Верификация моделей программ

ЛЕКТОР:
Владимир Анатольевич Захаров

zakh@cs.msu.su

Лекция 6.

Символьный алгоритм верификации моделей для CTL

1. Предпосылки для символьных алгоритмов верификации
2. Представления неподвижной точки
3. Неподвижные точки и темпоральные операторы
4. Символьная верификация моделей для CTL

Предпосылки для символьной верификации

1. Булевы функции представимы ROBDD;

Предпосылки для символьной верификации

1. Булевы функции представимы ROBDD;
2. Для ROBDD имеются эффективные процедуры
   - проверки равенства булевых функций $f(\bar{x}) = g(\bar{x})$,
   - подстановки констант $f(\bar{x})\{x_i/c\}$,
   - вычисления булевых операций $f(\bar{x}) * g(\bar{x})$;

Предпосылки для символьной верификации

1. Булевы функции представимы ROBDD;
2. Для ROBDD имеются эффективные процедуры
   - проверки равенства булевых функций $f(\bar{x}) = g(\bar{x})$,
   - подстановки констант $f(\bar{x})\{x_i/c\}$,
   - вычисления булевых операций $f(\bar{x}) * g(\bar{x})$;
3. Множество состояний S модели Крипке $M = (S, S_0, R, L)$ кодируется двоичными наборами $(\sigma_1, \sigma_2, \ldots, \sigma_n)$;

Предпосылки для символьной верификации

1. Булевы функции представимы ROBDD;
2. Для ROBDD имеются эффективные процедуры
   - проверки равенства булевых функций $f(\bar{x}) = g(\bar{x})$,
   - подстановки констант $f(\bar{x})\{x_i/c\}$,
   - вычисления булевых операций $f(\bar{x}) * g(\bar{x})$;
3. Множество состояний S модели Крипке $M = (S, S_0, R, L)$ кодируется двоичными наборами $(\sigma_1, \sigma_2, \ldots, \sigma_n)$;
4. Отношение переходов R модели Крипке $M = (S, S_0, R, L)$ рассматривается как булева функция $R(\bar{x}, \bar{x}')$, которая зависит от переменных, представляющих состояния модели;

Предпосылки для символьной верификации

1. Булевы функции представимы ROBDD;
2. Для ROBDD имеются эффективные процедуры
   - проверки равенства булевых функций $f(\bar{x}) = g(\bar{x})$,
   - подстановки констант $f(\bar{x})\{x_i/c\}$,
   - вычисления булевых операций $f(\bar{x}) * g(\bar{x})$;
3. Множество состояний S модели Крипке $M = (S, S_0, R, L)$ кодируется двоичными наборами $(\sigma_1, \sigma_2, \ldots, \sigma_n)$;
4. Отношение переходов R модели Крипке $M = (S, S_0, R, L)$ рассматривается как булева функция $R(\bar{x}, \bar{x}')$, которая зависит от переменных, представляющих состояния модели;
5. Функция разметки L модели Крипке $M = (S, S_0, R, L)$ рассматривается как набор булевых функций $L_p(\bar{x}),\ p \in AP$, выделяющих те состояния, на которых выполнимы атомарные формулы.

Предпосылки для символьной верификации

Операции над множествами и отношениями моделируются операциями над булевыми функциями:

- пересечение: $A \cap B = C \implies f_C = f_A \wedge f_B$,
- объединение: $A \cup B = C \implies f_C = f_A \vee f_B$,
- дополнение: $S \setminus A = C \implies f_C = \neg f_A$,
- образ: $R(A) = C \implies f_C = \exists \bar{x}(f_A(\bar{x}) \wedge R(\bar{x}, \bar{y}))$,
- прообраз: $R^{-1}(A) = C \implies f_C = \exists \bar{y}(f_A(\bar{y}) \wedge R(\bar{x}, \bar{y}))$.

Здесь $\exists y \ h(x, y) = h(x, 0) \vee h(x, 1)$.

Предпосылки для символьной верификации

Алгоритмы анализа моделей, использующие их представления в виде OBDD, называются символьным или неявными , поскольку он оперирует не с самими множествами, элементы которых явно перечислены, а с символьными описаниями (представлениями) множеств.

Когда модель имеет большой размер ($\geq 2^{32}$), приходится оперировать с целыми множествами, а не с отдельными состояниями и переходами. Табличные алгоритмы model checking для этой цели непригодны. Нужны итеративные алгоритмы, работающие с описаниями множеств.

Для этого воспользуемся определением операторов темпоральной логики в терминах неподвижной точки .

Покажем, как описать множество состояний, удовлетворяющих некоторой CTL-формуле, при помощи наименьшей или наибольшей неподвижной точки подходящей функции.

Представления неподвижной точки

Пусть задана произвольная модель Крипке $M = (S, S_0, R, L)$ с конечным множеством состояний.

Совокупность $\wp(S)$ всех подмножеств множества S образует решетку по отношению включения $\subseteq$:

- точная верхняя грань $sup(S_1, S_2) = S_1 \cup S_2$,
- точная нижняя грань $inf(S_1, S_2) = S_1 \cap S_2$,

Всякий элемент S' решетки $(\wp(S), \subseteq)$ может рассматриваться как предикат (отношение) на S, принимающий значение $true$ в точности на всех состояниях из S'.

Наименьшим элементом решетки является пустое множество, которое будет обозначаться $False$, а наибольшим элементом решетки служит само множество S, которое мы будем обозначать $True$.

Функцию, отображающую $\wp(S)$ в $\wp(S)$, будем называть преобразователем предикатов.

Представления неподвижной точки

Пусть $\tau : \wp(S) \to \wp(S)$ — один из таких преобразователей предикатов. Тогда

1) Функция τ считается монотонной, если из включения $P \subseteq Q$ следует $\tau(P) \subseteq \tau(Q)$;
2) Функция τ считается $\cup$-непрерывной, если для всякой монотонно неубывающей последовательности предикатов $P_1 \subseteq P_2 \subseteq \dots$ верно равенство $\tau(\bigcup\limits_{i=1}^{\infty} P_i) = \bigcup\limits_{i=1}^{\infty} \tau(P_i)$;
3) Функция τ считается $\cap$-непрерывной, если для всякой монотонно невозрастающей последовательности предикатов $P_1 \supseteq P_2 \supseteq \dots$ верно $\tau(\bigcap\limits_{i=1}^{\infty} P_i) = \bigcap\limits_{i=1}^{\infty} \tau(P_i)$

Предикат P называется неподвижной точкой преобразователя τ, если имеет место равенство $\tau(P) = P$.

