Математическая логика

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

Блок 48

Проверка моделей
относительно логики деревьев вычислений
(CTL model checking)

Лектор:
Подымов Владислав Васильевич
E-mail:
valdus@yandex.ru

ВМК МГУ, 2021/2022, 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 монеты

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 монеты

Поведение кофейного автомата можно представить себе так.

В каждый *момент времени* он находится в некотором **состоянии**: ожидает монету, 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 времени 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 монеты

Поведение кофейного автомата можно представить себе так.

Для проверки правильности работы автомата можно выделить *представляющие интерес* и при этом *достаточно простые* свойства его состояний (**атомарные высказывания**):
приёмник открыт, монета принята, выдаётся чай, выдаётся кофе, ...

AP — так будем обозначать заранее заданное конечное множество атомарных высказываний

2^{AP} — множество всех подмножеств AP

Размеченные системы переходов

Размеченной системой переходов (СП)[1] над множеством AP называется система $TS = (S, S_0, \to, \rho)$, 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 состояний высказываниями, истинными в этом состоянии

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)

Путём в TS (из состояния s_1) будем называть бесконечную последовательность состояний вида

$$s_1 \to s_2 \to s_3 \to \dots$$

1 Такая СП имеет особое название: модель Крипке — но так же называются интерпретации формул модальных логик, поэтому будем использовать название «СП»

Размеченные системы переходов

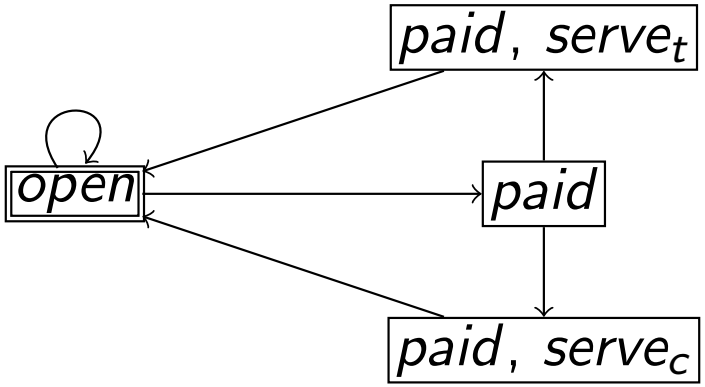
Пример: СП кофейного автомата

Атомарные высказывания:

$open$ = «приёмник открыт» $serve_t$ = «выдаётся чай»

$paid$ = «в приёмнике есть монета» $serve_c$ = «выдаётся кофе»

СП:



□ — состояние

▣ — начальное состояние

Высказывания, размечающие состояние, записаны внутри этого состояния

Логика деревьев вычислений (CTL)

Фрагмент этой логики уже возникал в качестве иллюстрации блоке 45

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

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

(φ — ctl-формула, $p \in \mathrm{AP}$)

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 45 к СП

Для этого определим отношение выполнимости ctl-формулы φ в состоянии s СП TS: $TS, s \models \varphi$

Логика деревьев вычислений (CTL)

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

Символы ctl-формул, введённые в последних двух строках синтаксиса, можно *содержательно* трактовать так:

- **A**Φ: как бы ни продолжала выполняться система, будет верно Φ
- **E**Φ: существует возможность продолжить выполнение системы так, чтобы было верно Φ
- **F**φ: рано или поздно станет верно φ
- **G**φ: в будущем всегда будет верно φ
- **X**φ: после следующего перехода будет верно φ
- φ_1**U**φ_2: рано или поздно станет верно φ_2, а до тех пор будет верно φ_1

Логика деревьев вычислений (CTL)

Правила, задающие отношение выполнимости ctl-формул:

- всегда верно $TS, s \models \mathbb{t}$
- $TS, s \models p \Leftrightarrow p \in \rho(s)$, если $p \in \mathrm{AP}$
- $TS, s \models \psi_1 \,\&\, \psi_2 \Leftrightarrow TS, s \models \psi_1$ 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 пути из s
  и любого состояния s' этого пути 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)

Правила, задающие отношение выполнимости ctl-формул:

- $TS, s \models \mathbf{AX}\psi \Leftrightarrow$ 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 обозначенного 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 обозначенного s' (не включительно) верно $TS, s'' \models \psi_1$

Логика деревьев вычислений (CTL)

