

Математическая логика и логическое программирование

mk.cs.msu.ru → Лекционные курсы
→ Математическая логика и логическое программирование (3-й поток)

Блок 52

Темпоральные логики

Лектор:

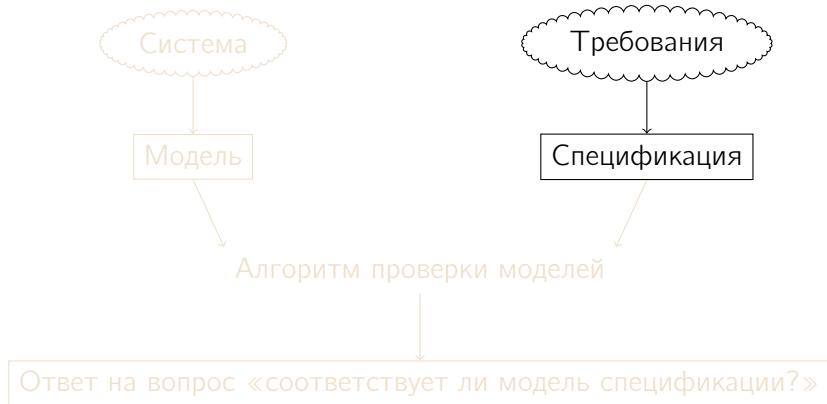
Подымов Владислав Васильевич

E-mail:

valdus@yandex.ru

ВМК МГУ, 2025, сентябрь–декабрь

Вступление



Вступление

Самый простой логический язык записи спецификаций, совместимый с МК и на котором можно основывать другие языки — это язык **логики высказываний**

Формулу логики высказываний можно понимать как запись соотношения между значениями АВ в конкретном состоянии

Но система в процессе вычисления (с течением времени) изменяет свои состояния, и соответственно изменяются значения АВ

Темпоральная логика — это логика, в которой учитывается изменение значения высказываний с течением времени и содержатся средства записи и анализа

взаимосвязи значений высказываний в разные моменты времени

Операции языка темпоральной логики, как правило, делятся на «классические» (отвечающие логике высказываний или предикатов) и **темпоральные**, позволяющие рассуждать о разных моментах времени

Рассмотрим ключевые языки темпоральной логики, применяющиеся в области проверки моделей

CTL*: синтаксис и семантика

Начнём с языка CTL*,

оставив расшифровку этой аббревиатуры на потом

Синтаксис формул CTL* над множеством АВ $\mathcal{AP}$ задаётся БНФ

$$\begin{aligned}\Phi &::= \top \mid p \mid (\Phi \ \& \ \Phi) \mid (\Phi \ \vee \ \Phi) \mid (\neg \Phi) \mid (\Phi \rightarrow \Phi) \\ &\quad \mid (\mathbf{A}\varphi) \mid (\mathbf{E}\varphi) \\ \varphi &::= \Phi \mid (\varphi \ \& \ \varphi) \mid (\varphi \ \vee \ \varphi) \mid (\neg \varphi) \mid (\varphi \rightarrow \varphi) \\ &\quad \mid (\mathbf{F}\varphi) \mid (\mathbf{G}\varphi) \mid (\mathbf{X}\varphi) \mid (\varphi \mathbf{U} \varphi),\end{aligned}$$

где

- ▶ Φ — формула CTL*, или, по-другому, формула состояния,
- ▶ φ — формула пути и
- ▶ $p \in \mathcal{AP}$

Для двух видов формул соответственно определяется

два вида выполнимости:

- ▶ Выполнимость формулы состояния Φ
в заданном состоянии s МК M : $M, s \models \Phi$
- ▶ Выполнимость формулы пути φ
на заданном бесконечном пути π в МК M : $M, \pi \models \varphi$

CTL*: синтаксис и семантика

Начнём с языка CTL*,

оставив расшифровку этой аббревиатуры на потом

Синтаксис формул CTL* над множеством АВ $\mathcal{P}$ задаётся БНФ

$$\begin{aligned}\Phi & ::= \top \mid p \mid (\Phi \& \Phi) \mid (\Phi \vee \Phi) \mid (\neg \Phi) \mid (\Phi \rightarrow \Phi) \\ & \quad \mid (\mathbf{A}\varphi) \mid (\mathbf{E}\varphi) \\ \varphi & ::= \Phi \mid (\varphi \& \varphi) \mid (\varphi \vee \varphi) \mid (\neg \varphi) \mid (\varphi \rightarrow \varphi) \\ & \quad \mid (\mathbf{F}\varphi) \mid (\mathbf{G}\varphi) \mid (\mathbf{X}\varphi) \mid (\varphi \mathbf{U} \varphi),\end{aligned}$$

