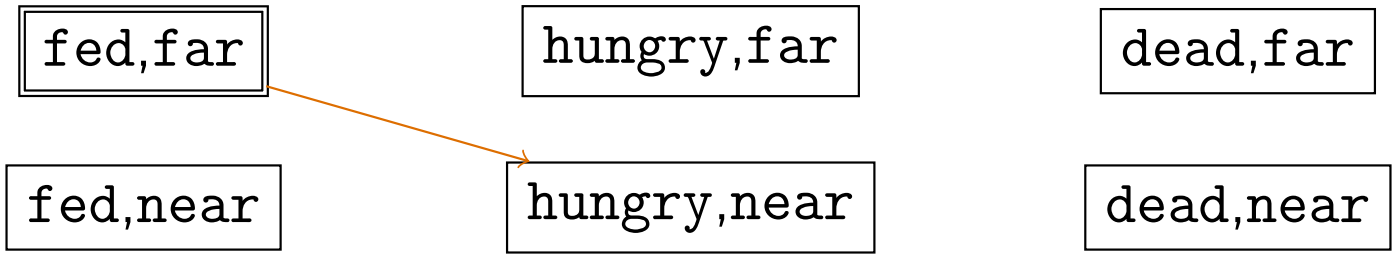
Может ли птица быть голодной бесконечно долго?

Может ли птица быть голодной сколь угодно долго?

Системы реального времени

Попробуем формализовать такой “реальный” пример:

- состояния: $\{\texttt{fed}, \texttt{hungry}, \texttt{dead}\} \times \{\texttt{far}, \texttt{near}\}$
- начальное состояние: (fed,far)
- если долго fed, то наступает hungry
- если долго hungry, то наступает dead
- если hungry, то птица летит ближе к рою
- если hungry и near, то птица пытается наесться, схватив комара
- рой, заметив подлетевшую птицу, становится far

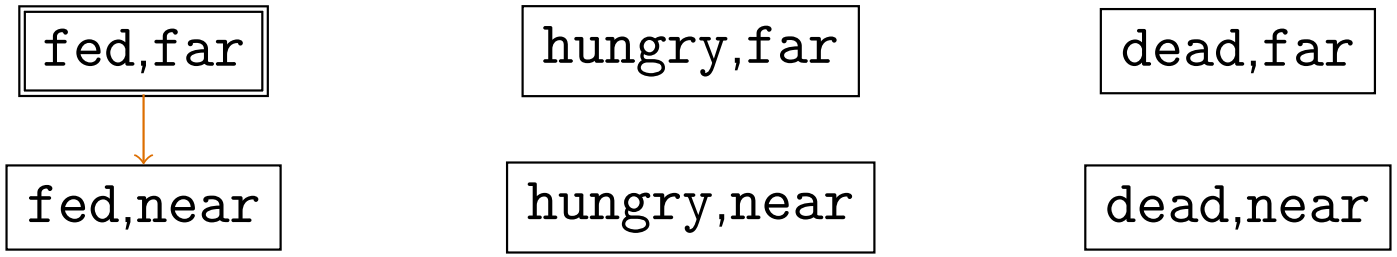


Может ли птица одновременно подлететь к рою и стать голодной?

Системы реального времени

Попробуем формализовать такой “реальный” пример:

- состояния: $\{\texttt{fed}, \texttt{hungry}, \texttt{dead}\} \times \{\texttt{far}, \texttt{near}\}$
- начальное состояние: (fed,far)
- если долго fed, то наступает hungry
- если долго hungry, то наступает dead
- если hungry, то птица летит ближе к рою
- если hungry и near, то птица пытается наесться, схватив комара
- рой, заметив подлетевшую птицу, становится far

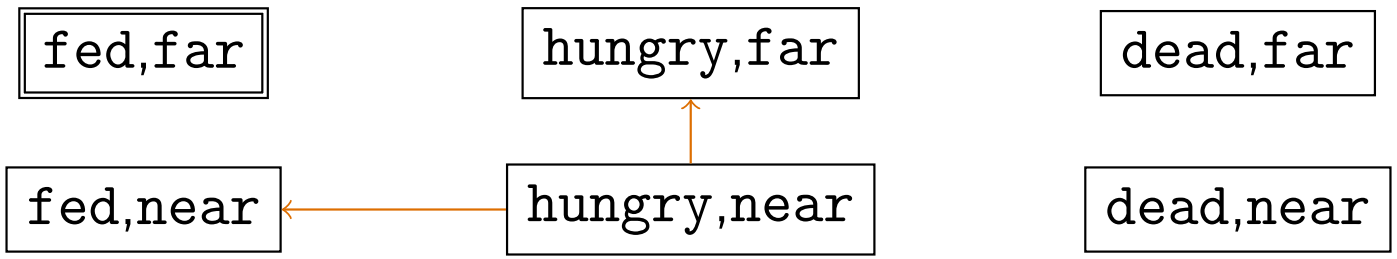


Может ли птица мгновенно подлететь, проголодаться, схватить комара и улететь?

