Математическая логика

mk.cs.msu.ru → Лекционные курсы → Математическая логика (318, 319/2, 241, 242)

Блок 47

Проверка моделей
относительно логики деревьев вычислений
(CTL model checking)

Лектор:
Подымов Владислав Васильевич

E-mail:
valdus@yandex.ru

ВМК МГУ, 2020/2021, весенний семестр

Вступление

Рассмотрим такую систему, состоящую из кофейного автомата и покупателя

Кофейный автомат имеет приёмник для монет и кнопки “чай” и “кофе” и запрограммирован следующим образом:

- Ожидается монета, приёмник открыт
- После приёма монеты:
  - Приёмник закрывается, ожидается нажатие на одну из кнопок
  - После нажатия на кнопку соответствующий напиток выдаётся покупателю, после чего монета удаляется из приёмника и автомат переходит в режим ожидания

Покупатель в зависимости от своего желания может кидать монету в приёмник и нажимать на кнопки

Как можно устроить модель такой системы?

Какие требования разумно было бы предъявить к такой системе?

Как проверить, удовлетворяет ли модель этим требованиям?

Размеченные системы переходов

Ожидается монета, приёмник открыт

После приёма монеты:

- Приёмник закрывается, ожидается нажатие на одну из кнопок
- После нажатия на кнопку соответствующий напиток выдаётся покупателю, после чего монета удаляется из приёмника и автомат переходит в режим ожидания

Поведение кофейного автомата можно представить себе так.

В каждый *момент времени* он находится в некотором **состоянии**: ожидает приёма монеты, ожидает нажатия кнопки, выдаёт чай, выдаёт кофе

В некоторые моменты времени он может **переходить** (или обязательно переходит) из одного состояния в другое согласно своему устройству

В некоторый момент автомат запускается в некотором (**начальном**) состоянии

Размеченные системы переходов

Ожидается монета, приёмник открыт

После приёма монеты:

- Приёмник закрывается, ожидается нажатие на одну из кнопок
- После нажатия на кнопку соответствующий напиток выдаётся покупателю, после чего монета удаляется из приёмника и автомат переходит в режим ожидания

Поведение кофейного автомата можно представить себе так.

Для проверки правильности работы автомата выделяется набор *простых* свойств (**атомарных высказываний**), и для каждого состояния и каждого свойства говорится, обладает ли состояние этим свойством (истинно ли это высказывание в состоянии)

Например: приёмник открыт, в приёмнике есть монета, выдаётся чай, выдаётся кофе

Размеченные системы переходов

AP — так будем обозначать заранее заданное конечное множество атомарных высказываний

2^{AP} — множество всех подмножеств AP

Размеченной системой переходов (СП) называется система $(S, S_0, \to, \rho)$, где:

- S — конечное множество состояний
- S_0 — множество начальных состояний, $S_0 \subseteq S$
- $\to \subseteq S \times S$ — **тотальное** отношение переходов
- $\rho : S \to 2^{AP}$ — функция разметки

Тотальность отношения переходов означает, что для любого состояния s существует состояние s', такое что $s \to s'$ (то есть из любого состояния можно выполнить хотя бы один переход)

Функция разметки ставит в соответствие состоянию s множество $\rho(s)$ всех атомарных высказываний, истинных в этом состоянии

Этот вид СП имеет особое название: модели Крипке — но это название использовать не будем, чтобы не путать СП с интерпратациями формул в модальных логиках

Размеченные системы переходов

Пример: СП кофейного автомата

Атомарные высказывания:

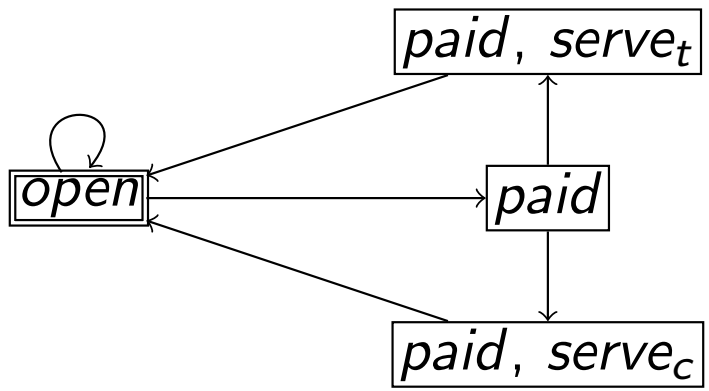
$open$ = “приёмник открыт”