В виде ctl-формул можно записать, **например**,
такие свойства 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
  $$\neg \mathbf{EF}(\neg paid \,\&(serve_c \vee serve_t))$$
- Если в приёмнике есть монета,
  то рано или поздно он выдаст напиток ...
  $$\mathbf{AG}(paid \to \mathbf{AF}(serve_c \vee serve_t))$$
- ... 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 (MC-CTL)
формулируется так:

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

$$TS \models \varphi$$

Решение задачи MC-CTL

Утверждение (об упрощении ctl-формул)

Для любых СП TS, её состояния s и ctl-формулы φ верно:

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

Доказательство. Попробуйте самостоятельно

Значит, для решения задачи MC-CTL можно *без ограничения общности* рассматривать только 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

Проверку соотношения $TS \models \varphi$ можно устроить так (`Sat(`TS`, `φ`)`):

- вычислить множество X всех состояний s, таких что $TS, s \models \varphi$
- проверить, вложено ли множество всех начальных состояний TS в X

```
Sat(TS, φ) { let TS = (S, S0, →, ρ); return S0 ⊆ SatSet(TS, φ) }
```

Вычислить такое множество X можно
индукцией (рекурсией) по структуре формулы:

```
SatSet(TS, φ) {
  let TS = (S, S0, →, ρ);
  switch(φ) {
     𝕥:              return S;
     p, p ∈ AP:      return {s | s ∈ S, p ∈ ρ(s)};
     ¬ψ:             return S \ SatSet(TS, ψ);
     ψ1 ∨ ψ2:        return SatSet(TS, ψ1) ∪ SatSet(TS, ψ2);
     EXψ:            return SatSetEX(TS, ψ);
     EGψ:            return SatSetEG(TS, ψ);
     ψ1EUψ2:         return SatSetEU(TS, ψ1, ψ2);
  }
}
```

Решение задачи MC-CTL

По семантике **EX**:

$$TS, s \models \mathbf{EX}\varphi$$
$$\Leftrightarrow$$
существует состояние s', такое что $s \to s'$ и $TS, s' \models \varphi$

```
SatSetEX(TS, φ) {
  let TS = (S, S0, →, ρ);
  X = SatSet(TS, φ);
  return {s | s ∈ S,
            ∃s' ∈ X : s → s'};
}
```

Решение задачи MC-CTL

По семантике **EU**:

$$TS, s \models \varphi_1 \mathbf{EU} \varphi_2$$
$$\Leftrightarrow$$

в TS как в графе существует путь $s_1 \to \cdots \to s_k$ из s, такой что $TS, s_k \models \varphi_2$ и для всех i, $1 \le i \le k-1$, верно $TS, s_i \models \varphi_1$

```
SatSetEU(TS, φ1, φ2) {
  X = SatSet(TS, φ1);
  Y = SatSet(TS, φ2);
  return {s | s ∈ S,
              ∃s1 → ··· → sk :
                1 ≤ k ≤ |S|,
                s1 = s,
                sk ∈ Y,
                {s1, ..., sk−1} ⊆ X}
}
```

Решение задачи MC-CTL

Утверждение. Для любого ориентированного графа G верно:
существует бесконечный путь из вершины s,
проходящий только через вершины заданного подмножества X
$\Leftrightarrow$
существует путь $s \to \cdots \to s' \to \cdots \to s'$,
проходящий только через вершины подмножества X

Решение задачи MC-CTL

Тогда по семантике **EG**:

$$TS, s \models \mathbf{EG}\varphi$$
$$\Leftrightarrow$$

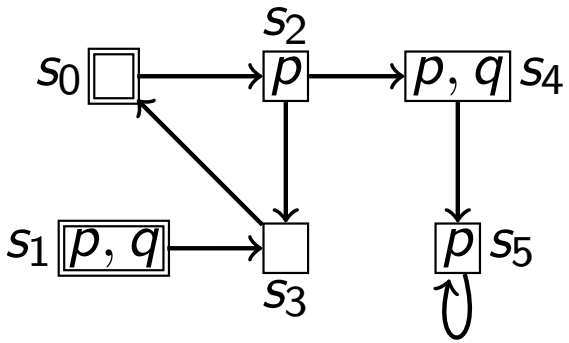
в TS как в графе существует путь $s_1 \to \cdots \to s_k$ из s, проходящий только через вершины s_i, такие что $TS, s_i \models \varphi$, и в котором $s_k = s_i$ хотя бы для одного i, $i < k$.

```
SatSetEG(TS, φ) {
  let TS = (S, S_0, →, ρ);
  X = SatSet(TS, φ);
  return {s | s ∈ S,
              ∃s_1 → s_2 → ··· → s_k :
                1 ≤ k ≤ |S| + 1,
                s_1 = s,
                {s_1, ..., s_k} ⊆ X,
                ∃i : 0 ≤ i < k, s_i = s_k}
}
```

