Введение в моделирование и верификацию аппаратных и программных систем

Лекция 13: Свидетельства и контрпримеры. Абстракция.

Борис Юрьевич Конев

konev@liverpool.ac.uk

Liverpool University

Октябрь-Ноябрь 2007

Свидетельства и контрпримеры

Рассмотрим два подъязыка CTL:

- ACTL: формулы построенные только при помощи **AX**, **AF**, **AG**, **AU** (и отрицания — только “внутри” временных операторов)
- ECTL: формулы построенные только при помощи **EX**, **EF**, **EG**, **EU** (и отрицания — только “внутри” временных операторов)

Если φ — ACTL-формула, то $\neg\varphi$ эквивалентна ECTL-формуле, и наоборот

- Для ECTL формул можно предъявлять свидетельства,
- для ACTL — контрпримеры

Поиск свидетельства для $\mathbf{EG}\psi$

Т.к. $\mathbf{EG}\psi$ — наибольшая неподвижная точка функции

$$H(Z) = [\psi_1] \cap \mathbf{EX}Z$$

если для начальной вершины q_0 мы имеем

$$(S, q_0) \models \mathbf{EG}\psi$$

то найдется последовательность вершин q_0, q_1,... т.ч.

$$(S, q_i) \models \mathbf{EG}\psi \text{ и } (q_i, q_{i+1}) \in T.$$

Так как S конечна, найдутся $i < j : q_i = q_j$

Тогда $q_0, \ldots, q_{i-1}, (q_i, \ldots, q_j)^*$ — свидетельство для $\mathbf{EG}\psi$.

Поиск свидетельства для $\mathbf{EX}\psi$, $\mathbf{EF}\psi$, $\mathbf{E}\psi_1\mathbf{U}\psi_2$

- Для $\mathbf{EX}\psi$, $\mathbf{EF}\psi$ построить путь $q_0, q_1, \ldots, q_f$, такой, что

$$(S, q_f) \models \psi$$

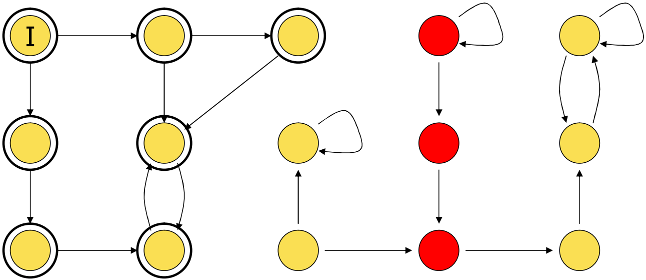
- Для $\mathbf{E}\psi_1\mathbf{U}\psi_2$ построить путь $q_0, q_1, \ldots, q_f$, такой, что

$$(S, q_f) \models \psi_2 \text{ и } (S, q_i) \models \psi_1 \text{ для } 1 \leqslant i < f$$

Контрпримеры и свидетельства

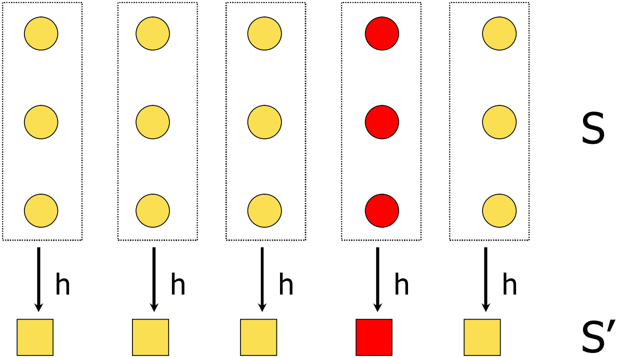
- Контрпример для ACTL формулы φ == свидетельство для ECTL формулы $\neg\varphi$

Контрпример для AGφ

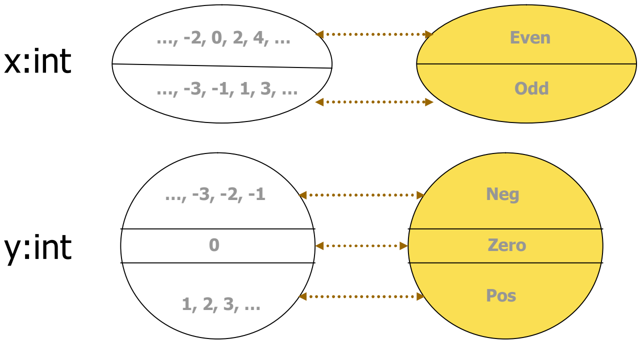


- Система переходов слишком велика, чтобы найти контрпример
- Абстракция
  - Скомбинируем некоторые состояния