Системы реального времени

Попробуем формализовать такой “реальный” пример:

- состояния: $\{\texttt{fed}, \texttt{hungry}, \texttt{dead}\} \times \{\texttt{far}, \texttt{near}\}$
- начальное состояние: (fed,far)
- если долго fed, то наступает hungry
- если долго hungry, то наступает dead
- если hungry, то птица летит ближе к рою
- если hungry и near, то птица пытается наесться, схватив комара
- рой, заметив подлетевшую птицу, становится far



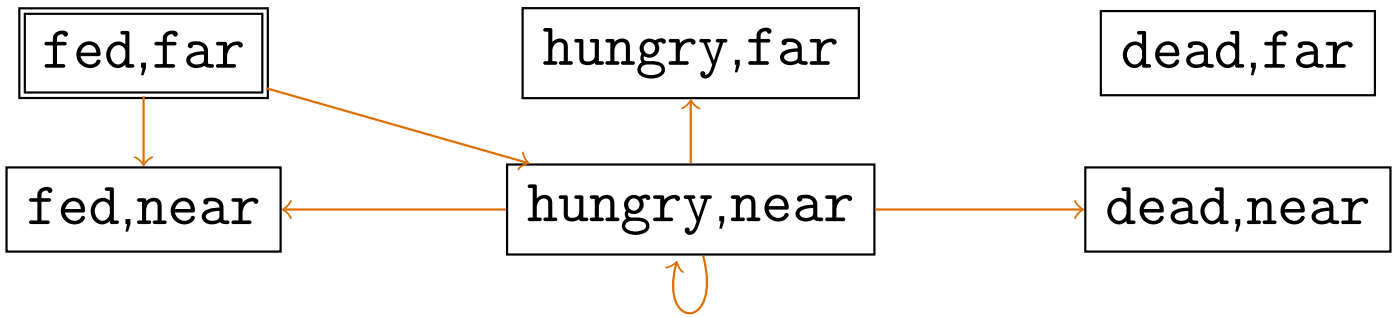
Всегда ли подлетевшая голодная птица имеет возможность перекусить комарами?

Всегда ли комары имеют возможность улететь от птицы?

Системы реального времени

Попробуем формализовать такой “реальный” пример:

- состояния: $\{\texttt{fed}, \texttt{hungry}, \texttt{dead}\} \times \{\texttt{far}, \texttt{near}\}$
- начальное состояние: (fed,far)
- если долго fed, то наступает hungry
- если долго hungry, то наступает dead
- если hungry, то птица летит ближе к рою
- если hungry и near, то птица пытается наесться, схватив комара
- рой, заметив подлетевшую птицу, становится far



Насколько наличие и отсутствие упомянутых и других дуг подходит для точного описания всех птиц в целом и **каждой конкретной** птицы?

Системы реального времени

Попробуем формализовать такой “реальный” пример:

- состояния: $\{\texttt{fed}, \texttt{hungry}, \texttt{dead}\} \times \{\texttt{far}, \texttt{near}\}$
- начальное состояние: (`fed`,`far`)
- если долго `fed`, то наступает `hungry`
- если долго `hungry`, то наступает `dead`
- если `hungry`, то птица летит ближе к рою
- если `hungry` и `near`, то птица пытается наесться, схватив комара
- рой, заметив подлетевшую птицу, становится `far`

Чтобы формализовать этот пример для “содержательной” проверки того, умрёт или нет конкретная птица, следует ввести **временные характеристики** птицы и роя:

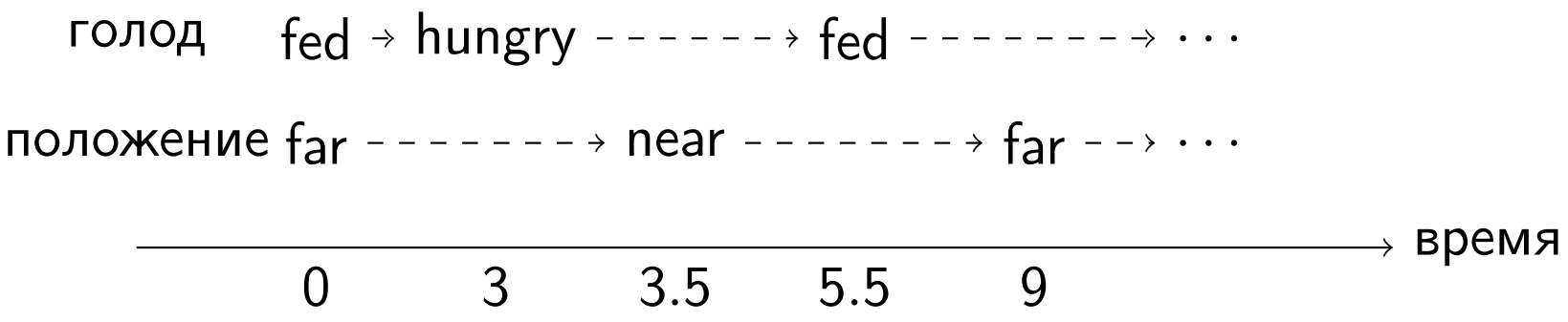
- сколько птица способна прожить без еды
- как долго птица переваривает комара
- как быстро рой замечает птицу
- ...

Системы реального времени

Примеры временных характеристик:

- птица переваривает комара ровно 3 минуты
- поголодав 5 минут, птица умирает
- голодная птица не более чем за 1 минуту подлетает к рою комаров
- птице требуется не более 2 минут, чтобы поймать комара, когда она подлетела к рою
- рой замечает птицу не менее чем за 3 минуты
- заметив птицу, рой немедленно улетает

С учётом временных характеристик система может работат, например, так:

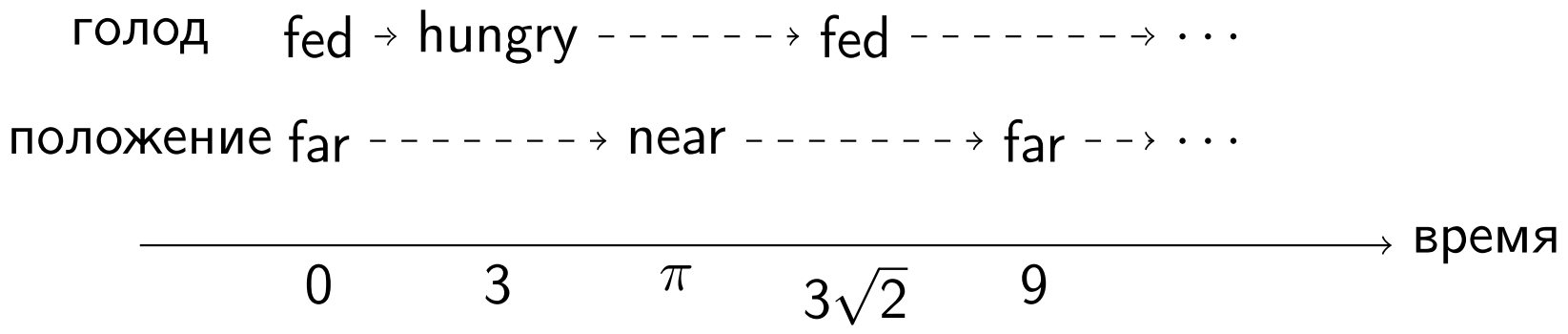


Системы реального времени

Примеры временных характеристик:

- птица переваривает комара ровно 3 минуты
- поголодав 5 минут, птица умирает
- голодная птица не более чем за 1 минуту подлетает к рою комаров
- птице требуется не более 2 минут, чтобы поймать комара, когда она подлетела к рою
- рой замечает птицу не менее чем за 3 минуты
- заметив птицу, рой немедленно улетает

С учётом временных характеристик система может работат, например, так:

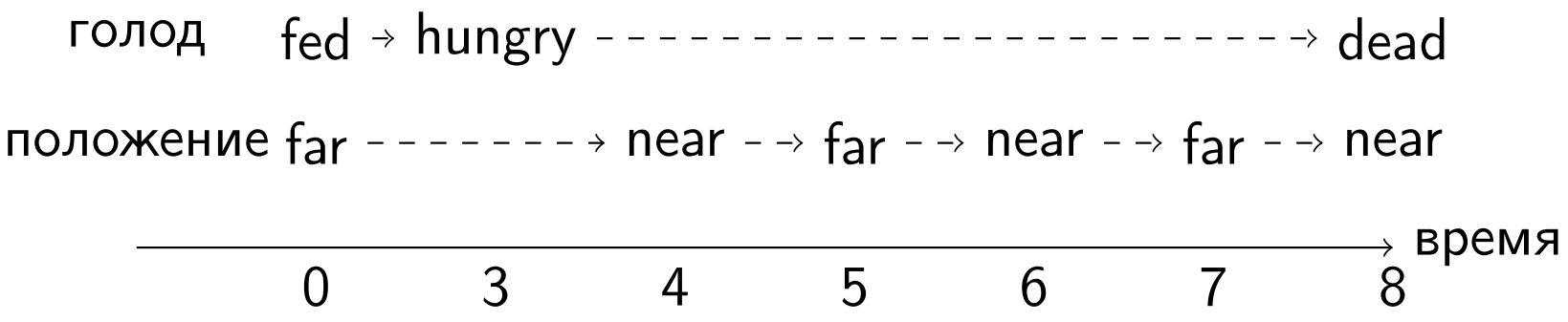


Системы реального времени

Примеры временных характеристик:

- птица переваривает комара ровно 3 минуты
- поголодав 5 минут, птица умирает
- голодная птица не более чем за 1 минуту подлетает к рою комаров
- птице требуется не более 2 минут, чтобы поймать комара, когда она подлетела к рою
- рой замечает птицу не менее чем за 1 минуту
- заметив птицу, рой немедленно улетает

С учётом временных характеристик система может работат, например, так:



Системы реального времени

Примеры временных характеристик:

- птица переваривает комара ровно 3 минуты
- поголодав 5 минут, птица умирает
- голодная птица не более чем за 1 минуту подлетает к рою комаров
- птице требуется не более 2 минут, чтобы поймать комара, когда она подлетела к рою
- рой замечает птицу не менее чем за ? минут
- заметив птицу, рой немедленно улетает

В зависимости от того, насколько быстро будут действовать птица и рой, птица может как накормиться, так и умереть

Системы реального времени

Система реального времени (СРВ) — это система, поведение которой существенно зависит не только от того, в каком порядке изменяются состояния компонентов системы, но и от того, **за какое время** происходит изменение состояний

Некоторые СРВ содержат директивные сроки выполнения задач, несоблюдение которых приводит к неблагоприятным последствиям, из-за которых дальнейшее выполнение системы становится неважным:

- птица не поела вовремя — умерла
- парашют не раскрылся в срок — тоже не очень хорошо
- подушка безопасности раскрылась на долю секунды позже — и опять всё плохо

СРВ с такими директивными сроками принято называть жёсткими

Системы реального времени

Система реального времени (СРВ) — это система, поведение которой существенно зависит не только от того, в каком порядке изменяются состояния компонентов системы, но и от того, **за какое время** происходит изменение состояний

В некоторых системах срыв директивных сроков возможен, хотя и может приводить к ухудшению качества работы системы:

- поспал на два часа меньше — будешь вялым, но не умрёшь
- процесс освободил память на полминуты позже плана — операционная система “подвиснет”, но не сломается
- почта задержалась на год — печально, но бывает

СРВ с такими директивными сроками принято называть мягкими

Далее рассматриваются только **жёсткие** СРВ

Системы реального времени

Система реального времени (СРВ) — это система, поведение которой существенно зависит не только от того, в каком порядке изменяются состояния компонентов системы, но и от того, **за какое время** происходит изменение состояний