Решение задачи MC-CTL

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

TS:

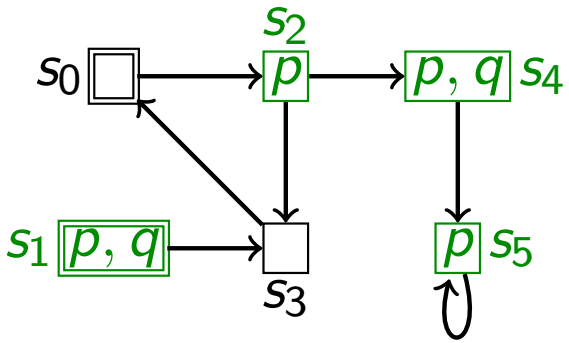


$$\texttt{SatSet}(TS,\ p) = \{s_1, s_2, s_4, s_5\}$$
$$\texttt{SatSet}(TS,\ \neg p) = \{s_0, s_3\}$$
$$\texttt{SatSet}(TS,\ \mathbf{EX}\neg p) = \{s_1, s_2, s_3\}$$
$$\texttt{SatSet}(TS,\ \neg p \vee \mathbf{EX}\neg p) = \{s_0, s_1, s_2, s_3\}$$
$$\texttt{SatSet}(TS,\ q) = \{s_1, s_4\}$$
$$\texttt{SatSet}(TS,\ (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}q) = \{s_0, s_1, s_2, s_3, s_4\}$$
$$\texttt{SatSet}(TS,\ \varphi) = \{s_0, s_1, s_2, s_3\}$$
$$S_0 = \{s_0, s_1\} \subseteq \{s_0, s_1, s_2, s_3\} = \texttt{SatSet}(TS,\ \varphi)$$
$$TS \models \varphi$$

Решение задачи MC-CTL

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

TS:



$\texttt{SatSet}(TS,\ p) = \{s_1, s_2, s_4, s_5\}$

$\texttt{SatSet}(TS,\ \neg p) = \{s_0, s_3\}$

$\texttt{SatSet}(TS,\ \mathbf{EX}\neg p) = \{s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ \neg p \vee \mathbf{EX}\neg p) = \{s_0, s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ q) = \{s_1, s_4\}$

$\texttt{SatSet}(TS,\ (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}q) = \{s_0, s_1, s_2, s_3, s_4\}$

$\texttt{SatSet}(TS,\ \varphi) = \{s_0, s_1, s_2, s_3\}$

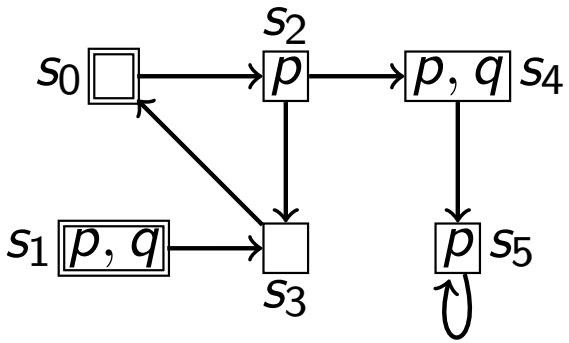
$S_0 = \{s_0, s_1\} \subseteq \{s_0, s_1, s_2, s_3\} = \texttt{SatSet}(TS,\ \varphi)$

$TS \models \varphi$

Решение задачи MC-CTL

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

TS:



$\texttt{SatSet}(TS,\ p) = \{s_1, s_2, s_4, s_5\}$

$\texttt{SatSet}(TS,\ \neg p) = \{s_0, s_3\}$

$\texttt{SatSet}(TS,\ \mathbf{EX}\neg p) = \{s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ \neg p \vee \mathbf{EX}\neg p) = \{s_0, s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ q) = \{s_1, s_4\}$

$\texttt{SatSet}(TS,\ (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}q) = \{s_0, s_1, s_2, s_3, s_4\}$