$paid$ = “в приёмнике есть монета”

$serve_t$ = “выдаётся чай”

$serve_c$ = “выдаётся кофе”

СП:



□ — состояние

▣ — начальное состояние

Атомарные высказывания, истинные в состоянии, записаны внутри этого состояния

Логика деревьев вычислений (CTL)

Фрагмент этой логики уже возникал в качестве иллюстрации *блоке 44*

Синтаксис ctl-формул (над множеством AP) задаётся следующей БНФ:

$$\varphi ::= \mathbb{t} \mid a \mid (\varphi \,\&\, \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \to \varphi) \mid (\neg\varphi) \mid$$
$$(\mathbf{AF}\varphi) \mid (\mathbf{AG}\varphi) \mid (\mathbf{AX}\varphi) \mid (\varphi\mathbf{AU}\varphi) \mid$$
$$(\mathbf{EF}\varphi) \mid (\mathbf{EG}\varphi) \mid (\mathbf{EX}\varphi) \mid (\varphi\mathbf{EU}\varphi) \mid$$

Здесь φ — ctl-формула и $a \in \mathrm{AP}$

Приоритеты операций по убыванию:

AF, **AG**, **AX**, **EF**, **EG**, **EX** и $\neg$;

затем **AU** и **EU**; затем $\&$; затем $\vee$; затем $\to$

Зададим семантику ctl-формул, адаптировав семантику из *блока 44* к СП

Для этого определим отношение выполнимости ctl-формулы φ в состоянии s СП $TS = (S, S_0, \to, \rho)$: $TS, s \models \varphi$

Путём в СП $TS = (S, S_0, \to, \rho)$ (из состояния s_1) назовём бесконечную последовательность состояний вида

$$s_1 \to s_2 \to s_3 \to \ldots$$

Логика деревьев вычислений (CTL)

- всегда верно $TS, s \models \mathbb{t}$
- $TS, s \models a \Leftrightarrow a \in \rho(s)$, если $a \in \mathrm{AP}$
- $TS, s \models \psi_1 \,\&\, \psi_2 \Leftrightarrow TS, s \models \psi_1$ и $TS, s \models \psi_2$
- $TS, s \models \psi_1 \vee \psi_2 \Leftrightarrow TS, s \models \psi_1$ или $TS, s \models \psi_2$
- $TS, s \models \psi_1 \to \psi_2 \Leftrightarrow TS, s \not\models \psi_1$ или $TS, s \models \psi_2$
- $TS, s \models \neg\psi \Leftrightarrow TS, s \not\models \psi$
- $TS, s \models \mathbf{AF}\psi \Leftrightarrow$ в любом пути из s существует состояние s', для которого верно $TS, s' \models \psi$
- $TS, s \models \mathbf{EF}\psi \Leftrightarrow$ хотя бы в одном пути из s существует состояние s', для которого верно $TS, s' \models \psi$
- $TS, s \models \mathbf{AG}\psi \Leftrightarrow$ для любого состояния s' любого пути из s верно $TS, s' \models \psi$
- $TS, s \models \mathbf{EG}\psi \Leftrightarrow$ существует путь из s, для любого состояния s' которого верно $TS, s' \models \psi$

Логика деревьев вычислений (CTL)