Представления неподвижной точки

У монотонного преобразователя τ на $\wp(S)$ всегда есть
наименьшая неподвижная точка $\mu Z \,.\, \tau(Z)$ и
наибольшая неподвижная точка $\nu Z \,.\, \tau(Z)$

Наименьшая неподвижная точка имеет вид
$\mu Z \,.\, \tau(Z) = \cap\{Z \mid \tau(Z) \subseteq Z\}$
для монотонного τ, и при этом
$$\mu Z \,.\, \tau(Z) = \bigcup_{i=1}^{\infty} \tau^i(\mathit{False}),$$
если преобразователь τ является $\cup$-непрерывным.

Аналогично, наибольшая неподвижная точка имеет вид
$\nu Z \,.\, \tau(Z) = \cup\{Z \mid \tau(Z) \supseteq Z\}$
для монотонного τ, и при этом
$$\mu Z \,.\, \tau(Z) = \bigcap_{i=1}^{\infty} \tau^i(\mathit{True}),$$
если преобразователь τ является $\cap$-непрерывным.

Представления неподвижной точки

У монотонного преобразователя τ на $\wp(S)$ всегда есть наименьшая неподвижная точка $\mu Z \,.\, \tau(Z)$ и наибольшая неподвижная точка $\nu Z \,.\, \tau(Z)$

Наименьшая неподвижная точка имеет вид

$$\mu Z \,.\, \tau(Z) = \cap\{Z \mid \tau(Z) \subseteq Z\}$$

для монотонного τ, и при этом

$$\mu Z \,.\, \tau(Z) = \bigcup_{i=1}^{\infty} \tau^i(\mathit{False}),$$

если преобразователь τ является $\cup$-непрерывным.

Аналогично, наибольшая неподвижная точка имеет вид

$$\nu Z \,.\, \tau(Z) = \cup\{Z \mid \tau(Z) \supseteq Z\}$$

для монотонного τ, и при этом

$$\mu Z \,.\, \tau(Z) = \bigcap_{i=1}^{\infty} \tau^i(\mathit{True}),$$

если преобразователь τ является $\cap$-непрерывным.

Попробуйте доказать самостоятельно. Это несложный математический этюд из теории множеств.

Представления неподвижной точки

Следующие леммы весьма полезны, когда приходится иметь дело с преобразователями предикатов, определенными на конечных моделях Крипке.

Лемма 1. Если S — конечное множество, а τ — монотонный преобразователь, то τ также $\bigcup$-непрерывен и $\bigcap$-непрерывен.

Док-во: Рассмотрим последовательность $P_1 \subseteq P_2 \subseteq \ldots$ подмножеств множества S.

Представления неподвижной точки

Следующие леммы весьма полезны, когда приходится иметь дело с преобразователями предикатов, определенными на конечных моделях Крипке.

Лемма 1. Если S — конечное множество, а τ — монотонный преобразователь, то τ также $\bigcup$-непрерывен и $\bigcap$-непрерывен.

Док-во: Рассмотрим последовательность $P_1 \subseteq P_2 \subseteq \ldots$ подмножеств множества S. Так как S — конечное множество, существует такое j_0, что $P_j = P_{j_0}$ для любого $j \geqslant j_0$. При этом $P_j \subseteq P_{j_0}$ для каждого $j < j_0$.

Представления неподвижной точки

Следующие леммы весьма полезны, когда приходится иметь дело с преобразователями предикатов, определенными на конечных моделях Крипке.

Лемма 1. Если S — конечное множество, а τ — монотонный преобразователь, то τ также $\cup$-непрерывен и $\cap$-непрерывен.

Док-во: Рассмотрим последовательность $P_1 \subseteq P_2 \subseteq \ldots$ подмножеств множества S. Так как S — конечное множество, существует такое j_0, что $P_j = P_{j_0}$ для любого $j \geqslant j_0$. При этом $P_j \subseteq P_{j_0}$ для каждого $j < j_0$. Таким образом, $\cup_i P_i = P_{j_0}$, и вследствие этого мы получаем $\tau(\cup_i P_i) = \tau(P_{j_0})$.

Представления неподвижной точки

Следующие леммы весьма полезны, когда приходится иметь дело с преобразователями предикатов, определенными на конечных моделях Крипке.

Лемма 1. Если S — конечное множество, а τ — монотонный преобразователь, то τ также $\cup$-непрерывен и $\cap$-непрерывен.

Док-во: Рассмотрим последовательность $P_1 \subseteq P_2 \subseteq \ldots$ подмножеств множества S. Так как S — конечное множество, существует такое j_0, что $P_j = P_{j_0}$ для любого $j \geqslant j_0$. При этом $P_j \subseteq P_{j_0}$ для каждого $j < j_0$. Таким образом, $\cup_i P_i = P_{j_0}$, и вследствие этого мы получаем $\tau(\cup_i P_i) = \tau(P_{j_0})$. С другой стороны, в силу монотонности τ мы имеем $\tau(P_1) \subseteq \tau(P_2) \subseteq \ldots$

Представления неподвижной точки

Следующие леммы весьма полезны, когда приходится иметь дело с преобразователями предикатов, определенными на конечных моделях Крипке.

Лемма 1. Если S — конечное множество, а τ — монотонный преобразователь, то τ также $\cup$-непрерывен и $\cap$-непрерывен.

Док-во: Рассмотрим последовательность $P_1 \subseteq P_2 \subseteq \ldots$ подмножеств множества S. Так как S — конечное множество, существует такое j_0, что $P_j = P_{j_0}$ для любого $j \geqslant j_0$. При этом $P_j \subseteq P_{j_0}$ для каждого $j < j_0$. Таким образом, $\cup_i P_i = P_{j_0}$, и вследствие этого мы получаем $\tau(\cup_i P_i) = \tau(P_{j_0})$. С другой стороны, в силу монотонности τ мы имеем $\tau(P_1) \subseteq \tau(P_2) \subseteq \ldots$ Поэтому $\tau(P_j) \subseteq \tau(P_{j_0})$ для каждого $j < j_0$, и $\tau(P_j) = \tau(P_{j_0})$ для каждого $j \geqslant j_0$.

Представления неподвижной точки

Следующие леммы весьма полезны, когда приходится иметь дело с преобразователями предикатов, определенными на конечных моделях Крипке.

Лемма 1. Если S — конечное множество, а τ — монотонный преобразователь, то τ также $\cup$-непрерывен и $\cap$-непрерывен.