$\texttt{SatSet}(TS,\ \varphi) = \{s_0, s_1, s_2, s_3\}$

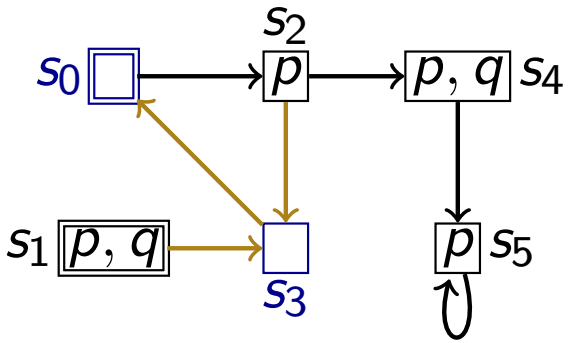
$S_0 = \{s_0, s_1\} \subseteq \{s_0, s_1, s_2, s_3\} = \texttt{SatSet}(TS,\ \varphi)$

$TS \models \varphi$

Решение задачи MC-CTL

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

TS:



$\texttt{SatSet}(TS,\ p) = \{s_1, s_2, s_4, s_5\}$

$\texttt{SatSet}(TS,\ \neg p) = \{s_0, s_3\}$

$\texttt{SatSet}(TS,\ \mathbf{EX}\neg p) = \{s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ \neg p \vee \mathbf{EX}\neg p) = \{s_0, s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ q) = \{s_1, s_4\}$

$\texttt{SatSet}(TS,\ (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}q) = \{s_0, s_1, s_2, s_3, s_4\}$

$\texttt{SatSet}(TS,\ \varphi) = \{s_0, s_1, s_2, s_3\}$

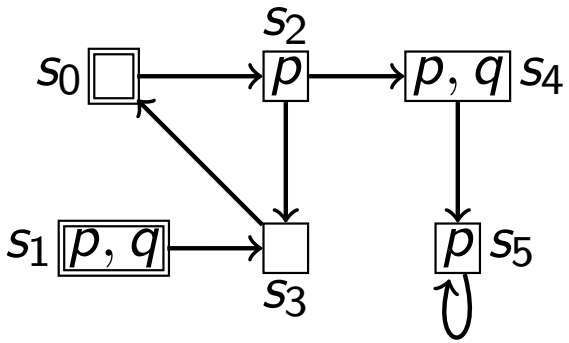
$S_0 = \{s_0, s_1\} \subseteq \{s_0, s_1, s_2, s_3\} = \texttt{SatSet}(TS,\ \varphi)$

$TS \models \varphi$

Решение задачи MC-CTL

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

TS:



$\texttt{SatSet}(TS,\ p) = \{s_1, s_2, s_4, s_5\}$

$\texttt{SatSet}(TS,\ \neg p) = \{s_0, s_3\}$

$\texttt{SatSet}(TS,\ \mathbf{EX}\neg p) = \{s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ \neg p \vee \mathbf{EX}\neg p) = \{s_0, s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ q) = \{s_1, s_4\}$

$\texttt{SatSet}(TS,\ (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}q) = \{s_0, s_1, s_2, s_3, s_4\}$

$\texttt{SatSet}(TS,\ \varphi) = \{s_0, s_1, s_2, s_3\}$

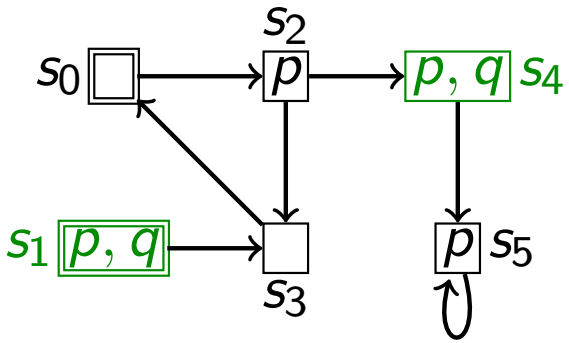
$S_0 = \{s_0, s_1\} \subseteq \{s_0, s_1, s_2, s_3\} = \texttt{SatSet}(TS,\ \varphi)$

$TS \models \varphi$

Решение задачи MC-CTL

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

TS:



$\texttt{SatSet}(TS,\ p) = \{s_1, s_2, s_4, s_5\}$

$\texttt{SatSet}(TS,\ \neg p) = \{s_0, s_3\}$

$\texttt{SatSet}(TS,\ \mathbf{EX}\neg p) = \{s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ \neg p \vee \mathbf{EX}\neg p) = \{s_0, s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ q) = \{s_1, s_4\}$