- $TS, s \models \mathbf{AX}\psi \Leftrightarrow$ для любого состояния s', такого что $s \to s'$, верно $TS, s' \models \psi$
- $TS, s \models \mathbf{EX}\psi \Leftrightarrow$ существует состояние s', такое что $s \to s'$ и верно $TS, s' \models \psi$
- $TS, s \models \psi_1 \mathbf{AU} \psi_2 \Leftrightarrow$ в любом пути из s существует состояние s', такое что
  - $TS, s' \models \psi_2$ и
  - для любого состояния s'' от начала пути до s' (не включительно) верно $TS, s'' \models \psi_1$
- $TS, s \models \psi_1 \mathbf{EU} \psi_2 \Leftrightarrow$ существуют путь из s и состояние s' этого пути, такие что
  - $TS, s' \models \psi_2$ и
  - для любого состояния s'' от начала пути до s' (не включительно) верно $TS, s'' \models \psi_1$

Логика деревьев вычислений (CTL)

Что это всё означает

Представим себе, как систему, заданную в виде СП $TS = (S, S_0, \to, \rho)$ и в результате выполнения в дискретном времени достигшую состояния s

Тогда символы ctl-формул можно *содержательно* трактовать как следующие высказывания о состояниях из S:

- $\mathbf{A}\Phi$ = "как бы ни продолжала выполняться система, будет верно Φ"
- $\mathbf{E}\Phi$ = "существует возможность продолжить выполнение системы так, чтобы было верно Φ"
- $\mathbf{F}\psi$ = "рано или поздно станет верно ψ"
- $\mathbf{G}\psi$ = "в будущем всегда будет верно ψ"
- $\mathbf{X}\psi$ = "после следующего перехода будет верно ψ"
- $\psi_1 \mathbf{U} \psi_2$ = "рано или поздно станет верно ψ_2, а до тех пор будет верно ψ_1"

Логика деревьев вычислений (CTL)

Например, в виде ctl-формул можно записать такие свойства кофейного автомата:

- В начале выполнения 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

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

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

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

- ... 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

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

Задача проверки моделей относительно CTL

Ctl-формула φ выполняется на СП TS ($TS \models \varphi$), если она выполняется в каждом начальном состоянии TS

Задача проверки моделей (model checking) относительно CTL формулируется так:

Для заданной СП TS и заданной ctl-формулы φ проверить справедливость соотношения

$$TS \models \varphi$$

Будем обозначать эту задачу так: MC-CTL

Решение задачи MC-CTL

Утверждение(*об упрощении ctl-формул*) Для любой СП TS и любой ctl-формулы φ верно:

- $TS \models \varphi_1 \,\&\, \varphi_2 \Leftrightarrow TS \models \neg(\neg\varphi_1 \vee \neg\varphi_2)$
- $TS \models \varphi_1 \to \varphi_2 \Leftrightarrow TS \models \neg\varphi_1 \vee \varphi_2$
- $TS \models \mathbf{EF}\varphi \Leftrightarrow TS \models \mathbb{t}\mathbf{EU}\varphi$
- $TS \models \mathbf{AX}\varphi \Leftrightarrow TS \models \neg\mathbf{EX}\neg\varphi$
- $TS \models \mathbf{AG}\varphi \Leftrightarrow TS \models \neg\mathbf{EF}\neg\varphi$
- $TS \models \mathbf{AF}\varphi \Leftrightarrow TS \models \neg\mathbf{EG}\neg\varphi$
- $TS \models \varphi_1\mathbf{AU}\varphi_2 \Leftrightarrow TS \models \mathbf{AF}\varphi_2 \,\&\, \neg((\neg\varphi_2)\mathbf{EU}(\neg\varphi_1 \,\&\, \neg\varphi_2))$

Доказательство. Попробуйте самостоятельно

Значит, для решения задачи MC-CTL можно *без ограничения общности* рассматривать ctl-формулы, содержащие **только** операции