Для верификации СРВ необходимо иметь

1. средство описания моделей, учитывающее их временные характеристики
2. формальный язык описания требований, учитывающий реальное время
3. алгоритм проверки требований для составленной модели

Системы реального времени

Состояние СРВ должно в том или ином виде включать в себя текущее реальное время

$$\boxed{\text{hungry,near,2'53''}} \rightarrow ?$$

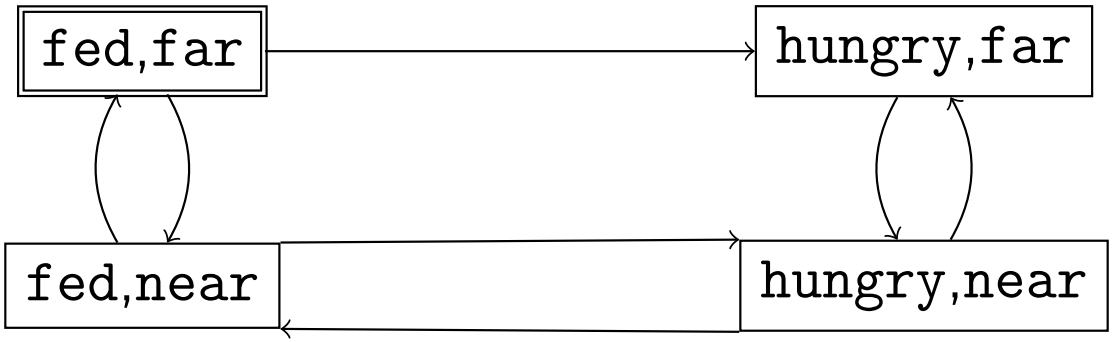
Если явно включить время в состояния модели Крипке, то введённый *в предыдущих лекциях* аппарат проверки свойств становится неприменимым:

- пути и трассы моделей — это дискретные последовательности, а время течёт непрерывно
  - при этом изменение состояний СРВ, как правило, дискретно, что позволит адаптировать вводившиеся ранее понятия к реальному времени
- число состояний модели континуально, а решающие алгоритмы формулировались только для конечных моделей
  - в следующей лекции будет показано, как свести проверку свойств модели с реальным временем к проверке свойств конечной модели

Временные автоматы

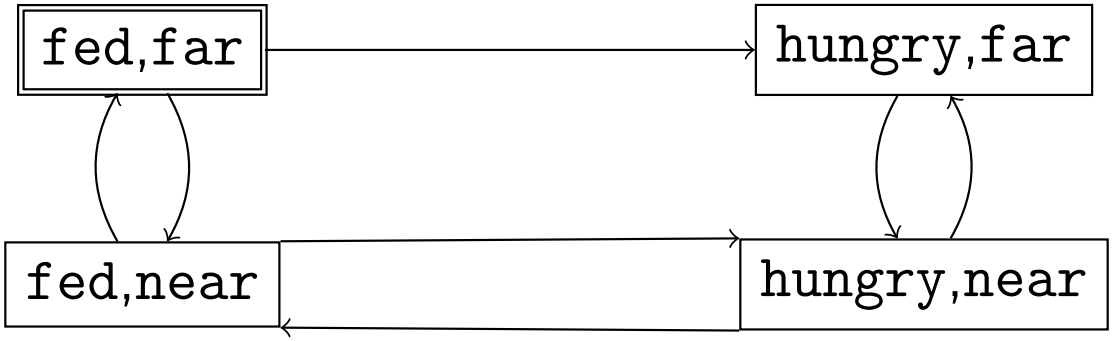
Перед строгими определениями рассмотрим **пример:** модель Крипке для птицы и роя,

- немного упрощённую: птица бессмертна
- не учитывающую реальное время
- содержащую переходы, согласно которым изменяется **либо** состояние птицы, **либо** состояние роя



Попробуем разметить эту модель утверждениями о реальном времени, позволяющими исследовать поведение **конкретной** птицы и **конкретного** роя

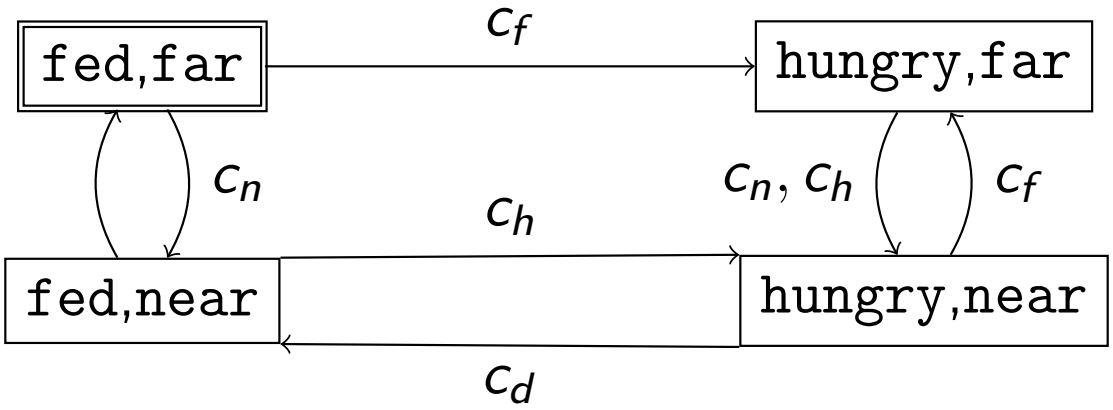
Временные автоматы



Представим себе, что в нашем распоряжении есть сколько угодно **часов**, которые можно использовать как секундомеры:

- часы начинают отсчёт времени с ноля
- время на часах увеличивается в том же темпе, что и реальное время
- в любой момент мы можем нажать кнопку, чтобы сбросить показание часов в ноль