$\texttt{SatSet}(TS,\ (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}q) = \{s_0, s_1, s_2, s_3, s_4\}$

$\texttt{SatSet}(TS,\ \varphi) = \{s_0, s_1, s_2, s_3\}$

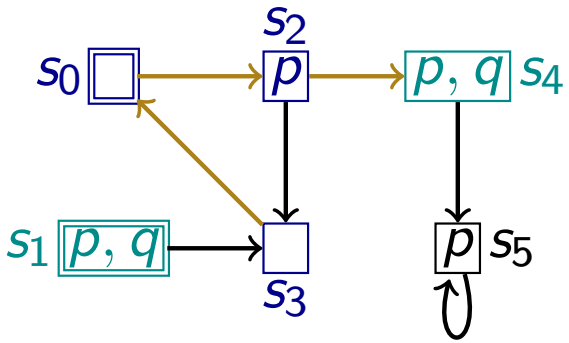
$S_0 = \{s_0, s_1\} \subseteq \{s_0, s_1, s_2, s_3\} = \texttt{SatSet}(TS,\ \varphi)$

$TS \models \varphi$

Решение задачи MC-CTL

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

TS:



$\texttt{SatSet}(TS,\ p) = \{s_1, s_2, s_4, s_5\}$

$\texttt{SatSet}(TS,\ \neg p) = \{s_0, s_3\}$

$\texttt{SatSet}(TS,\ \mathbf{EX}\neg p) = \{s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ \neg p \vee \mathbf{EX}\neg p) = \{s_0, s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ q) = \{s_1, s_4\}$

$\texttt{SatSet}(TS,\ (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}q) = \{s_0, s_1, s_2, s_3, s_4\}$

$\texttt{SatSet}(TS,\ \varphi) = \{s_0, s_1, s_2, s_3\}$

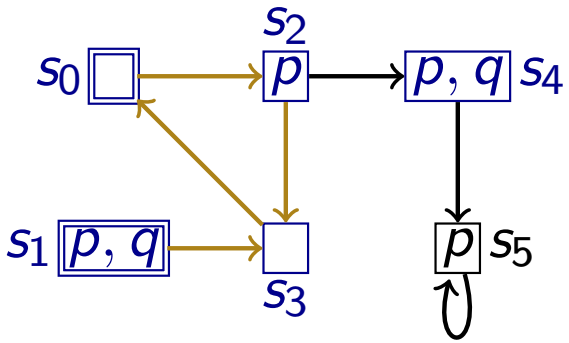
$S_0 = \{s_0, s_1\} \subseteq \{s_0, s_1, s_2, s_3\} = \texttt{SatSet}(TS,\ \varphi)$

$TS \models \varphi$

Решение задачи MC-CTL

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

TS:



$\texttt{SatSet}(TS,\ p) = \{s_1, s_2, s_4, s_5\}$

$\texttt{SatSet}(TS,\ \neg p) = \{s_0, s_3\}$

$\texttt{SatSet}(TS,\ \mathbf{EX}\neg p) = \{s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ \neg p \vee \mathbf{EX}\neg p) = \{s_0, s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ q) = \{s_1, s_4\}$

$\texttt{SatSet}(TS,\ (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}q) = \{s_0, s_1, s_2, s_3, s_4\}$

$\texttt{SatSet}(TS,\ \varphi) = \{s_0, s_1, s_2, s_3\}$

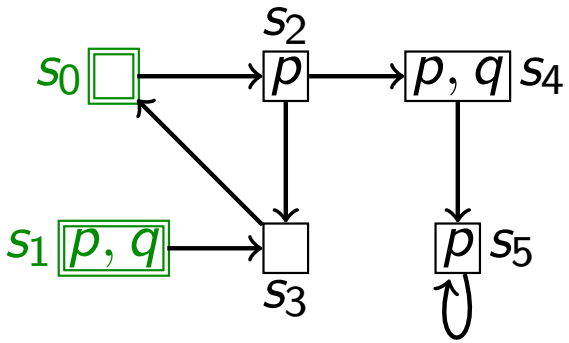
$S_0 = \{s_0, s_1\} \subseteq \{s_0, s_1, s_2, s_3\} = \texttt{SatSet}(TS,\ \varphi)$

$TS \models \varphi$

Решение задачи MC-CTL

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

TS:



$\texttt{SatSet}(TS,\ p) = \{s_1, s_2, s_4, s_5\}$

$\texttt{SatSet}(TS,\ \neg p) = \{s_0, s_3\}$

$\texttt{SatSet}(TS,\ \mathbf{EX}\neg p) = \{s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ \neg p \vee \mathbf{EX}\neg p) = \{s_0, s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ q) = \{s_1, s_4\}$

$\texttt{SatSet}(TS,\ (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}q) = \{s_0, s_1, s_2, s_3, s_4\}$

$\texttt{SatSet}(TS,\ \varphi) = \{s_0, s_1, s_2, s_3\}$

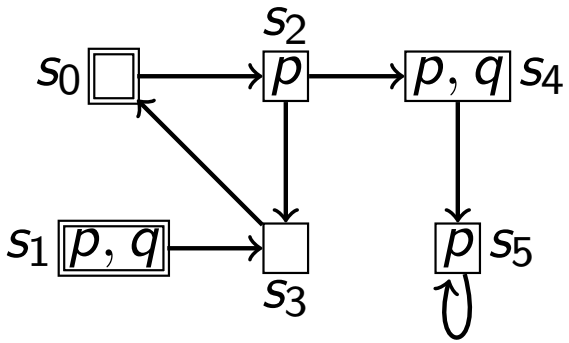
$S_0 = \{s_0, s_1\} \subseteq \{s_0, s_1, s_2, s_3\} = \texttt{SatSet}(TS,\ \varphi)$

$TS \models \varphi$

Решение задачи MC-CTL

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

TS:



$\texttt{SatSet}(TS,\ p) = \{s_1, s_2, s_4, s_5\}$

$\texttt{SatSet}(TS,\ \neg p) = \{s_0, s_3\}$

$\texttt{SatSet}(TS,\ \mathbf{EX}\neg p) = \{s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ \neg p \vee \mathbf{EX}\neg p) = \{s_0, s_1, s_2, s_3\}$

$\texttt{SatSet}(TS,\ q) = \{s_1, s_4\}$

$\texttt{SatSet}(TS,\ (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}q) = \{s_0, s_1, s_2, s_3, s_4\}$

$\texttt{SatSet}(TS,\ \varphi) = \{s_0, s_1, s_2, s_3\}$

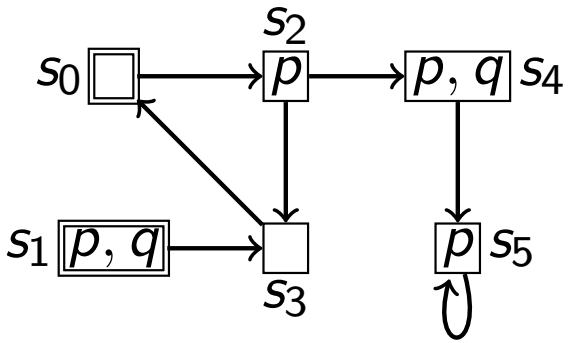
$S_0 = \{s_0, s_1\} \subseteq \{s_0, s_1, s_2, s_3\} = \texttt{SatSet}(TS,\ \varphi)$

$TS \models \varphi$

Решение задачи MC-CTL

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

TS:



$$\texttt{SatSet}(TS,\ p) = \{s_1, s_2, s_4, s_5\}$$
$$\texttt{SatSet}(TS,\ \neg p) = \{s_0, s_3\}$$
$$\texttt{SatSet}(TS,\ \mathbf{EX}\neg p) = \{s_1, s_2, s_3\}$$
$$\texttt{SatSet}(TS,\ \neg p \vee \mathbf{EX}\neg p) = \{s_0, s_1, s_2, s_3\}$$
$$\texttt{SatSet}(TS,\ q) = \{s_1, s_4\}$$
$$\texttt{SatSet}(TS,\ (\neg p \vee \mathbf{EX}\neg p)\mathbf{EU}q) = \{s_0, s_1, s_2, s_3, s_4\}$$
$$\texttt{SatSet}(TS,\ \varphi) = \{s_0, s_1, s_2, s_3\}$$
$$S_0 = \{s_0, s_1\} \subseteq \{s_0, s_1, s_2, s_3\} = \texttt{SatSet}(TS,\ \varphi)$$
$$TS \models \varphi$$

Для самостоятельного размышления:

Всегда ли процедура Sat работает правильно, и какова её сложность?