

Задача проверки модели

Евтушенко Н.В., Винарский Е.М.

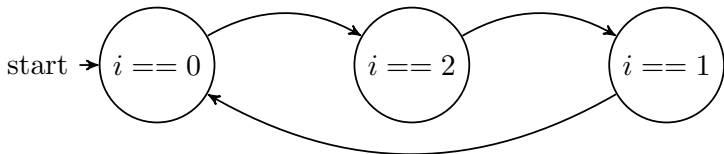
по всем вопросам писать на vinevg2015@gmail.com

9 ноября 2018 г.

- 1 LTL-формулы
- 2 Структура Крипке
- 3 Проверка модели (Model Checking)
- 4 Схема решения задачи проверки модели
- 5 Автоматы Бюхи

Algorithm 1 "Текущий пример"

```
1:  $i = 0$   
2: for true do  
3:    $i = i + 2$   
4:    $i = i - 1$   
5:    $i = i - 1$   
6: end for
```



- Трасса run – бесконечная последовательность событий
- $run[j]$ – j -ое событие трассы τ
- run^j – суффикс трассы τ , начинающийся с j -ого события
- AP – множество атомарных высказываний

В "текущем примере"

- события – $\{i = 0; i = 1; i = 2\}$
- $AP = \{[i == 0], [i == 1], [i == 2]\}$

LTL -формула

- $\phi = a \in AP$
- ϕ – LTL -формула, тогда $\neg\phi$ – LTL -формула
- ϕ_1, ϕ_2 – LTL -формулы, тогда $\phi_1 \wedge \phi_2$ – LTL -формула
- ϕ – LTL -формула, тогда $\mathbb{X}\phi$ – LTL -формула
- ϕ_1, ϕ_2 – LTL -формулы, тогда $\phi_1 \mathbb{U} \phi_2$ – LTL -формула
- ϕ – LTL -формула, тогда $\mathbb{F}\phi$ – LTL -формула
- ϕ – LTL -формула, тогда $\mathbb{G}\phi$ – LTL -формула

LTL-формулы (2)

- $run \models a$: выполняется если и только если $a \in run[0]$
- $run \models \neg\phi$: выполняется если и только если $run \not\models \phi$
- $run \models \psi_1 \wedge \psi_2$: выполняется если и только если $run \models \psi_1$ и $run \models \psi_2$
- $run \models \mathbb{X}\phi$: выполняется если и только если $run^1 \models \phi$ (формула выполнима в следующий момент времени)
- $run \models \psi_1 \mathbb{U} \psi_2$: выполняется если и только если $\exists k, k \geq 0$: $run^k \models \psi_2$ и $run^m \models \psi_1$ для всех $m \in [0, k)$ (в какой-то момент времени выполнится ψ_2 , а до этого всегда выполняется ψ_1)
- $run \models \mathbb{F}\phi$: выполняется если и только если $\exists k \geq 0$: $run^k \models \phi$ (когда-то в будущем выполнится ϕ)
- $run \models \mathbb{G}\phi$: выполняется если и только если $\forall k \geq 0$: $run^k \models \phi$ (всегда в будущем выполнится ϕ)

LTL-формулы (примеры)

LTL-формулы описывают требования к верифицируемой системе

- Если мы отправили запрос, то когда-нибудь в будущем обязательно получим ответ

LTL-формулы (примеры)

LTL-формулы описывают требования к верифицируемой системе

- Если мы отправили запрос, то когда-нибудь в будущем обязательно получим ответ
 $\mathbb{G}(request \Rightarrow \mathbb{F}reply)$
- Если мы отправили сообщение, то не сможем отправить следующее, до тех пор, пока не получим ответ
 $\mathbb{G}(send \Rightarrow \mathbb{X}(\neg send \mathbb{U} recieve))$
- Флаг, отвечающий за то, что система никогда не будет находиться в "тупиковой" ситуации всегда
 $\mathbb{G}(deadlock_flag == false)$
- Если система послала сообщение, то ответ будет обязательно получен, и не наступит момента, когда мы больше не сможем отправлять сообщения
 $\mathbb{G}((send \Rightarrow \mathbb{X}\mathbb{F}) \wedge \mathbb{F}send)$