Приоритеты операций: $\neg$, **A**, **E**, **F**, **G** и **X**; затем **U**;

затем остальные операции с обычными приоритетами

Символ $\top$, связки $\&$, $\vee$, $\neg$, $\rightarrow$ и АВ p

имеют привычный содержательный смысл

Буквы **A** и **E** — это **кванторы пути**:

- ▶ «**A** φ » = «для любого бесконечного пути, исходящего из текущего состояния, верно φ » и
- ▶ «**E** φ » = «существует бесконечный путь, исходящий из текущего состояния и такой что для него верно φ »

CTL*: синтаксис и семантика

Начнём с языка CTL*,

оставив расшифровку этой аббревиатуры на потом

Синтаксис формул CTL* над множеством АВ $\mathcal{AP}$ задаётся БНФ

$$\begin{aligned}\Phi &::= \top \mid p \mid (\Phi \& \Phi) \mid (\Phi \vee \Phi) \mid (\neg \Phi) \mid (\Phi \rightarrow \Phi) \\ &\quad \mid (\mathbf{A}\varphi) \mid (\mathbf{E}\varphi) \\ \varphi &::= \Phi \mid (\varphi \& \varphi) \mid (\varphi \vee \varphi) \mid (\neg \varphi) \mid (\varphi \rightarrow \varphi) \\ &\quad \mid (\mathbf{F}\varphi) \mid (\mathbf{G}\varphi) \mid (\mathbf{X}\varphi) \mid (\varphi \mathbf{U} \varphi),\end{aligned}$$

Буквы **F**, **G**, **X**, **U** — это **темпоральные операторы**:

- ▶ «**F** φ » = «когда-нибудь, рано или поздно, станет верно φ » (in **F**uture)
- ▶ «**G** φ » = «всегда будет верно φ » (**G**lobally)
- ▶ «**X** φ » = «в следующем состоянии будет верно φ » (ne**X**t step)
- ▶ « $\varphi \mathbf{U} \psi$ » = «когда-нибудь станет верно ψ ,
а пока оно не стало верным, обязательно верно φ » (**U**ntil)

CTL*: синтаксис и семантика

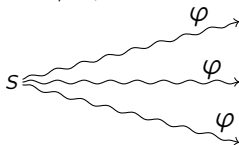
Отношения выполнимости формул для МК $M = (S, S_0, \mapsto, L)$, состояния s и бесконечного пути π заданы следующими правилами:

- ▶ Соотношение $M, s \models \mathbf{t}$ верно всегда
- ▶ $M, s \models p$, где $p \in \mathcal{AP} \iff p \in L(s)$
- ▶ $M, s \models \Phi \ \& \ \Psi \iff M, s \models \Phi \text{ и } M, s \models \Psi$
- ▶ $M, \pi \models \varphi \ \& \ \psi \iff M, \pi \models \varphi \text{ и } M, \pi \models \psi$
- ▶ $M, s \models \Phi \ \vee \ \Psi \iff M, s \models \Phi \text{ или } M, s \models \Psi$
- ▶ $M, \pi \models \varphi \ \vee \ \psi \iff M, \pi \models \varphi \text{ или } M, \pi \models \psi$
- ▶ $M, s \models \neg \Phi \iff M, s \not\models \Phi$
- ▶ $M, \pi \models \neg \varphi \iff M, \pi \not\models \varphi$
- ▶ $M, s \models \Phi \rightarrow \Psi \iff M, s \not\models \Phi \text{ или } M, s \models \Psi$
- ▶ $M, \pi \models \varphi \rightarrow \psi \iff M, \pi \not\models \varphi \text{ или } M, \pi \models \psi$
- ▶ $M, \pi \models \Phi$ для формулы состояния $\Phi \iff M, \pi[1] \models \Phi$
 - ▶ $\pi[i]$ — i -е состояние пути π **при нумерации с единицы**