Док-во: Рассмотрим последовательность $P_1 \subseteq P_2 \subseteq \ldots$ подмножеств множества S. Так как S — конечное множество, существует такое j_0, что $P_j = P_{j_0}$ для любого $j \geqslant j_0$. При этом $P_j \subseteq P_{j_0}$ для каждого $j < j_0$. Таким образом, $\cup_i P_i = P_{j_0}$, и вследствие этого мы получаем $\tau(\cup_i P_i) = \tau(P_{j_0})$. С другой стороны, в силу монотонности τ мы имеем $\tau(P_1) \subseteq \tau(P_2) \subseteq \ldots$ Поэтому $\tau(P_j) \subseteq \tau(P_{j_0})$ для каждого $j < j_0$, и $\tau(P_j) = \tau(P_{j_0})$ для каждого $j \geqslant j_0$. В результате получаем соотношение $\cup_i \tau(P_i) = \tau(P_{j_0})$, а это означает, что преобразователь τ является $\cup$-непрерывным.

Представления неподвижной точки

Следующие леммы весьма полезны, когда приходится иметь дело с преобразователями предикатов, определенными на конечных моделях Крипке.

Лемма 1. Если S — конечное множество, а τ — монотонный преобразователь, то τ также $\cup$-непрерывен и $\cap$-непрерывен.

Док-во: Рассмотрим последовательность $P_1 \subseteq P_2 \subseteq \ldots$ подмножеств множества S. Так как S — конечное множество, существует такое j_0, что $P_j = P_{j_0}$ для любого $j \geqslant j_0$. При этом $P_j \subseteq P_{j_0}$ для каждого $j < j_0$. Таким образом, $\cup_i P_i = P_{j_0}$, и вследствие этого мы получаем $\tau(\cup_i P_i) = \tau(P_{j_0})$. С другой стороны, в силу монотонности τ мы имеем $\tau(P_1) \subseteq \tau(P_2) \subseteq \ldots$ Поэтому $\tau(P_j) \subseteq \tau(P_{j_0})$ для каждого $j < j_0$, и $\tau(P_j) = \tau(P_{j_0})$ для каждого $j \geqslant j_0$. В результате получаем соотношение $\cup_i \tau(P_i) = \tau(P_{j_0})$, а это означает, что преобразователь τ является $\cup$-непрерывным. Обоснование $\cap$-непрерывности преобразователя τ проводится аналогично.

Представления неподвижной точки

Запись $\tau^i(Z)$ будет обозначать i-кратное применение τ к Z.
Более строго $\tau^i(Z)$ определяется рекурсивно соотношениями $\tau^0(Z) = Z$ и $\tau^{i+1}(Z) = \tau(\tau^i(Z))$.

Лемма 2. Если τ — монотонный преобразователь, то для любого натурального числа i имеют место включения

$$\tau^i(\mathit{False}) \subseteq \tau^{i+1}(\mathit{False}) \text{ и } \tau^i(\mathit{True}) \supseteq \tau^{i+1}(\mathit{True}).$$

Представления неподвижной точки

Запись $\tau^i(Z)$ будет обозначать i-кратное применение τ к Z.
Более строго $\tau^i(Z)$ определяется рекурсивно соотношениями $\tau^0(Z) = Z$ и $\tau^{i+1}(Z) = \tau(\tau^i(Z))$.

Лемма 2. Если τ — монотонный преобразователь, то для любого натурального числа i имеют место включения

$$\tau^i(False) \subseteq \tau^{i+1}(False) \text{ и } \tau^i(True) \supseteq \tau^{i+1}(True).$$

Лемма 3. Если τ — монотонный преобразователь, а S — конечное множество, то существуют такие натуральные числа i_0 и j_0, что для любого $i \geqslant i_0$ верно равенство $\tau^i(False) = \tau^{i_0}(False)$, и для любого $j \geqslant j_0$ верно равенство $\tau^j(True) = \tau^{j_0}(True)$.

Представления неподвижной точки

Запись $\tau^i(Z)$ будет обозначать i-кратное применение τ к Z.
Более строго $\tau^i(Z)$ определяется рекурсивно соотношениями $\tau^0(Z) = Z$ и $\tau^{i+1}(Z) = \tau(\tau^i(Z))$.

Лемма 2. Если τ — монотонный преобразователь, то для любого натурального числа i имеют место включения

$$\tau^i(False) \subseteq \tau^{i+1}(False) \text{ и } \tau^i(True) \supseteq \tau^{i+1}(True).$$

Лемма 3. Если τ — монотонный преобразователь, а S — конечное множество, то существуют такие натуральные числа i_0 и j_0, что для любого $i \geqslant i_0$ верно равенство $\tau^i(False) = \tau^{i_0}(False)$, и для любого $j \geqslant j_0$ верно равенство $\tau^j(True) = \tau^{j_0}(True)$.

Лемма 4. Если τ — монотонный преобразователь, а S — конечное множество, то существуют такие натуральные числа i_0 и j_0, что

$$\mu Z\ .\ \tau(Z) = \tau^{i_0}(False) \text{ и } \nu Z\ .\ \tau(Z) = \tau^{j_0}(True)$$

Представления неподвижной точки

На основании приведенных лемм для вычисления наименьшей неподвижной точки монотонного преобразователя τ можно воспользоваться программой

function $Lfp(Tau: PredicateTransformer): Predicate$
$Q := False;\ Q' := Tau(Q);$
while $(Q \neq Q')$ **do**
$Q := Q';\ Q' := Tau(Q');$
end while;
return(Q)
end function

Представления неподвижной точки

Инвариант цикла **while** в теле процедуры задается отношением

$$(Q' = \tau(Q)) \wedge (Q' \subseteq \mu Z \,.\, \tau(Z)) \,.$$

Нетрудно заметить, что в начале i-й итерации цикла выполняется $Q \subseteq \tau^{i-1}(\mathit{False})$ и $Q' \subseteq \tau^{i}(\mathit{False})$.

Представления неподвижной точки

Инвариант цикла **while** в теле процедуры задается отношением

$$(Q' = \tau(Q)) \wedge (Q' \subseteq \mu Z \,.\, \tau(Z)) .$$

Нетрудно заметить, что в начале i-й итерации цикла выполняется $Q \subseteq \tau^{i-1}(\mathit{False})$ и $Q' \subseteq \tau^{i}(\mathit{False})$.

Из леммы 4 вытекает $\mathit{False} \subseteq \tau(\mathit{False}) \subseteq \tau^2(\mathit{False}) \subseteq \ldots$

Поэтому максимальное число итераций оператора цикла ограничено количеством элементов в S. При выходе из цикла, $Q = \tau(Q)$ и $Q \subseteq \mu Z \,.\, \tau(Z)$. Так как Q является неподвижной точкой, $\mu Z \,.\, \tau(Z) \subseteq Q$, и поэтому $Q = \mu Z \,.\, \tau(Z)$. Тем самым показано, что значение, которое возвращает процедура, — это действительно наименьшая неподвижная точка.