Абстракция данных



Пример



Популярный подход

- Разобъем переменные на два множества
  - Видимые (V)
  - Невидимые (I)
- Сгруппируем состояния с идентичными видимыми переменными
- Например,

$\mathbf{x}_1$	$\mathbf{x}_2$	x_3	x_4
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1

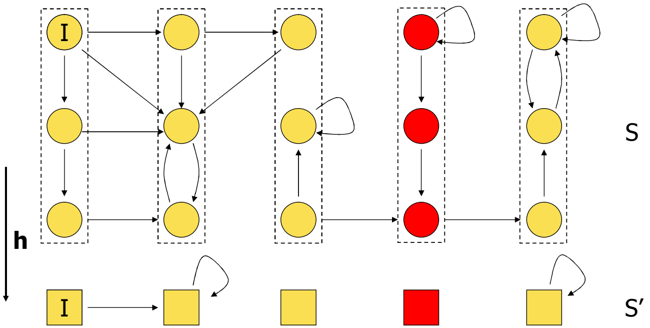
$\longrightarrow$

$\mathbf{x}_1$	$\mathbf{x}_2$
0	0

- Как определить отношение переходов?

Универсальная абстракция

- Перейти из абстрактного состояния, если для всех конкретных состояний, представленных данным абстрактным состоянием, можно сделать переход.

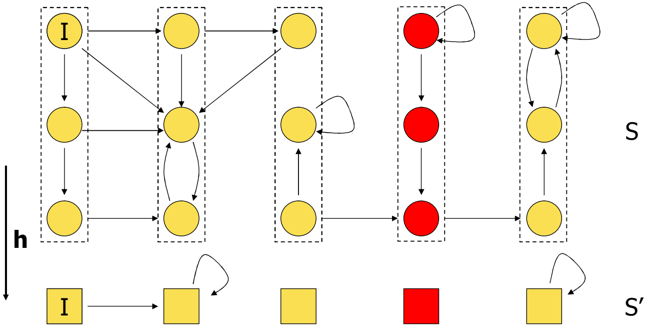


Универсальная абстракция и ECTL формулы

- S_A — универсальная абстракция системы S
- φ — ECTL формула

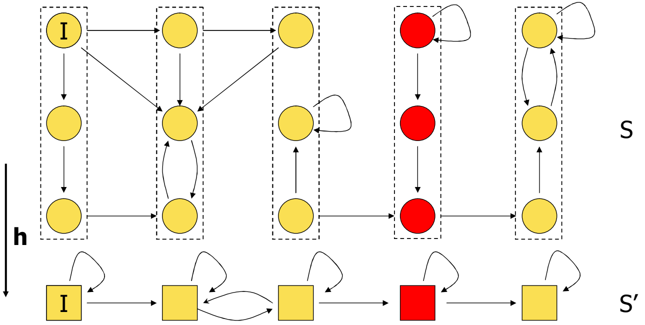
$$S \not\models \varphi \Longrightarrow S_A \not\models \varphi$$

- Поиск ошибок



Экзистенциальная абстракция

- Перейти из абстрактного состояния, если можно перейти из хотя бы одного конкретного состояние, представленного данным абстрактным состоянием

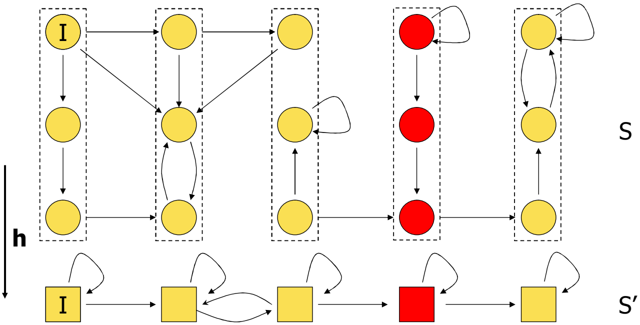


Экзистенциальная абстракция и ACTL формулы

- S_A — экзистенциальная абстракция системы S
- φ — ACTL формула

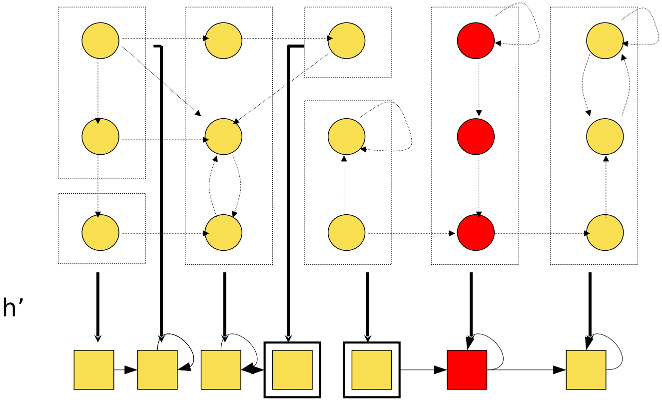
$$S_A \models \varphi \Longrightarrow S \models \varphi$$