Временные автоматы



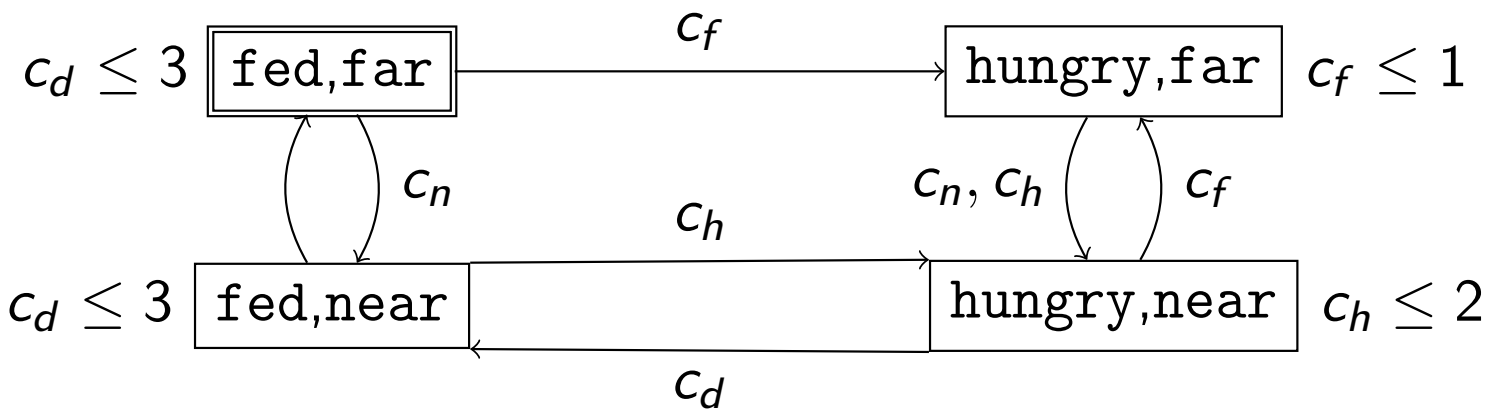
Добавим в модель часы для отсчёта того, сколько времени

- птица переваривает последнего комара: c_d
- птица летает вблизи роя: c_n
- голодная птица подлетает к рою: c_f
- голодная птица вблизи роя охотится на комара: c_h

Сбрасывать часы будем при совершении тех переходов, после которых следует начинать отсчёт

Разметим каждый переход множеством часов, сбрасываемых при совершении этого перехода

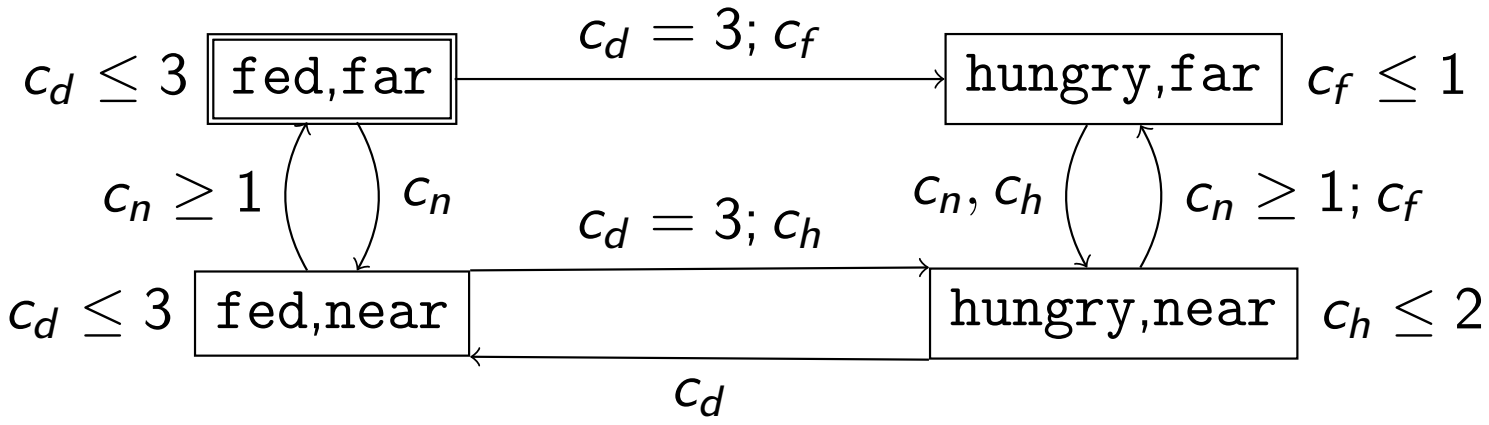
Временные автоматы



Пометим некоторые состояния s **инвариантами** $I(s)$: ограничениями на значения часов, которые обязаны выполняться, пока система находится в состоянии s

- голодная птица подлетает к рою не более чем за 1 минуту:
$$I(\texttt{hungry}, \texttt{far}) = (c_f \leq 1)$$
- птица охотится на комара не более 2 минут:
$$I(\texttt{hungry}, \texttt{near}) = (c_h \leq 2)$$
- птица переваривает комара ровно 3 минуты:
$$I(fed, \cdot) = (c_d \leq 3)$$

Временные автоматы



Пометим некоторые переходы e **предусловиями** $G(e)$: ограничениями на значения часов, которые запрещают совершение перехода при их невыполнении

- рой замечает птицу не менее чем за минуту:
$$G((\cdot, \texttt{near}) \to (\cdot, \texttt{far})) = (c_n \geq 1)$$
- птица переваривает комара ровно 3 минуты:
$$G((\texttt{fed}, \cdot) \to (\texttt{hungry}, \cdot)) = (c_d = 3)$$