Представления неподвижной точки

Наибольшая неподвижная точка вычисляется подобным же образом при помощи другой программы. Применяя по сути дела ту же самую аргументацию, можно показать, что и эта процедура всегда завершает свои вычисления и возвращает в качестве значения $\nu Z \,.\, \tau(Z)$.

function $Gfp(Tau : PredicateTransformer) : Predicate$
    $Q := True;\ Q' := Tau(Q);$
    while $(Q \neq Q')$ **do**
        $Q := Q';\ Q' := Tau(Q');$
    end while;
    return(Q)
end function

Неподвижные точки и темпоральные операторы

Если сопоставить каждой формуле f предикат $\{s \mid M, s \models f\}$ на $\wp(S)$, то каждый темпоральный оператор CTL можно описать в терминах наименьшей или наибольшей неподвижной точки подходящего преобразователя предикатов. Мы обоснуем описания в терминах неподвижных точек только для **EG** и **EU**.

- $\mathbf{EG}\, f_1 = \nu Z \,.\, f_1 \wedge \mathbf{EX}\, Z$,
- $\mathbf{E}[f_1\, \mathbf{U}\, f_2] = \mu Z \,.\, f_2 \vee (f_1 \wedge \mathbf{EX}\, Z)$,

Интуитивно понятно, что наименьшие неподвижные точки соответствуют свойствам, которые должны выполняться когда-нибудь, а наибольшие неподвижные точки соответствуют свойствам, которые должны выполняться всегда. Поэтому $\mathbf{AF}\, f_1$ характеризуется при помощи наименьшей неподвижной точки, а $\mathbf{EG}\, f_1$ имеет описание в терминах наибольшей неподвижной точки.

Неподвижные точки и темпоральные операторы

Лемма 5. Преобразователь $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$ монотонный.

Неподвижные точки и темпоральные операторы

Лемма 5. Преобразователь $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$ монотонный.

Док-во. Предположим, что $P_1 \subseteq P_2$.

Неподвижные точки и темпоральные операторы

Лемма 5. Преобразователь $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$ монотонный.

Док-во. Предположим, что $P_1 \subseteq P_2$. Чтобы проверить включение $\tau(P_1) \subseteq \tau(P_2)$, возьмем произвольное состояние $s \in \tau(P_1)$. Тогда $s \models f_1$ и существует такое состояние s', что $(s, s') \in R$ и $s' \in P_1$.

Неподвижные точки и темпоральные операторы

Лемма 5. Преобразователь $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$ монотонный.

Док-во. Предположим, что $P_1 \subseteq P_2$. Чтобы проверить включение $\tau(P_1) \subseteq \tau(P_2)$, возьмем произвольное состояние $s \in \tau(P_1)$. Тогда $s \models f_1$ и существует такое состояние s', что $(s, s') \in R$ и $s' \in P_1$. Но ввиду того что $P_1 \subseteq P_2$, верно также и включение $s' \in P_2$.

Неподвижные точки и темпоральные операторы

Лемма 5. Преобразователь $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$ монотонный.

Док-во. Предположим, что $P_1 \subseteq P_2$. Чтобы проверить включение $\tau(P_1) \subseteq \tau(P_2)$, возьмем произвольное состояние $s \in \tau(P_1)$. Тогда $s \models f_1$ и существует такое состояние s', что $(s, s') \in R$ и $s' \in P_1$. Но ввиду того что $P_1 \subseteq P_2$, верно также и включение $s' \in P_2$. Значит, $s \in \tau(P_2)$. $\square$

Неподвижные точки и темпоральные операторы

Лемма 6. Пусть $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$. Обозначим через $\tau^{i_0}(True)$ предел последовательности $True \supseteq \tau(True) \supseteq \ldots$ Тогда для любого состояния s из $\tau^{i_0}(True)$ мы имеем $s \models f_1$ и, кроме того, найдется такое состояние s', что $(s, s') \in R$ и $s' \in \tau^{i_0}(True)$.

Неподвижные точки и темпоральные операторы

Лемма 6. Пусть $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$. Обозначим через $\tau^{i_0}(True)$ предел последовательности $True \supseteq \tau(True) \supseteq \ldots$ Тогда для любого состояния s из $\tau^{i_0}(True)$ мы имеем $s \models f_1$ и, кроме того, найдется такое состояние s', что $(s, s') \in R$ и $s' \in \tau^{i_0}(True)$.

Док-во. Предположим, что $s \in \tau^{i_0}(True)$. Поскольку предикат $\tau^{i_0}(True)$ является неподвижной точкой τ, имеет место равенство $\tau^{i_0}(True) = \tau(\tau^{i_0}(True))$.

Неподвижные точки и темпоральные операторы

Лемма 6. Пусть $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$. Обозначим через $\tau^{i_0}(True)$ предел последовательности $True \supseteq \tau(True) \supseteq \ldots$ Тогда для любого состояния s из $\tau^{i_0}(True)$ мы имеем $s \models f_1$ и, кроме того, найдется такое состояние s', что $(s, s') \in R$ и $s' \in \tau^{i_0}(True)$.

Док-во. Предположим, что $s \in \tau^{i_0}(True)$. Поскольку предикат $\tau^{i_0}(True)$ является неподвижной точкой τ, имеет место равенство $\tau^{i_0}(True) = \tau(\tau^{i_0}(True))$. Таким образом, $s \in \tau(\tau^{i_0}(True))$.

Неподвижные точки и темпоральные операторы

Лемма 6. Пусть $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$. Обозначим через $\tau^{i_0}(True)$ предел последовательности $True \supseteq \tau(True) \supseteq \ldots$ Тогда для любого состояния s из $\tau^{i_0}(True)$ мы имеем $s \models f_1$ и, кроме того, найдется такое состояние s' , что $(s, s') \in R$ и $s' \in \tau^{i_0}(True)$.

Док-во. Предположим, что $s \in \tau^{i_0}(True)$. Поскольку предикат $\tau^{i_0}(True)$ является неподвижной точкой τ , имеет место равенство $\tau^{i_0}(True) = \tau(\tau^{i_0}(True))$. Таким образом, $s \in \tau(\tau^{i_0}(True))$. По определению преобразователя τ получаем, что $s \models f_1$ и, кроме того, имеется такое состояние s' , что $(s, s') \in R$ и $s' \in \tau^{i_0}(True)$. □

Неподвижные точки и темпоральные операторы

Лемма 7. Предикат $\mathbf{EG}\, f_1$ является неподвижной точкой преобразователя $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$.

Неподвижные точки и темпоральные операторы

Лемма 7. Предикат $\mathbf{EG}\, f_1$ является неподвижной точкой преобразователя $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$.