- Верификация



- Проблема ложных контрпримеров

Уточнение абстракции



Автоматическое уточнение абстракции

Counter **E**xample **G**uided **A**bstraction **R**efinement (CEGAR)

1. Построить абстрактную систему S_A
2. Проверить ACTL свойство φ
   - Если $S_A \models \varphi$, конкретная система обладает свойством φ.
   - Иначе, найти контрпример (путь π_A в S_A, ведущий в $\neg\varphi$)
     - Если существует путь π в S, $H(\pi) = \pi_A$, то π — контрпример в конкретной системе
     - Иначе, уточнить абстракцию и goto 2

Видимые и невидимые переменные

- Проверим, ложен ли контрпример

$\mathbf{x_1}$	$\mathbf{x_2}$	x_3	x_4
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1

$\longrightarrow$

$\mathbf{x_1}$	$\mathbf{x_2}$
0	0

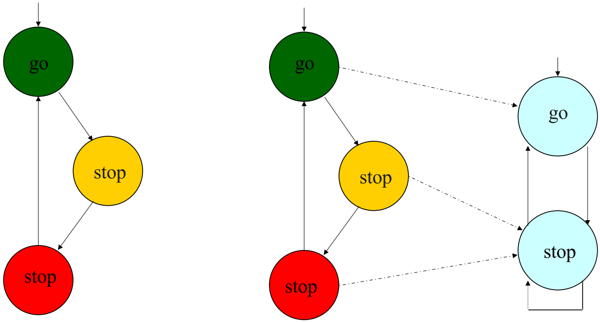
- Если ложен, сделаем какую-то из невидимых переменных видимой

$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$	x_4
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1

$\longrightarrow$

$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$
0	0	0
0	0	1

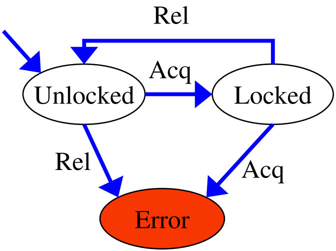
Пример



- $\mathbf{AG}(go \rightarrow \mathbf{AX}stop)$
- $\mathbf{AGF}go$
  - $go, stop, stop, stop, stop, stop \ldots$

Абстракция предикатов

Помните?

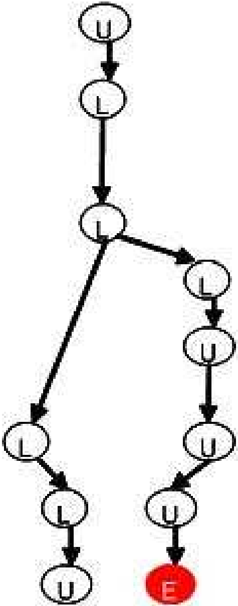



```
do {
    KeAcquireSpinLock();

    nPacketsOld = nPackets;

    if(request){
      request = request->Next;
      KeReleaseSpinLock();
      nPackets++;
    }
} while (nPackets !=
    nPacketsOld);
KeReleaseSpinLock();
```

Булева программа




```
do {
      KeAcquireSpinLock();

      if(*){

            KeReleaseSpinLock();

      }
} while (*);

KeReleaseSpinLock();
```

Ложный контрпример

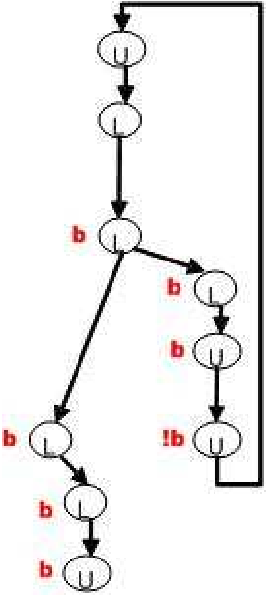
Контрпример ложен из-за того, что условие выхода из цикла `nPackets == nPacketsOld`, но `KeReleaseSpinLock` может исполнится только вместе с `nPackets++`

- Введем булеву переменную `b`

```
b = (nPackets == nPacketsOld)
```

- проследим, как операторы меняют `b`

Уточненная булева программа




```
do {
        KeAcquireSpinLock();

        b := true;

        if(*){

                KeReleaseSpinLock();
                b := b? false : *;
        }
} while ( !b );

KeReleaseSpinLock();
```