Результат разметки модели сбросами, инвариантами и предусловиями — это **временной автомат**

Временные автоматы

$\mathbb{N}_0$ — множество всех целых неотрицательных чисел

Элементарное временное ограничение — это выражение вида **true**, $c < k$ или $c \leq k$, где c — часы и $k \in \mathbb{N}_0$

$ATC(C)$ — множество всевозможных элементарных временных ограничений над множеством часов C

Элементарное разностное ограничение — это выражение вида $c_1 - c_2 < k$ или $c_1 - c_2 \leq k$, где c_1 и c_2 — часы и $k \in \mathbb{N}_0$

$ADC(C)$ — множество всевозможных элементарных разностных ограничений над множеством часов C

$AC(C) = ATC(C) \cup ADC(C)$ — множество всевозможных элементарных ограничений

Временные автоматы

Временное ограничение tc (*над множеством часов* C) определяется следующей БНФ, где $e \in AC(C)$:

$$tc ::= e \mid (tc) \mid tc \,\&\, tc \mid \neg tc$$

Предусловие — это синоним временного ограничения

$Guard(C)$ — это множество всех предусловий над C

Инвариант — это конъюнкция элементарных временных ограничений (**true**, $c < k$, $c \leq k$)

$Inv(C)$ — это множество всех инвариантов над C

Временные автоматы

$\mathbb{R}_{\geq 0}$ — множество всех неотрицательных действительных чисел

Оценка часов C — это любая функция $\nu : C \to \mathbb{R}_{\geq 0}$

Выполнимость временного ограничения tc на оценке часов ν ($\nu \models tc$) определяется естественным образом:

- $\nu \models (tc) \quad \Leftrightarrow \quad \nu \models tc$
- $\nu \models c \ op \ k \quad \Leftrightarrow \quad \nu(c) \ op \ k$
- $\nu \models c_1 - c_2 \ op \ k \quad \Leftrightarrow \quad \nu(c_1) - \nu(c_2) \ op \ k$
- $\nu \models tc_1 \ \& \ tc_2 \quad \Leftrightarrow \quad \nu \models tc_1$ и $\nu \models tc_2$
- $\nu \models \neg tc_1 \quad \Leftrightarrow \quad \nu \not\models tc_1$

Временные автоматы

Также в записи предусловий можно использовать

- все булевы связки
  - $\{\&, \neg\}$ — *полная система булевых функций*
- константу **false**
  - **false** $\equiv$ $\neg$**true**
- отношения $>, \geq, =, \neq$: (*помимо* $<$ *и* $\leq$)
  - $\alpha > \beta \quad \equiv \quad \neg(\alpha \leq \beta)$
  - $\alpha \geq \beta \quad \equiv \quad \neg(\alpha < \beta)$
  - $\alpha = \beta \quad \equiv \quad (\alpha \leq \beta) \,\&\, (\alpha \geq \beta)$
  - $\alpha \neq \beta \quad \equiv \quad \neg(\alpha = \beta)$

Временные автоматы

Временной автомат над множеством атомарных высказываний AP — это система $(L, \ell_0, C, \xi, I, T)$, где

- L — конечное множество состояний автомата
- ℓ_0 — начальное состояние, $\ell_0 \in L$
- C — конечное множество **часов**
- $\xi : L \to 2^{AP}$ — разметка состояний высказываниями
- $I : L \to Inv(C)$ — разметка состояниий **инвариантами**
- $T \subseteq L \times Guard(C) \times 2^C \times L$ — отношение переходов
  - (l_1, g, X, l_2) — переход из состояния l_1 в состояние l_2 с **предусловием** g и множеством **сбрасываемых** часов X
  - такой переход будем записывать и в другом виде: $l_1 \xrightarrow{g,X} l_2$

Целые числа во временных ограничениях

Во всех элементарных ограничениях значения часов сравниваются с **целыми** неотрицательными числами

Это означает, например, что выражение "$c < \sqrt{2}$" **не является** элементарным ограничением

Принято считать, что элементарные ограничения с нецелыми числами в правой части излишни:

- числа, используемые на практике при описании СРВ, **рациональны**: любое действительное число может быть приближено рациональным с любой точностью
- если все числа, используемые в ограничениях временного автомата, рациональны, то можно привести их к общему знаменателю d и вычеркнуть знаменатель, оставив только числитель
- при формулировке свойств модели достаточно учесть, что модельное время течёт в d раз медленнее реального

Семантика временных автоматов

Содержательное вступление

Текущее состояние вычисления (*конфигурация*) СРВ определятся тем, в каких состояниях находятся компоненты системы, и действительным значением текущего времени

В процессе вычисления время непрерывно неограниченно увеличивается, и состояния компонентов “плавно” изменяются

При этом семантика временного автомата дискретна: в процессе вычисления он

- либо увеличивает время на заданную величину (и “проживает” все моменты времени от исходного до конечного, не изменяя состояния)
- либо мгновенно изменяет состояния компонентов

При моделировании СРВ временными автоматами следует учитывать это расхождение “реальной” и “модельной” семантик

Семантика временных автоматов

Конфигурация временного автомата $A = (L, \ell_0, C, \xi, I, T)$ — это пара (ℓ, ν), где $\ell \in L$ и $\nu : C \to \mathbb{R}_{\geq 0}$

Для простоты иногда будем полагать, что часы автомата упорядочены: $C = (x_1, \ldots, x_m)$ — и записывать конфигурацию автомата как состояние, после которого располагаются значения всех таймеров: $(\ell, d_1, \ldots, d_m)$