Док-во. Предположим, что $s_0 \models \mathbf{EG}\, f_1$.

Неподвижные точки и темпоральные операторы

Лемма 7. Предикат $\mathbf{EG}\, f_1$ является неподвижной точкой преобразователя $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$.

Док-во. Предположим, что $s_0 \models \mathbf{EG}\, f_1$. Тогда по определению отношения выполнимости $\models$ существует такой путь $s_0, s_1, \ldots$ в M, что для всякого k имеет место соотношение $s_k \models f_1$.

Неподвижные точки и темпоральные операторы

Лемма 7. Предикат $\mathbf{EG}\, f_1$ является неподвижной точкой преобразователя $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$.

Док-во. Предположим, что $s_0 \models \mathbf{EG}\, f_1$. Тогда по определению отношения выполнимости $\models$ существует такой путь $s_0, s_1, \ldots$ в M, что для всякого k имеет место соотношение $s_k \models f_1$. Отсюда следует, что $s_0 \models f_1$ и $s_1 \models \mathbf{EG}\, f_1$. Иными словами, $s_0 \models f_1$ и $s_0 \models \mathbf{EX}\, \mathbf{EG}\, f_1$.

Неподвижные точки и темпоральные операторы

Лемма 7. Предикат $\mathbf{EG}\, f_1$ является неподвижной точкой преобразователя $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$.

Док-во. Предположим, что $s_0 \models \mathbf{EG}\, f_1$. Тогда по определению отношения выполнимости $\models$ существует такой путь $s_0, s_1, \ldots$ в M, что для всякого k имеет место соотношение $s_k \models f_1$. Отсюда следует, что $s_0 \models f_1$ и $s_1 \models \mathbf{EG}\, f_1$. Иными словами, $s_0 \models f_1$ и $s_0 \models \mathbf{EX}\, \mathbf{EG}\, f_1$. Поэтому $\mathbf{EG}\, f_1 \subseteq f_1 \wedge \mathbf{EX}\, \mathbf{EG}\, f_1$.

Неподвижные точки и темпоральные операторы

Лемма 7. Предикат $\mathbf{EG}\, f_1$ является неподвижной точкой преобразователя $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$.

Док-во. Предположим, что $s_0 \models \mathbf{EG}\, f_1$. Тогда по определению отношения выполнимости $\models$ существует такой путь $s_0, s_1, \ldots$ в M, что для всякого k имеет место соотношение $s_k \models f_1$. Отсюда следует, что $s_0 \models f_1$ и $s_1 \models \mathbf{EG}\, f_1$. Иными словами, $s_0 \models f_1$ и $s_0 \models \mathbf{EX}\,\mathbf{EG}\, f_1$. Поэтому $\mathbf{EG}\, f_1 \subseteq f_1 \wedge \mathbf{EX}\,\mathbf{EG}\, f_1$.

И наоборот, если $s_0 \models f_1 \wedge \mathbf{EX}\,\mathbf{EG}\, f_1$, то $s_0 \models \mathbf{EG}\, f_1$. В результате получаем $\mathbf{EG}\, f_1 = f_1 \wedge \mathbf{EX}\,\mathbf{EG}\, f_1$. □

Неподвижные точки и темпоральные операторы

Лемма 8. Предикат $\mathbf{EG}\, f_1$ является наибольшей неподвижной точкой преобразователя $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$.

Неподвижные точки и темпоральные операторы

Лемма 8. Предикат $\mathbf{EG}\, f_1$ является наибольшей неподвижной точкой преобразователя $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$.

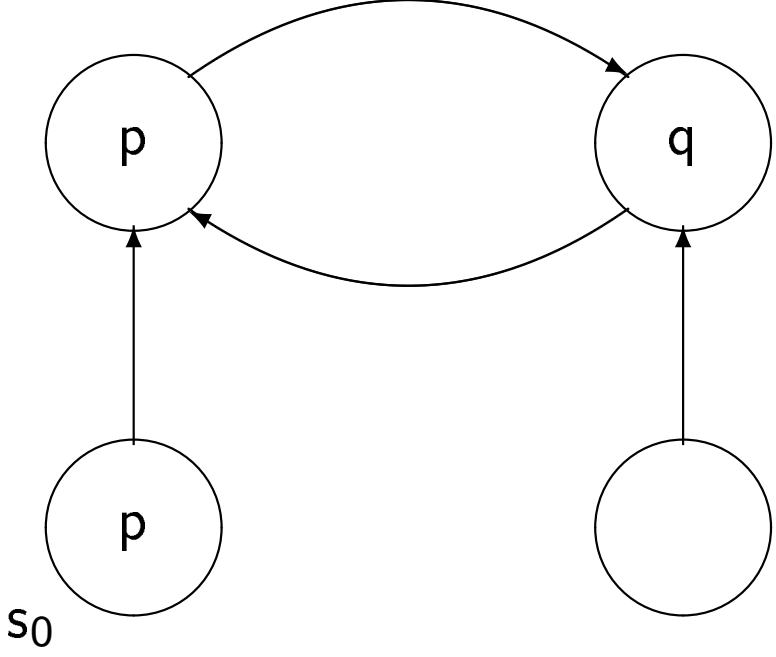
Лемма 9. Предикат $\mathbf{E}[f_1\ \mathbf{U}\ f_2]$ является наименьшей неподвижной точкой преобразователя $\tau(Z) = f_2 \vee (f_1 \wedge \mathbf{EX}\, Z)$.

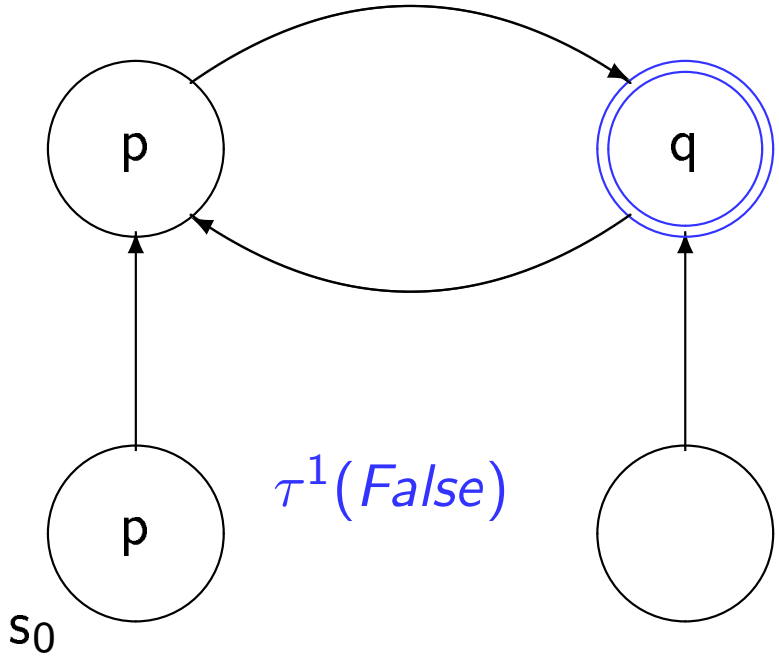
Неподвижные точки и темпоральные операторы

Лемма 8. Предикат $\mathbf{EG}\, f_1$ является наибольшей неподвижной точкой преобразователя $\tau(Z) = f_1 \wedge \mathbf{EX}\, Z$.

