Языки описания схем

(mk.cs.msu.ru $\to$ Лекционные курсы $\to$ Языки описания схем)

Блок 18

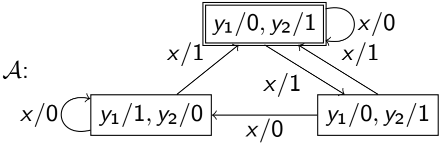
Пара слов о символьных автоматах

Лектор:
Подымов Владислав Васильевич

E-mail:
valdus@yandex.ru

ВМК МГУ, 2019, осенний семестр

Вступление



Разработка автомата для последующей схемной реализации начинается с того, что заданы:

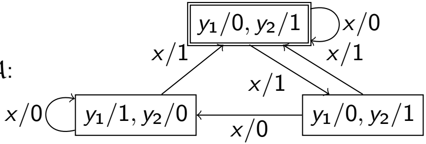
- Входы предполагаемой схемы
  - и входной алфавит автомата — это множество значений на входах
- Выходы предполагаемой схемы
  - и выходной алфавит автомата — это множество значений на выходах

При этом общее количество переходов автомата ($|T|$) **однозначно** задаётся количеством его состояний ($|Q|$) и общим числом булевых разрядов на входах схемы (n):

$$|T| = |Q| \cdot 2^n$$

Вступление

$\mathcal{A}$:



Даже для небольших значений $|Q|$ и n
число $|T|$ может оказаться настолько большим,
что явно “рисовать” каждый переход будет бессмысленно

Например, если $|Q| = 10$ и $n = 10$, то $|T| = 10 \cdot 2^{10} = 10240$ —
непросто удержать в голове десять тысяч переходов
и не ошибиться, когда их описываешь

Далее обсуждается наиболее популярный способ
представления автоматов, позволяющий наглядно и коротко
записывать большие семейства похожих переходов

Вступление

Начнём с примера: предположим, что в автомате, построенном для одноразрядных входных сигналов x, y, u, v, содержатся следующие 8 переходов:

$$q_1 \xrightarrow{0010} q_2 \qquad q_1 \xrightarrow{0100} q_2 \qquad q_1 \xrightarrow{1000} q_2 \qquad q_1 \xrightarrow{1110} q_2$$
$$q_1 \xrightarrow{0011} q_2 \qquad q_1 \xrightarrow{0101} q_2 \qquad q_1 \xrightarrow{1001} q_2 \qquad q_1 \xrightarrow{1111} q_2$$

Выпишем отдельно множество наборов значений на переходах:

$$S = \{(0010), (0011), (0100), (0101), (1000), (1001), (1110), (1111)\}$$

Множество S можно трактовать как множество наборов, на которых *4-местная булева функция* f_S принимает значение 1 [1]

Эта функция реализуется, например, формулой $x \oplus y \oplus u$

Тогда обозначенные 8 переходов можно записать в виде одного **символьного перехода**:

$$q_1 \xrightarrow{x \oplus y \oplus u} q_2$$

[1] У этой функции даже есть своё название: характеристическая функция множества

Предикатные формулы

Доменом назовём систему, состоящую из:

- Набора переменных $x_1, \dots, x_n$
- Множества значений D_{x_i} каждой переменной x_i
- Множества базовых предикатов: отображений вида
$$p : D_{x_1} \times \dots \times D_{x_n} \to \{0, 1\}$$
  - *простыми словами:*
    базовый предикат для каждого набора значений переменных говорит
    "да" (1; набор значений входит в предикат) или
    "нет" (0; набор значений не входит в предикат)

Предикатные формулы (*над заданным доменом*) получаются из булевых формул заменой каждой переменной на базовый предикат

Например, если $(x_1 = x_2)$ и $(x_2 < x_3)$ — базовые предикаты, то $(x_1 = x_2) \,\&\, \neg(x_2 < x_3)$ — предикатная формула

Множество всех предикатных формул обозначим символом $\mathfrak{P}$

Предикатные формулы

Каждой предикатной формулой φ (над переменными $x_1, \ldots, x_n$) реализуется отображение $[\varphi] : D_{x_1} \times \cdots \times D_{x_n} \to \{0, 1\}$ (*то есть предикат*) следующего вида:

- если φ — базовый предикат, то $[\varphi] = \varphi$
- $[f(\varphi_1, \ldots, \varphi_k)] = f([\varphi_1], \ldots, [\varphi_k])$ для любой k-местной булевой функции f

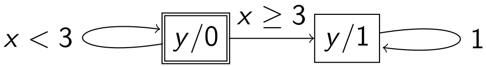
Пример: если домен состоит из переменных x_1, x_2, x_3, множеств значений $D_{x_1} = D_{x_2} = D_{x_3} = \{0, 1, 2\}$ и *естественных* предикатов сравнения чисел $(x_1 = x_2)$ и $(x_2 < x_3)$, то $[(x_1 = x_2) \,\&\, (x_2 < x_3)] = f : \{0, 1, 2\}^3 \to \{0, 1\}$, где $f(i_1, i_2, i_3) = 1 \Leftrightarrow (i_1, i_2, i_3) \in \{(0, 0, 1), (0, 0, 2), (1, 1, 2)\}$

Символьные автоматы

Символьный автомат(*-преобразователь Мура*)[1]
над доменом $\mathcal{D}$ и выходным алфавитом O —
это система $(Q, q_0, B, \mathfrak{T})$, где