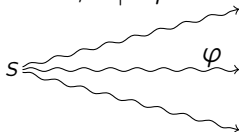
CTL*: синтаксис и семантика

Отношения выполнимости формул для МК $M = (S, S_0, \mapsto, L)$, состояния s и бесконечного пути π заданы следующими правилами:

- ▶ $M, s \models \mathbf{A}\varphi \iff$ для любого бесконечного пути π в M , исходящего из s , верно $M, \pi \models \varphi$



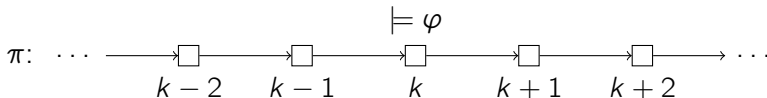
- ▶ $M, s \models \mathbf{E}\varphi \iff$ существует бесконечный путь в M , исходящий из s и такой что $M, \pi \models \varphi$



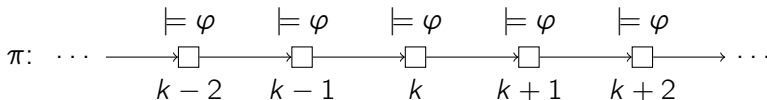
CTL*: синтаксис и семантика

Отношения выполнимости формул для МК $M = (S, S_0, \mapsto, L)$, состояния s и бесконечного пути π заданы следующими правилами:

- ▶ $M, \pi \models \mathbf{F}\varphi \Leftrightarrow$ существует номер k , $k \geq 1$, такой что $M, \pi^k \models \varphi$
- ▶ π^k — суффикс пути π , начинающийся с k -го состояния



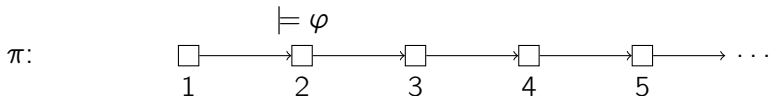
- ▶ $M, \pi \models \mathbf{G}\varphi \Leftrightarrow$ для любого номера k , $k \geq 1$, верно $M, \pi^k \models \varphi$



CTL*: синтаксис и семантика

Отношения выполнимости формул для МК $M = (S, S_0, \mapsto, L)$, состояния s и бесконечного пути π заданы следующими правилами:

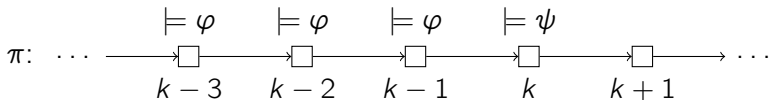
$$\triangleright M, \pi \models \mathbf{X}\varphi \Leftrightarrow M, \pi^2 \models \varphi$$



$$\triangleright M, \pi \models \varphi \mathbf{U} \psi \Leftrightarrow \text{существует номер } k, k \geq 1, \text{ такой что}$$

$$\triangleright M, \pi^k \models \psi \text{ и}$$

$$\triangleright \text{для любого номера } m, \text{ такого что } 1 \leq m < k, \text{ верно } M, \pi^m \models \varphi$$



CTL*: постановка задачи проверки моделей

Формула CTL* φ выполняется на МК M ($M \models \varphi$),
если она выполняется в любом начальном состоянии системы M

Задача проверки моделей для CTL* формулируется так:

для заданной **конечной** МК M

и заданной формулы φ CTL*

проверить справедливость соотношения $M \models \varphi$

На практике обычно рассматривается не весь язык CTL*, а два его более простых для понимания фрагмента

«LTL» расшифровывается как «Linear Time Logic»,
логики линейного времени

Формула LTL — это формула CTL* вида $\mathbf{A}\varphi$,
в которой подформула φ не содержит ни одного квантора пути

То есть устроенная так:

$$\begin{aligned}\Phi &::= (\mathbf{A}\varphi) \\ \varphi &::= \text{tt} \mid p \mid (\varphi \& \varphi) \mid (\varphi \vee \varphi) \mid (\neg\varphi) \mid (\varphi \rightarrow \varphi) \\ &\quad \mid (\mathbf{F}\varphi) \mid (\mathbf{G}\varphi) \mid (\mathbf{X}\varphi) \mid (\varphi \mathbf{U} \varphi)\end{aligned}$$

Формулы LTL предназначены для формализации
высказываний о вычислениях модели:

«все вычисления модели должны обладать следующим свойством: ...»