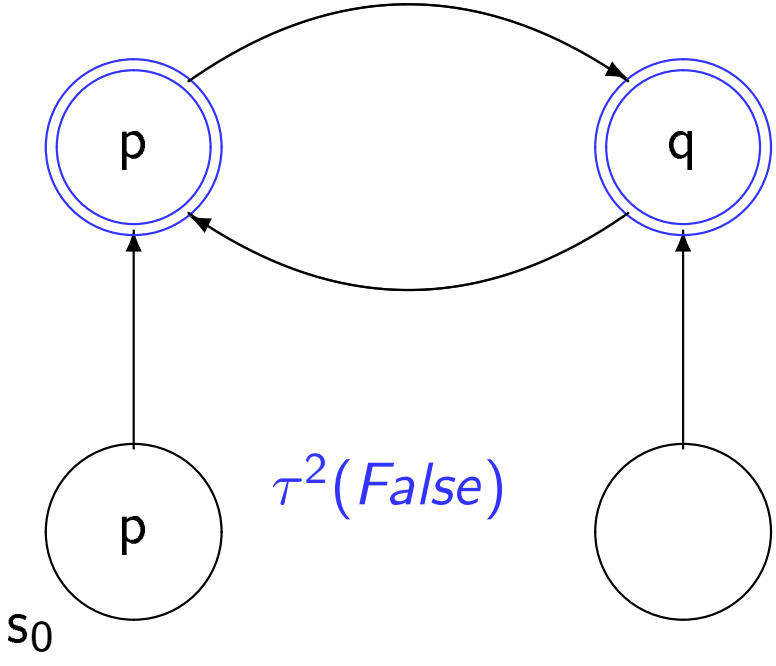
Лемма 9. Предикат $\mathbf{E}[f_1\ \mathbf{U}\ f_2]$ является наименьшей неподвижной точкой преобразователя $\tau(Z) = f_2 \vee (f_1 \wedge \mathbf{EX}\, Z)$.

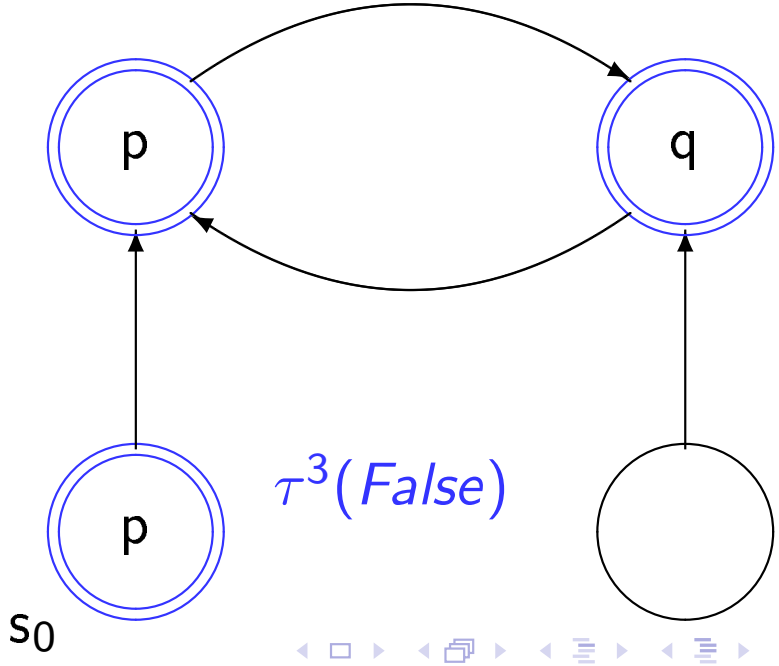
Док-во. **Самостоятельно.**





$$\mathbf{E}[p\,\mathbf{U}\,q] = \mu Z\ .\ q \vee (p \wedge \mathbf{EX}\,Z)$$





Символьная верификация моделей для CTL

Табличный агоритм верификации моделей для CTL с явным представлением модели имеет линейную сложность как по размеру модели, так и по длине формулы.

Однако размер модели параллельной системы, состоящей из многих процессов или компонентов, возрастает экспоненциально с увеличением числа процессов.

Символьные алгоритмы верификации моделей для CTL позволяют преодолет трудности **state explosion** (**«комбинаторного взрыва» числа состояний**). Чтобы сформулировать символьный алгоритм верификации моделей, лучше всего использовать сокращенную систему обозначаний для сложных операций на булевых формулах. Для этого пригодны квантифицированные булевы формулы (QBF).

Символьная верификация моделей для CTL

Пусть задано множество $V = \{v_0, \ldots, v_{n-1}\}$ пропозициональных переменных. Обозначим через $\mathrm{QBF}(V)$ наименьшее множество формул, удовлетворяющих следующим условиям:

- каждая переменная из V является формулой;
- если f и g — формулы, то формулами являются и выражения $\neg f$, $f \wedge g$, $f \vee g$;
- если f — формула и $v \in V$, то формулами являются и выражения $\exists v\ f$ и $\forall v\ f$.

