

Машины на състояния – варианти – Резюме

1. Проблеми при стандартните машини на състояния

- Големият брой състояния дори при прости системи може да усложни анализа.
- Необходимост от компактно представяне на състояния и действия:
 - Използване на предикати за описване на състояния.
 - Дефиниране на целите чрез промени в източника.
 - Фокус върху текстови описания вместо графични за улеснение на кодирането.

2. Основни варианти на машини на състояния

А) Структуриране на състоянията

- Представяне на състоянията като функции от множества променливи към стойности.
Пример: Състоянията на брояч, където всяко състояние е тотална функция с конкретни преходи.

Б) Действия с аргументи и резултати

- Действията могат да се дефинират като функции с аргументи.
Пример:
 - $\text{inc}(i: \text{int})$
 $\text{pre: } i > 0 \quad i > 0$
 $\text{post: } x' = x + i \quad x' = x + i$
- Включване на резултати за наблюдение на действията:

- **Нормално приключване** (e.g., ok) и **извънредно приключване** (e.g., empty).

В) Обобщение на функцията на преходите

- Недетерминистични машини, при които едно действие може да води до множество състояния.

Пример: Random Counter с алтернативни преходи.

Г) Съвместно използване на структуриране на състояния, действия и преходи

- Дефиниране чрез **pre-post нотация**:

action(inputs)/ terminates(outputs)

pre F(v)

post Y(v, v')

3. Спецификация и доказване на инварианти

- **Инвариантност:** Условия, които остават валидни за всички достижими състояния.

Пример: За брояч, инвариантът $x \geq 0$.

- Методи за доказване:
 - Индукция върху състоянията.
 - Анализ на всички действия чрез техните pre/post условия.

4. Видове машини на състояния

- **Детерминистични крайни автомати (DFA):** Всеки вход води до точно един преход.

- **Недетерминистични машини:** Позволяват множество възможни състояния при дадено действие.
- **Модели с крайни и безкрайни пътеки:** Практически използвани за софтуерни системи с ограничено или неограничено поведение.

5. Приложения

- Машини на състояния намират приложение в:
 - Анализ и моделиране на сложни системи.
 - Установяване на справедливост и избягване на "deadlocks".
 - Спецификация на действия и изходи при взаимодействие с потребители.