- Требования к системе описываются на языке *LTL*-формул
- Для формального доказательства факта, что система удовлетворяет этим требованиям, необходима формальная модель системы
- В качестве такой модели выступает структура Крипке
- Задача проверки модели заключается в проверке выполнимости формулы на структуре Крипке, т.е. выполнимость формулы на всех трассах данной системы

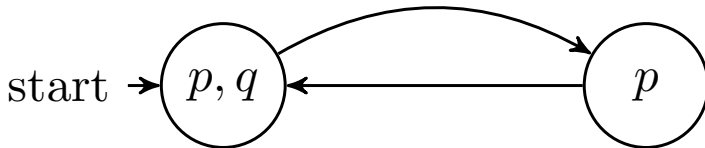
Структура Крипке (формальное определение)

AP – множество атомарных высказываний

Структура Крипке – система $M = (S, S_0, \rightarrow, L)$

- S – конечное непустое множество состояний
- $S_0 \subseteq S$ – конечное непустое множество начальных состояний
- $\rightarrow \subseteq S \times S$ – *тотальное* отношение переходов
- $L : S \rightarrow 2^{AP}$ – функция разметки состояний

Отношение переходов *тотальное*, если для любого состояния $s \in S$ существует $s' \in S$ такое, что существует переход из s в s'



- Путь π из состояния s – бесконечная последовательность состояний вида
 $s \rightarrow s_1 \rightarrow s_2 \dots$
- Трасса $\alpha(\pi)$ пути π – это бесконечная последовательность событий
 $L(s)L(s_1)L(s_2) \dots$
- $\Pi(M)$ – множество всех путей из начальных состояний структуры Крипке M
- $Tr(M) = \{\alpha(\pi) | \pi \in \Pi(M)\}$

Задача проверки модели

ϕ – *LTL*-формула, M – структура Крипке

- Формула ϕ выполняется на пути π в M ($M, \pi \models \phi$), если $\alpha(\pi) \models \phi$
- Формула ϕ выполняется на структуре M ($M \models \phi$), если она выполняется на каждом пути π множества $\Pi(M)$, т.е.
$$Tr(M) \subseteq Tr(\phi)$$

Задача Model Checking – проверить справедливость соотношения
 $M \models \phi$

Схема решения задачи проверки модели

- ❶ По модели M строится автомат Бюхи A_M , распознающий множество бесконечных трасс $Tr(M)$
- ❷ Строится отрицание формулы ϕ , затем по ней строится автомат $A_{\neg\phi}$, распознающий множество бесконечных трасс $Tr(\neg\phi)$
- ❸ Строится автомат A , распознающий множество бесконечных трасс $Tr(M) \cap Tr(\neg\phi)$
- ❹ анализируется язык, распознаваемый автоматом Бюхи $Tr(M) \cap Tr(\neg\phi)$
 - если язык, распознаваемый A – пустой, то $M \models \phi$
 - если язык, распознаваемый A – НЕ пустой, то $M \not\models \phi$ и слова, принадлежащие этому языку – контрпримеры

Автоматом Бюхи (над алфавитом Σ) называется система
 $A = (S, S_0, \rightarrow, F)$

- S – конечное непустое множество состояний
- $S_0 \subseteq S$ – конечное непустое множество начальных состояний
- $\rightarrow \subseteq S \times \Sigma \times S$ – отношение переходов
- $F \subseteq S$ – конечное непустое множество финальных состояний

$\text{inf}(\text{run})$ – состояния, встречающиеся бесконечно часто на трассе run

- Автомат Бюхи работает с бесконечными словами вида $\sigma_1\sigma_2\dots\sigma_n\dots$, где $\sigma_i \in \Sigma$
- Трасса автомата Бюхи – бесконечная последовательность состояний вида: $\text{run} = s_0 \xrightarrow{\sigma_1} s_1 \xrightarrow{\sigma_2} \dots s_{n-1} \xrightarrow{\sigma_n} s_n \dots$
- слово принимается автоматом Бюхи, если и только если $\text{inf}(\text{run}) \cap F \neq \emptyset$