Символьная верификация моделей для CTL

Оценкой значений истинности для множества $\mathrm{QBF}(V)$ называется функция $\sigma\colon V \to \{0,1\}$. Если $a \in \{0,1\}$, то запись $\sigma\langle v \leftarrow a\rangle$ обозначает истинностную оценку, которое определяется следующим соотношением:

$$\sigma\langle v \leftarrow a\rangle(w) = \begin{cases} a\text{, если } v = w\text{,} \\ \sigma(w)\text{, если } v \neq w\text{.} \end{cases}$$

Символьная верификация моделей для CTL

Оценкой значений истинности для множества $\mathrm{QBF}(V)$ называется функция $\sigma\colon V \to \{0,1\}$. Если $a \in \{0,1\}$, то запись $\sigma\langle v \leftarrow a\rangle$ обозначает истинностную оценку, которое определяется следующим соотношением:

$$\sigma\langle v \leftarrow a\rangle(w) = \begin{cases} a, \text{ если } v = w, \\ \sigma(w), \text{ если } v \neq w. \end{cases}$$

Запись $\sigma \models f$, где f — квантифицированная булева формула, а σ — истинностная оценка, обозначает, что формула f истинна при оценке σ. Отношение $\models$ определяется индуктивно очевидным образом:

- $\sigma \models v \iff \sigma(v) = 1$,
- $\sigma \models \neg f \iff \sigma \not\models f$,
- $\sigma \models f \vee g \iff \sigma \models f$ или $\sigma \models g$,
- $\sigma \models f \wedge g \iff \sigma \models f$ и $\sigma \models g$,
- $\sigma \models \exists v f \iff \sigma\langle v \to 0\rangle \models f$ или $\sigma\langle v \to 1\rangle \models f$,
- $\sigma \models \forall v f \iff \sigma\langle v \to 0\rangle \models f$ и $\sigma\langle v \to 1\rangle \models f$.

Символьная верификация моделей для CTL

Выразительные возможности QBF точно такие же, как и у обычных булевых формул, просто QBF более лаконичны.

Каждая QBF задает некоторое отношение на множестве V, и это отношение можно реализовать при помощи ROBDD, если воспользоваться композицией операции ограничения и операции *Apply*:

- $\exists x f = f|_{x \leftarrow 0} \vee f|_{x \leftarrow 1}$,
- $\forall x f = f|_{x \leftarrow 0} \wedge f|_{x \leftarrow 1}$.

Кванторы используются преимущественно в операциях реляционного произведения, которые можно представить в виде $\exists \overline{v}[f(\overline{w}, \overline{v}) \wedge g(\overline{v}, \overline{x})]$.

Символьная верификация моделей для CTL

Символьный алгоритм верификации моделей реализован в виде процедуры *Check*. Она получает в качестве аргумента CTL формулу φ, которую необходимо проверить, и OBDD, представляющей отношение переходов R анализируемой модели Крипке M. Процедура вычисляет OBDD, представляющую множество $S_\varphi = \{s \mid M, s \models \varphi\}$ всех тех состояний модели M, в которых выполняется φ.

Символьная верификация моделей для CTL

Символьный алгоритм верификации моделей реализован в виде процедуры $Check$. Она получает в качестве аргумента CTL формулу φ, которую необходимо проверить, и OBDD, представляющей отношение переходов R анализируемой модели Крипке M. Процедура вычисляет OBDD, представляющую множество $S_\varphi = \{s \mid M, s \models \varphi\}$ всех тех состояний модели M, в которых выполняется φ.

Процедура $Check$ проводит вычисления индуктивно, сообразно структуре CTL формул.

Если f — это атомарное высказывание a, то $Check(f, R)$ — это OBDD, представляющая множество состояний системы, в которых выполняется a.

Если $f = f_1 \vee f_2$ или $f = \neg f_1$, то $Check(f, R)$ получается применением функции $Apply$ к аргументам $Check(f_1, R)$ и $Check(f_2, R)$.

Символьная верификация моделей для CTL

Формулы вида $\mathbf{EX}\, f$, $\mathbf{E}[f\ \mathbf{U}\ g]$ и $\mathbf{EG}\, f$ обрабатываются при помощи отдельных процедур

$$Check(\mathbf{EX}\, f, R) = CheckEX(Check(f, R)),$$
$$Check(\mathbf{E}[f\ \mathbf{U}\ g], R) = CheckEU(Check(f, R), Check(g, R)),$$
$$Check(\mathbf{EG}\, f, R) = CheckEG(Check(f, R)).$$

Символьная верификация моделей для CTL

Формулы вида $\mathbf{EX}\, f$, $\mathbf{E}[f\ \mathbf{U}\ g]$ и $\mathbf{EG}\, f$ обрабатываются при помощи отдельных процедур

$$Check(\mathbf{EX}\, f, R) = CheckEX(Check(f, R)),$$
$$Check(\mathbf{E}[f\ \mathbf{U}\ g], R) = CheckEU(Check(f, R), Check(g, R)),$$
$$Check(\mathbf{EG}\, f, R) = CheckEG(Check(f, R)).$$

Процедура $CheckEX$ опирается на определение темпорального оператора $\mathbf{EX}$:

$$CheckEX(f(\overline{v})) = \exists \overline{v}'[f(\overline{v}') \wedge R(\overline{v}, \overline{v}')],$$

где $R(\overline{v}, \overline{v}')$ — это OBDD, представляющая отношение переходов.

Располагая OBDD для f и R, можно вычислить OBDD для

$$\exists \overline{v}'[f(\overline{v}') \wedge R(\overline{v}, \overline{v}')]$$

при помощи операций над QBF.

Символьная верификация моделей для CTL

Процедура *CheckEU* опирается на формулу наименьшей неподвижной точки, описывающую CTL оператор **EU**:

$$\mathbf{E}[f_1 \,\mathbf{U}\, f_2] = \mu Z \,.\, f_2 \vee (f_1 \wedge \mathbf{EX}\, Z)\,.$$

При помощи функции **Lfp** вычисляется последовательность

$$Q_0, Q_1, Q_2, \ldots, Q_i, \ldots\,,$$

множеств состояний, сходящаяся к $\mathbf{E}[f \,\mathbf{U}\, g]$ за конечнное число шагов.

Располагая ROBDD для f, g и текущим приближением Q_i, можно вычислить OBDD для следующего приближения Q_{i+1}.

Поскольку ROBDD — это каноническая форма представления булевых функций, сходимость легко проверить, сравнивая последовательные приближения. Как только выполнится равенство $Q_i = Q_{i+1}$, функция **Lfp** прекратит вычисления.

Множество состояний, в которых выполняется $\mathbf{E}[f \,\mathbf{U}\, g]$, будет представлено ROBDD для Q_i.

Символьная верификация моделей для CTL

Процедура *CheckEG* устроена сходным образом. В ее основе лежит формула наибольшей неподвижной точки для CTL оператора **EG**:

$$\mathbf{EG}\, f_1 = \nu Z \;.\; f_1 \wedge \mathbf{EX}\, Z \;.$$

Располагая OBDD для f_1, можно использовать функцию **Gfp** для вычисления ROBDD, представляющей множество состояний, в которых выполняется формула $\mathbf{EG}\, f_1$.

Символьная верификация моделей для CTL

Метод символьной верификации моделей работает так.

Символьная верификация моделей для CTL

Метод символьной верификации моделей работает так.

1. для каждого оператора act всякого процесса P проверяемой распределенной системы $Syst$ строится ROBDD R_{act} , описывающая отношение переходов на множестве состояний вычисления, которое задается этим оператором.