Примеры спецификаций на языке LTL:

- ▶ Данные на печать передаются не более чем одним принтером:

$$\mathbf{AG}\neg(print_1 \ \& \ print_2)$$

- ▶ После завершения процедур начисления стипендии и зарплаты на счёте будет ровно 1 001 000 рублей:

$$\mathbf{AFG}p_{\text{счёт}=1\ 001\ 000}$$

- ▶ На следующем шаге после оплаты напитка он будет выдаваться:

$$\mathbf{AG}(\neg paid \ \& \ \mathbf{X}paid \rightarrow \mathbf{XX}(serve_t \vee serve_c))$$

- ▶ Если компьютер достаточно часто посылает запрос на печать, то рано или поздно он начнёт печатать:

$$\mathbf{A}(\mathbf{GF}request \rightarrow \mathbf{F}print)$$

- ▶ Если идёт печать, то сеанс печати рано или поздно завершится, и до завершения принтер будет занят:

$$\mathbf{AG}(print \rightarrow busy\mathbf{U}\neg print)$$

CTL

«CTL» расшифровывается как «Computational Tree Logic», *логика деревьев вычислений*, или, если хочется подчеркнуть различия с LTL, — **логика ветвящегося времени** в противовес линейному

Формула CTL — это формула CTL* частного вида, отвечающего БНФ

$$\begin{aligned}\Phi &::= \mathfrak{t} \mid p \mid (\Phi \& \Phi) \mid (\Phi \vee \Phi) \mid (\neg \Phi) \mid (\Phi \rightarrow \Phi) \\ &\quad \mid (\mathbf{A}\varphi) \mid (\mathbf{E}\varphi) \\ \varphi &::= (\mathbf{F}\Phi) \mid (\mathbf{G}\Phi) \mid (\mathbf{X}\Phi) \mid (\Phi \mathbf{U} \Phi)\end{aligned}$$

То есть в формуле CTL под квантором пути обязательно располагается темпоральный оператор, и под ним — формула состояния

Иными словами, в формуле CTL кванторы пути и темпоральные операторы используются только в парах:

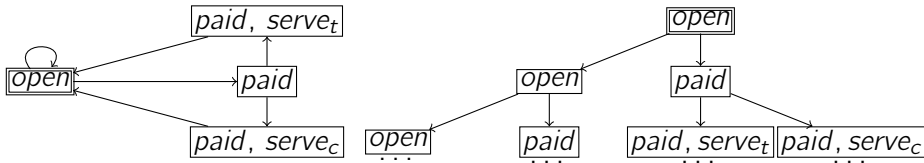
AG, EG, AF, EF, AX, EX, AU, EU

CTL

Лучше понять смысл формул CTL помогает **развёртка** МК — бесконечное дерево, устроенное так:

- ▶ Корень — это выбранное начальное состояние
- ▶ Вершина развёртки отвечает конечному пути в МК и размечена теми же АВ, что и последняя вершина пути
- ▶ Дуга $v_1 \rightarrow v_2$ в развёртке означает, что путь v_1 можно продолжить до пути v_2 , добавив один переход

Например, ниже изображены МК и фрагмент её развёртки



CTL

Примеры спецификаций на языке CTL для кофейного автомата:

- ▶ В самом начале работы автомата приёмник монет открыт, в нём нет монеты, и автомат ничего не выдаёт:

$$open \ \& \ \neg paid \ \& \ \neg serve_t \ \& \ \neg serve_c$$

- ▶ Нельзя сделать так, чтобы автомат выдал напиток, не имея монеты в приёмнике:

$$\neg \mathbf{EF}(\neg paid \ \& \ (serve_c \vee serve_t))$$

- ▶ Если в приёмнике есть монета, то рано или поздно он выдаст напиток ...

$$\mathbf{AG}(paid \rightarrow \mathbf{AF}(serve_c \vee serve_t))$$

- ▶ ... но этот напиток не обязан быть чаем ...

$$\mathbf{EF}(paid \ \& \ \mathbf{EG} \neg serve_t)$$

- ▶ ... но при желании можно, опустив монету в приёмник, получить чай

$$\mathbf{AG}(\neg paid \rightarrow \mathbf{AX}(paid \rightarrow \mathbf{EF} serve_t))$$