Начальная конфигурация автомата имеет вид $(\ell_0, 0, 0, \ldots, 0)$

Вычисление временного автомата — это бесконечная последовательность конфигураций, такая что

- первая конфигурация последовательности — начальная
- каждая следующая конфигурация c_{i+1} получается из предыдущей c_i (сокращённо: $c_i \to c_{i+1}$) одним из двух способов:
  - продвижение времени: $c_i \mapsto c_{i+1}$
  - изменение состояния: $c_i \hookrightarrow c_{i+1}$

Семантика временных автоматов

Продвижение времени $(A = (L, \ell_0, C, \xi, I, T))$

$(\ell, \nu) \mapsto_d (\ell, \nu + d)$, где:

- d — положительное действительное число
- $\nu + d$ — это оценка часов, такая что $(\nu + d)(x) = \nu(x) + d$ для любых часов x
- $\nu + d \models I(\ell)$

$(\ell, \nu) \mapsto (\ell', \nu')$, если существует положительное действительное чило d, такое что $(\ell, \nu) \mapsto_d (\ell', \nu')$

Семантика временных автоматов

Изменение состояния $(A = (L, \ell_0, C, \xi, I, T))$

$(\ell, \nu) \hookrightarrow (\ell', \nu')$ в том и только том случае, если существуют предусловие g и множество часов X, такие что:

- $\ell \xrightarrow{g,X} \ell' \in T$
- $\nu \models g$
- $\nu' = \nu[X]$
  - $\nu[X]$ — это оценка часов, такая что для любых часов x верно
    - $\nu[X](x) = 0$, если $x \in X$
    - $\nu[X](x) = \nu(x)$, если $x \notin X$
- $\nu' \models I(\ell')$

Неправдоподобные вычисления

Не каждое вычисление временного автомата *адекватно* описывает поведение реальной СРВ

При рассуждении о выполнении реальной СРВ обычно предполагается, что

- любой момент времени рано или поздно “проживается” системой
  - время для наблюдателя неограниченно увеличивается в заданном темпе
  - время, “прожитое” в вычислении временного автомата, может быть ограничено сверху
- если система, начиная с текущего момента времени, всегда может изменить состояние, то рано или поздно она изменит состояние
  - это слабая справедливость для действия “изменение состояния”

Неправдоподобные вычисления

Конвергентное вычисление — это вычисление, некоторый суффикс которого имеет вид

$$c_0 \mapsto_{d_1} c_1 \mapsto_{d_2} c_2 \mapsto_{d_3} \dots ,$$

где $\sum_{i=1}^{\infty} d_i$ — сходящийся ряд

Другими словами, конвергентное вычисление оканчивается тем, что система "проживает" **конечное** время и не успевает ничего сделать, несмотря на то, что за пределами этого времени могла бы

Например, такое вычисление является конвергентным:

$$(\ell, 0) \mapsto (\ell, \tfrac{1}{2}) \mapsto (\ell, \tfrac{2}{3}) \mapsto \dots \mapsto (\ell, \tfrac{n-1}{n}) \mapsto \dots$$

Такое вычисление **неправдоподобно**: наблюдая работу системы, мы должны рано или поздно увидеть на часах **каждое** значение времени

Неправдоподобные вычисления

Вычисление Зенона — это вычисление, удовлетворяющее двум условиям:

- в нём бесконечно часто происходит изменение состояния
- либо продвижение времени происходит в нём только конечное число раз, либо время продвигается бесконечно часто согласно отношениям $\mapsto_{d_1}, \mapsto_{d_2}, \ldots$, где $\sum\limits_{i=1}^{\infty} d_i$ — сходящийся ряд

Вычисление Зенона в модели временных автоматов — это наглядная иллюстрация *апорий Зенона*, суть которых — это выполнение бесконечного числа действий за конечное время

Например, такое вычисление является вычислением Зенона:

$$(\ell_0, 0) \mapsto (\ell_0, 1) \hookrightarrow (\ell_1, 0) \mapsto \cdots \mapsto (\ell_i, \tfrac{1}{2^i}) \hookrightarrow (\ell_{i+1}, 0) \mapsto \ldots$$

Такое вычисление считается настолько же **неправдоподобным**, насколько и апории Зенона

Неправдоподобные вычисления

Стагнирующее вычисление — это вычисление, некоторый суффикс которого имеет вид

$$(\ell, \nu_1) \mapsto_{d_1} (\ell, \nu_2) \mapsto_{d_2} (\ell, \nu_3) \mapsto_{d_3} \dots ,$$

где ряд $\sum_{i=1}^{\infty} d_i$ расходится

Стагнирующие вычисления можно поделить на два класса:

- оканчивающиеся неограниченным продвижением времени с возможностью изменить состояние
  - такие вычисления несправедливы по отношению к изменению состояний, и потому **неправдоподобны**
- оканчивающиеся неограниченным продвижением времени с невозможностью изменить состояние
  - автомат, допускающий такое вычисление — это аналог *нетотальной модели Крипке*, и такие автоматы считаются **неправдоподобными** и не рассматриваются

Неправдоподобные вычисления

Прогрессивное вычисление — это вычисление, в котором после каждого продвижения времени обязательно происходит изменение состояния, то есть не существует фрагмента

$$c_i \mapsto c_{i+1} \mapsto c_{i+2}$$

В рамках анализа **правдободобных** вычислений временного автомата достаточно рассматривать только его **прогрессивные** вычисления:

- в любом правдоподобном вычислении изменение состояния происходит бесконечно часто
- любую конечную последовательность продвижений времени

  $$c_i \mapsto_{d_1} c_{i+1} \mapsto_{d_2} \cdots \mapsto_{d_k} c_{i+k}$$

  можно без изменения содержательного смысла вычисления заменить на одно продвижение времени

  $$c_i \mapsto_{d_1+\cdots+d_k} c_{i+k}$$

Неправдоподобные вычисления

Конвергентные и **стагнирующие** вычисления непрогрессивны

При этом даже при рассмотрении только прогрессивных вычислений остаются два случая, в которых временной автомат можно признать некорректным:

1. он содержит **прогрессивные вычисления Зенона**
2. из его начальной конфигурации продвижением времени и изменением состояний можно получить конфигурацию блокировки (deadlock), в которой невозможно изменить состояние как сразу, так и после любого продвижения времени

Чаще всего далее будут рассматриваться только корректные автоматы и их прогрессивные вычисления

Следить за корректностью автомата —
задача того, кто разрабатывает этот автомат

Timed CTL

Попробуем адаптировать логику ветвящегося времени к модели временных автоматов

Для этого достаточно предложить

- аналог модели Крипке, пригодный для описания всех прогрессивных поведений автомата в реальном времени
- аналог CTL, включающий в себя способ рассуждений о реальном времени, с семантикой, подходящей для введённого аналога модели Крипке

Синтаксис формул логики TCTL (Timed CTL) можно определить так: TCTL-формула над множеством атомарных высказываний AP и множеством часов C — это CTL-формула над множеством атомарных высказываний $AP \cup AC(C)$, не содержащая операторов $\mathbf{X}$

Пример TCTL-формулы над высказываниями $\{a, b\}$ и множеством часов $\{x, y\}$:

$$\mathbf{AG}(x < 5 \to (a \,\&\, \mathbf{EF}(y \leq 3 \vee b)))$$

Timed CTL

Модель $M(A)$ временного автомата $A = (L, \ell_0, C, \xi, I, T)$ — это бесконечная система переходов следующего вида:

- состояния модели — это всевозможные конфигурации автомата A
- начальное состояние модели — это начальная конфигурация автомата A
- состояния соединяются дугами двух типов: $c \hookrightarrow c'$ и $c \mapsto c'$ — согласно тому, можно ли получить c' из c в A изменением состояния и продвижением времени соответственно
- каждое состояние (ℓ, ν) помечается множеством $\xi(\ell) \cup \{e \mid e \in AC(C),\ \nu \models e\}$

Можно ли теперь сказать “семантика TCTL-формул на моделях временных автоматов почти дословно повторяет семантику CTL-формул на моделях Крипке”?

Timed CTL

Рассмотрим такие временной автомат A с часами x и TCTL-формулу φ:

$$\mathbf{AF}(x = 1)$$

Согласно содержательной трактовке оператора **F**, это свойство должно быть верным: **при наблюдении любого (прогрессивного) вычисления (корректного) автомата мы дождёмся реального момента времени** 1

При этом существует такое вычисление автомата:

$$(\ell, 0) \mapsto (\ell, 2) \to \dots$$

В этом вычислении момент 1 *подразумевается*,
но **в явном виде отсутствует**,
и с точки зрения CTL формула $\mathbf{AF}(x = 1)$ **не выполнена**

Timed CTL

Определим отношение выполнимости $A \models_T \varphi$ TCTL-формулы φ на корректном автомате A следующим образом:

- $A \models_T \varphi \quad \Leftrightarrow \quad M(A) \models_T \varphi$
- индуктивное определение отношения $\models_T$ TCTL-формулы φ на модели $M(A)$, в состоянии s и на пути tr этой модели дословно повторяет определение аналогичных отношений $\models$, кроме следующих отличий:
  - пути заменяются на прогрессивные вычисления
  - отношение $M, tr \models_T \psi \mathbf{U} \chi$ определяется особенным образом
  - отношения $M, tr \models_T \mathbf{F}\psi$ и $M, tr \models_T \mathbf{G}\psi$ определяются так, чтобы были справедливы равносильности $\mathbf{F}\psi \equiv \mathbf{true}\mathbf{U}\psi$ и $\mathbf{G}\psi \equiv \neg\mathbf{F}\neg\psi$

Timed CTL

Конфигурация (ℓ_1, ν) является промежуточной для пары конфигураций (ℓ_1, ν_1), (ℓ_2, ν_2) в двух случаях:

- $\nu = \nu_1$
- $(\ell_1, \nu_1) \mapsto_d (\ell_2, \nu_2)$ и $(\ell_1, \nu_1) \mapsto_{d'} (\ell, \nu)$, где $0 < d' < d$

Пусть $tr : (\ell_0, \nu_0) \to (\ell_1, \nu_1) \to \ldots$ — прогрессивное вычисление корректного автомата A

Тогда $M, tr \models_T \psi \mathbf{U} \chi$ в том и только том случае, если существуют число i, $i \in \mathbb{N}_0$, и конфигурация $(\ell_i, \nu_i + d)$, промежуточная для (ℓ_i, ν_i) и (ℓ_{i+1}, ν_{i+1}), такие что

- $M(A), (\ell_i, \nu_i + d) \models \chi$
- для любой конфигурации $(\ell_i, \nu_i + d')$, $d' < d$, промежуточной для пары (ℓ_i, ν_i), $(\ell_i, \nu_i + d)$, верно $M(A), (\ell_i, \nu_i + d') \models \psi$
- для любой конфигурации $(\ell_j, \nu_j + d'')$, промежуточной для пары (ℓ_j, ν_j), (ℓ_{j+1}, ν_{j+1}), где $0 \leq j < i$, верно $M(A), (\ell_j, \nu_j + d'') \models \psi$

Задача model checking для TCTL

Для заданного корректного временного автомата A и заданной TCTL-формулы φ проверить справедливость соотношения

$$A \models_T \varphi$$

Конец лекции 9