Символьная верификация моделей для CTL

Метод символьной верификации моделей работает так.

1. для каждого оператора act всякого процесса P проверяемой распределенной системы $Syst$ строится ROBDD R_{act} , описывающая отношение переходов на множестве состояний вычисления, которое задается этим оператором.
2. для каждого процесса P проверяемой распределенной системы $Syst$ строится ROBDD $R_P = \bigcup_{act \in P} R_{act}$, описывающая отношение переходов этого процесса.

Символьная верификация моделей для CTL

Метод символьной верификации моделей работает так.

1. для каждого оператора act всякого процесса P проверяемой распределенной системы $Syst$ строится ROBDD R_{act} , описывающая отношение переходов на множестве состояний вычисления, которое задается этим оператором.
2. для каждого процесса P проверяемой распределенной системы $Syst$ строится ROBDD $R_P = \bigcup_{act \in P} R_{act}$, описывающая отношение переходов этого процесса.
3. для проверяемой распределенной системы $Syst$ в зависимости от вида параллельной композиции строится ROBDD $R_{Syst} = \|_{P \in Syst} R_P$, описывающая отношение переходов системы.

Символьная верификация моделей для CTL

Метод символьной верификации моделей работает так.

1. для каждого оператора act всякого процесса P проверяемой распределенной системы $Syst$ строится ROBDD R_{act} , описывающая отношение переходов на множестве состояний вычисления, которое задается этим оператором.
2. для каждого процесса P проверяемой распределенной системы $Syst$ строится ROBDD $R_P = \bigcup_{act \in P} R_{act}$, описывающая отношение переходов этого процесса.
3. для проверяемой распределенной системы $Syst$ в зависимости от вида параллельной композиции строится ROBDD $R_{Syst} = \|_{P \in Syst} R_P$, описывающая отношение переходов системы.
4. к построенной ROBDD R_{Syst} и CTL спецификации φ применяется процедура $Check(\varphi, R_{Syst})$ и проверяется условие $S_0 \subseteq Check(\varphi, R_{Syst})$.

Символьная верификация моделей для CTL

Метод символьной верификации моделей работает так.

1. для каждого оператора act всякого процесса P проверяемой распределенной системы $Syst$ строится ROBDD R_{act} , описывающая отношение переходов на множестве состояний вычисления, которое задается этим оператором.
2. для каждого процесса P проверяемой распределенной системы $Syst$ строится ROBDD $R_P = \bigcup_{act \in P} R_{act}$, описывающая отношение переходов этого процесса.
3. для проверяемой распределенной системы $Syst$ в зависимости от вида параллельной композиции строится ROBDD $R_{Syst} = \|_{P \in Syst} R_P$, описывающая отношение переходов системы.
4. к построенной ROBDD R_{Syst} и CTL спецификации φ применяется процедура $Check(\varphi, R_{Syst})$ и проверяется условие $S_0 \subseteq Check(\varphi, R_{Syst})$.

На каком этапе возникают самые большие трудности?

Символьная верификация моделей для CTL

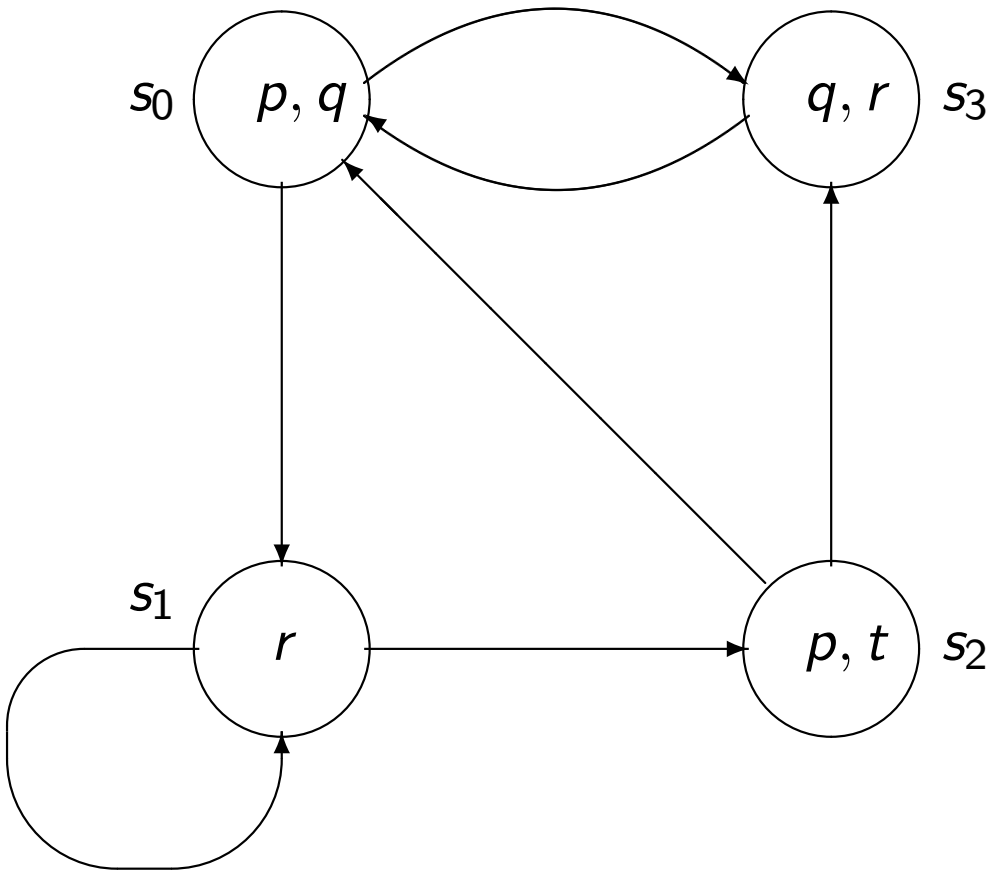
Упражнение.

Примените процедуры вычисления наименьшей неподвижной точки **Lfp** и наибольшей неподвижной точки **Gfp** для проверки выполнимости формул CTL на модели и проверьте, в каких состояний s соблюдаются отношения выполнимости $\mathcal{M}, s \models \varphi$ для приведенных ниже формул на модели $\mathcal{M}$,

- $\mathbf{AF}\, q$
- $\mathbf{AG}(\mathbf{AF}(p \vee r))$
- $\mathbf{EX}\,\mathbf{EX}\, r$
- $\mathbf{AG}\,\mathbf{A}[q\,\mathbf{R}\,p]$

Символьная верификация моделей для CTL

Модель $\mathcal{M}$



КОНЕЦ ЛЕКЦИИ 6.