$$\vee,\ \neg,\ \mathbf{EX},\ \mathbf{EG} \text{ и } \mathbf{EU}$$

Такие формулы будем дальше называть упрощёнными

Решение задачи MC-CTL

Опишем алгоритм Check(TS, φ) проверки соотношения $TS \models \varphi$ для СП $TS = (S, S_0, \to, \rho)$ и упрощённой ctl-формулы φ (*или нечто очень близкое к алгоритму*) в C-подобном псевдокоде:

```
bool Check(TS, ψ) {
  return &&_{s0∈S0} Check(TS, s0, φ);
}

bool Check(TS, s, ψ) {
  if(φ = ⊤) return true;
  if(φ = a ∈ AP) return a ∈ AP;
  if(φ = ¬ψ) return !Check(TS, s, ψ);
  if(φ = ψ1 ∨ ψ2) return Check(TS, s, ψ1)
                      || Check(TS, s, ψ2);
  if(φ = EXψ) return CheckEX(TS, s, ψ);
  if(φ = EGψ) return CheckEG(TS, s, ψ);
  if(φ = ψ1 EU ψ2) return CheckEU(TS, s, ψ1, ψ2);
}
```

Решение задачи MC-CTL

```
bool CheckEX(TS, s, φ) {
  V = { s' | s' ∈ S && Check(TS, s', φ)};
  return ∃s' : s → s' && s' ∈ V;
}
```

```
bool CheckEU(TS, s, φ1, φ2) {
  V1 = { s' | s' ∈ S && Check(TS, s', φ1)};
  V2 = { s' | s' ∈ S && Check(TS, s', φ2)};
  return ∃s1 → s2 → ··· → sk :
    s1 → s && sk ∈ V2 && {s1, ..., sk−1} ⊆ V1;
}
```

Решение задачи MC-CTL

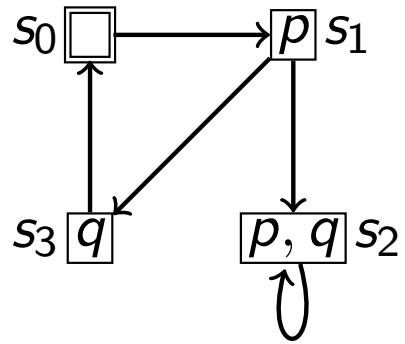
Утверждение. $TS, s \models \mathbf{EG}\varphi \Leftrightarrow$ **в** TS **как в размеченном графе существуют путь** $s \to \cdots \to s'$ **и цикл** $s' \to \cdots \to s'$**, такие что для каждой вершины** s'' **этих пути и цикла верно** $TS, s' \models \varphi$

```
bool CheckEG(TS, s, φ) {
  V = { s' | s' ∈ S && Check(TS, s', φ)};
  return ∃s_1 → s_2 → ··· → s_k :
    s_1 = s && {s_1, ..., s_k} ⊆ V &&
    ∃i : (0 ≤ i < k && s_i = s_k);
}
```

Решение задачи MC-CTL

Пример: $\varphi = \mathbf{EG}((\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q))$

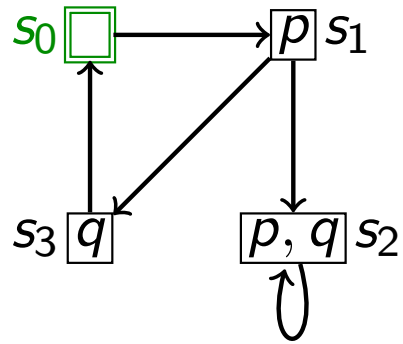
TS:



Решение задачи MC-CTL

Пример: $\varphi = \mathbf{EG}((\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q))$

TS:

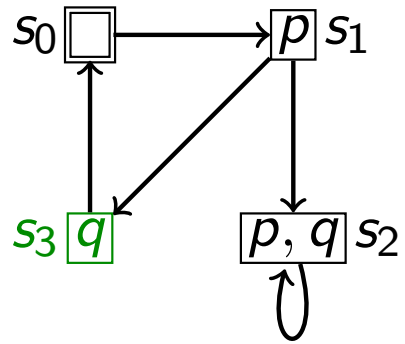


$TS, s_0 \models \neg p \,\&\, \neg q$

Решение задачи MC-CTL

Пример: $\varphi = \mathbf{EG}((\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q))$

TS:



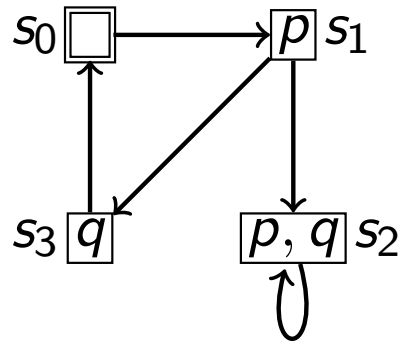
$TS, s_0 \models \neg p \,\&\, \neg q$

$TS, s_3 \models \neg p$

Решение задачи MC-CTL

Пример: $\varphi = \mathbf{EG}((\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q))$

TS:



$TS, s_0 \models \neg p \,\&\, \neg q$

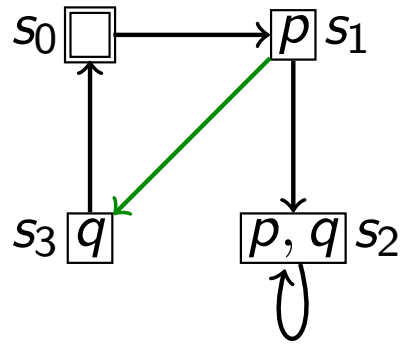
$TS, s_3 \models \neg p$

$TS, s_3 \models \neg p \vee \mathbf{EX}\neg p$

Решение задачи MC-CTL

Пример: $\varphi = \mathbf{EG}((\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q))$

TS:



$TS, s_0 \models \neg p \,\&\, \neg q$

$TS, s_3 \models \neg p$

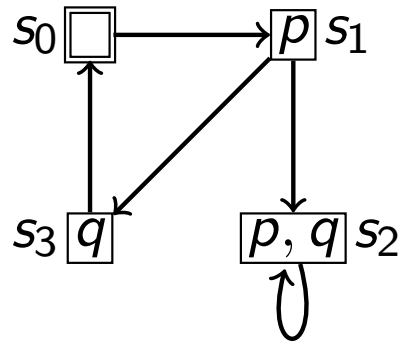
$TS, s_3 \models \neg p \vee \mathbf{EX}\neg p$

$TS, s_1 \models \mathbf{EX}\neg p$

Решение задачи MC-CTL

Пример: $\varphi = \mathbf{EG}((\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q))$

TS:



$TS, s_0 \models \neg p \,\&\, \neg q$

$TS, s_3 \models \neg p$

$TS, s_3 \models \neg p \vee \mathbf{EX}\neg p$

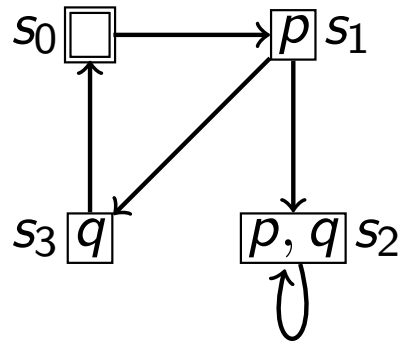
$TS, s_1 \models \mathbf{EX}\neg p$

$TS, s_1 \models \neg p \vee \mathbf{EX}\neg p$

Решение задачи MC-CTL

Пример: $\varphi = \mathbf{EG}((\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q))$

TS:



$TS, s_0 \models \neg p \,\&\, \neg q$

$TS, s_3 \models \neg p$

$TS, s_3 \models \neg p \vee \mathbf{EX}\neg p$

$TS, s_1 \models \mathbf{EX}\neg p$

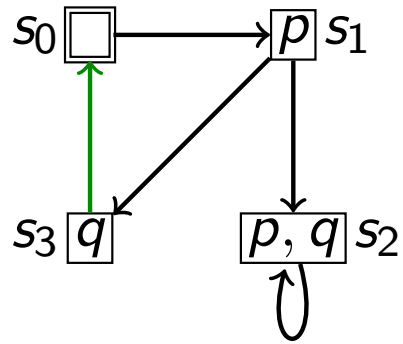
$TS, s_1 \models \neg p \vee \mathbf{EX}\neg p$

$TS, s_0 \models (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q)$

Решение задачи MC-CTL

Пример: $\varphi = \mathbf{EG}((\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q))$

TS:

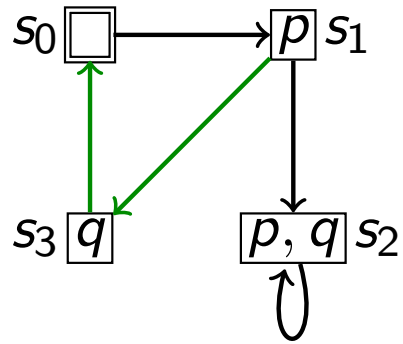


$$TS, s_0 \models \neg p \,\&\, \neg q$$
$$TS, s_3 \models \neg p$$
$$TS, s_3 \models \neg p \vee \mathbf{EX}\neg p$$
$$TS, s_1 \models \mathbf{EX}\neg p$$
$$TS, s_1 \models \neg p \vee \mathbf{EX}\neg p$$
$$TS, s_0 \models (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q)$$
$$TS, s_3 \models (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q)$$

Решение задачи MC-CTL

Пример: $\varphi = \mathbf{EG}((\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q))$

TS:

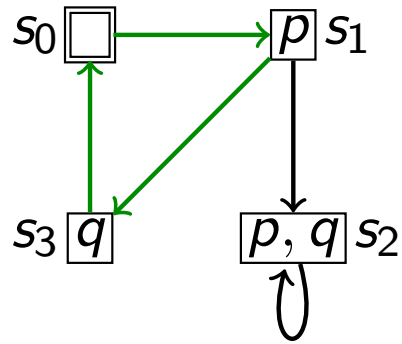


$$
\begin{aligned}
&TS, s_0 \models \neg p \,\&\, \neg q \\
&TS, s_3 \models \neg p \\
&TS, s_3 \models \neg p \vee \mathbf{EX}\neg p \\
&TS, s_1 \models \mathbf{EX}\neg p \\
&TS, s_1 \models \neg p \vee \mathbf{EX}\neg p \\
&TS, s_0 \models (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q) \\
&TS, s_3 \models (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q) \\
&TS, s_1 \models (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q)
\end{aligned}
$$

Решение задачи MC-CTL

Пример: $\varphi = \mathbf{EG}((\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q))$

TS:

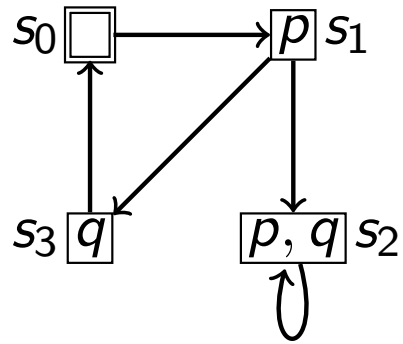


$$
\begin{aligned}
TS, s_0 &\models \neg p \,\&\, \neg q \\
TS, s_3 &\models \neg p \\
TS, s_3 &\models \neg p \vee \mathbf{EX}\neg p \\
TS, s_1 &\models \mathbf{EX}\neg p \\
TS, s_1 &\models \neg p \vee \mathbf{EX}\neg p \\
TS, s_0 &\models (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q) \\
TS, s_3 &\models (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q) \\
TS, s_1 &\models (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q) \\
TS, s_0 &\models \varphi
\end{aligned}
$$

Решение задачи MC-CTL

Пример: $\varphi = \mathbf{EG}((\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q))$

TS:



$$
\begin{aligned}
TS, s_0 &\models \neg p \,\&\, \neg q \\
TS, s_3 &\models \neg p \\
TS, s_3 &\models \neg p \vee \mathbf{EX}\neg p \\
TS, s_1 &\models \mathbf{EX}\neg p \\
TS, s_1 &\models \neg p \vee \mathbf{EX}\neg p \\
TS, s_0 &\models (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q) \\
TS, s_3 &\models (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q) \\
TS, s_1 &\models (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}(\neg p \,\&\, \neg q) \\
TS, s_0 &\models \varphi
\end{aligned}
$$

Вопрос на понимание:

А всегда ли этот алгоритм работает правильно, и какова его сложность?