- компоненты Q, q_0, B
  (множество состояний, начальное состояние, функция выхода)
  имеют те же устройство смысл, что и в *обычном* автомате
- $\mathfrak{T} \subseteq Q \times \mathfrak{P} \times Q$ — множество символьных переходов
  - $q_1 \xrightarrow{\varphi} q_2$ — графовое изображение перехода (q_1, φ, q_2)

Пример символьного автомата
над базовыми предикатами $x < 3$, $x \geq 3$ и 1:



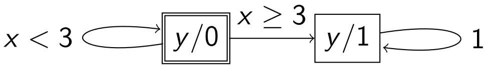
[1] На самом деле символьный автомат — более широкое понятие, а здесь оно упрощено и приближено к разработке схем

Символьные автоматы

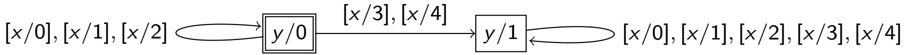
Выполнение символьного автомата $\mathfrak{A}$ определим
как выполнение *обычного* автомата $\mathcal{A}$, соответствующего автомату $\mathfrak{A}$:

- Посмотрим на автомат $\mathfrak{A}$ *как на размеченный граф*
- Заменим каждую дугу $q \xrightarrow{\varphi} q'$ графа $\mathfrak{A}$ на набор дуг:
  $q \xrightarrow{[x_1/v_1,\ldots,x_n/v_n]} q'$ для каждого набора значений $v_1, \ldots, v_n$
  переменных $x_1, \ldots, x_n$, такого что $[\varphi](v_1, \ldots, v_n) = 1$
- Если получившийся граф G' является автоматом Мура, то $\mathcal{A} = G'$

Например, если x — единственная переменная домена,
$D_x = \{0, 1, 2, 3, 4\}$, и $x < 3$, $x \geq 3$, 1 — *естественные* предикаты,
то символьный автомат



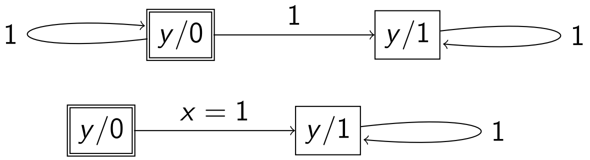
соответствует *обычному* автомату



Символьные автоматы

Символьному автомату
не всегда соответствует *обычный* автомат[1]

Например:



В какое состояние автомат, соответствующий какому-либо из этих двух, перейдёт из левого состояния по букве $[x/0]$?

[1] Более точно, соответствующий автомат всегда существует, но не всегда является **детерминированным**.
Здесь полагается, что недетерминированных автоматов не существует

Детерминированные символьные автоматы

Символьный автомат (Q, q_0, B, T) назовём детерминированным, если для него справедливо следующее:

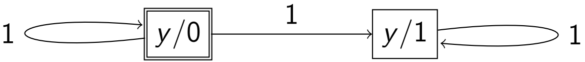
- если $q \xrightarrow{\varphi_1} q_1$ и $q \xrightarrow{\varphi_2} q_2$ — две различные дуги́, то $[\varphi_1 \,\&\, \varphi_2]$ — функция, всюду принимающая значение 0
  - то есть из каждого состояния по каждому символу можно перейти **не более чем по одной** дуге
- если $q \xrightarrow{\varphi_1} q_1, \quad \dots, \quad q \xrightarrow{\varphi_k} q_k$ — все ду́ги, исходящие из q, то $[\varphi_1 \vee \dots \vee \varphi_k]$ — функция, всюду принимающая значение 1
  - то есть из каждого состояния по каждому символу можно перейти **хотя бы по одной** дуге

Домен назовём конечным, если множество значений каждой переменной этого домена конечно

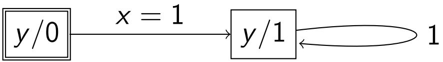
Очевидное (?) утверждение:
для любого детерминированного символьного автомата над конечным доменом существует единственный *обычный* автомат

Детерминированные символьные автоматы

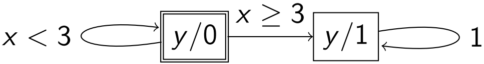
Примеры: (*все предикаты — естественные*)



Этот автомат не является детерминированным:
из левого состояния исходят различные ду́ги, помеченные формулой 1, и $[1 \,\&\, 1]$ — функция, всюду принимающая значение 1



Этот автомат также не является детерминированным:
из левого состояния не исходит ни одной дуги $\cdot \xrightarrow{\varphi} \cdot$, такой что $[\varphi](0) = 1$



Это детерминированный автомат

Вспоминаем про схемы

Ранее в лекциях встречалась такая фраза:

“*Наиболее часто встречающийся в схемотехнике вид автоматов — это конечные детерминированные автоматы-преобразователи Мура*”

Теперь эту фразу можно уточнить, приблизив к реальности:

Наиболее часто встречающийся в схемотехнике вид автоматов — это детерминированные символьные автоматы-преобразователи Мура над конечными доменами

Вспоминаем про схемы

Более точно, домен $\mathcal{D}$ символьного автомата, предназначенного для реализации схемой, обычно устроен так:

- Переменные домена — это все входы схемы, кроме тактового входа и входа сброса
- Множество значений каждого входа имеет вид $\{0, 1, \ldots, 2^k - 1\}$, где k — разрядность этого входа
- Множество предикатов — это все *булевы выражения* выбранного языка описания схем
  - *Первое уточнение.* Одноразрядные входы схемы, как правило, используются в качестве особых предикатов: селекторных функций
  - *Второе уточнение.* в языке могут использоваться и *не совсем* булевы выражения, но аппаратная семантика таких *почти булевых* выражений, как правило, достаточно близка к булевой