Языки описания схем

mk.cs.msu.ru → Лекционные курсы → Языки описания схем

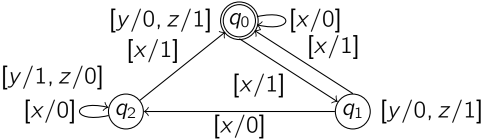
Блок 27

Пара слов о символьных автоматах

Лектор:
Подымов Владислав Васильевич
E-mail:
valdus@yandex.ru

ВМК МГУ, 2025, сентябрь–декабрь

Вступление



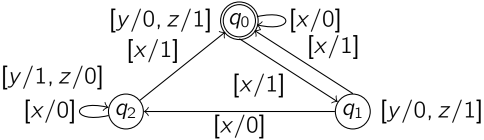
Разработка автомата для последующей схемной реализации начинается с того, что заданы:

- Входы предполагаемой схемы
  - Входной символ автомата — это набор значений на входах
- Выходы предполагаемой схемы
  - Выходной символ автомата — это набор значений на выходах

При этом общее количество переходов автомата ($|T|$) однозначно задаётся количеством его состояний ($|Q|$) и суммарной шириной всех входных шин (n):

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

Вступление



Например, если $|Q| = 10$ и $n = 10$, то $|T| = 10 \cdot 2^{10} = 10240$

Правильно разработать такой автомат вручную (на бумаге или с помощью инструментов рисования «обычных» автоматов) непросто: с таким количеством переходов трудно совладать не ошибившись

При этом и 10 состояний, и 10 разрядов на входе схемы — это немного

Поэтому полезно иметь средства «сжатой» записи переходов автоматов, достаточно простой для разработки, поддержки и отладки

На помощь здесь приходят

- логика предикатов (*не надо бояться, рассмотрим только очень простой фрагмент*) и
- символьные автоматы (*обычные автоматы + эта логика*)

Формулы из практики

Если автомат предназначен для реализации на *достаточно высокоуровневом* языке проектирования схем (например, $\mathcal{V}$), то в этом языке, как правило, в том или ином виде есть следующие конструкции:

- Переменные, обозначающие значения на входах схемы, отвечающей автомату
  - В $\mathcal{V}$ это входные порты-соединения
- Для каждой переменной — тип, которым задаётся соответствующий конечный домен (конечное множество допустимых значений переменной)
  - В $\mathcal{V}$ это, например, $\mathbb{Z}_{(n)}$ для беззнаковой точки ширины n
- Константы, обозначающие конкретные значения конкретных типов
- Операции, с помощью которых можно комбинировать константы и переменные в составные выражения со своими типами

Формулы из практики

Среди операций, как правило, особо выделяются две группы:

- Булевы операции, такие как $\vee$, $\&$ и $\neg$
- Отношения принимают в качестве аргументов небулевы значения и возвращают одно из значений «истина», «ложь» (1, 0)

Это означает, что некоторые выражения представляют собой формулы, в которых булевыми операциями связаны примитивы, описывающие отношения над доменами переменных

Более точно, это бескванторные формулы логики предикатов, но не будем перегружать рассказ и обсуждать логику предикатов: введём подходящее формулы

- строго, но при этом
- настолько упрощённо, чтобы можно было легко осознать язык для практической разработки автоматов

Для ясности будем далее называть эти формулы предикатными, тем самым противопоставляя их булевым формулам

Предикатные формулы

Далее будем считать заданными:

- Набор переменных $x_1, \ldots, x_n$
- Домен D_i каждой переменной x_i, $i \in \{1, \ldots, n\}$, то есть множество значений этой переменной
- Множество $\mathcal{P}$ базовых предикатов: отображений вида
$$p : D_1 \times \cdots \times D_n \to \{0, 1\}$$

Предикатом (над заданными доменами) будем называть всякое отображение вида $p : D_1 \times \cdots \times D_n \to \{0, 1\}$

Предикат p также будем считать множеством $p \subseteq D_1 \times \cdots \times D_n$:
$$p(d_1, \ldots, d_n) = 1 \quad \Leftrightarrow \quad (d_1, \ldots, d_n) \in p$$

Оценку $[x_1/d_1, \ldots, x_n/d_n]$ будем отождествлять с набором $(d_1, \ldots, d_n)$

Предикатные формулы

Особо будем выделять предикаты

- $\mathbb{t} = D_1 \times \cdots \times D_n$ и
- $\mathbb{f} = \emptyset$

В примерах будем использовать естественные арифметические предикаты:

- $x_i = d$, где $d \in D_i$
  - Этот предикат содержит все наборы с оценкой x_i/d
- $x_i > d$, где $d \in D_i$ и домен D_i содержит только целые числа
  - Этот предикат содержит все наборы с оценкой x_i/e, где $e > d$
- $x_i = x_j$
  - Этот предикат содержит все наборы с оценками x_i/d и x_j/e, где $d = e$
- ...

Предикатные формулы

Синтаксис предикатных формул совпадает с синтаксисом булевых формул над множеством переменных $\mathcal{P}$

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 $\mathfrak{P}$

Каждой предикатной формулой φ реализуется предикат $[\varphi]$ следующего вида:

- если $\varphi \in \mathcal{P}$, то $[\varphi] = \varphi$
- $[f(\varphi_1, \ldots, \varphi_k)] = f([\varphi_1], \ldots, [\varphi_k])$
  для любой k-местной булевой функции f

Пример: для переменных x_1, x_2, x_3 с доменами $D_1 = D_2 = D_3 = \{0, 1, 2\}$ формулой $(x_1 = x_2) \,\&(x_2 < x_3)$ реализуется предикат

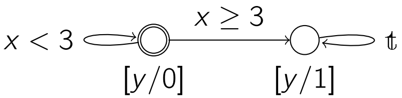
$$[(x_1 = x_2) \,\&(x_2 < x_3)] = \{(0,0,1), (0,0,2), (1,1,2)\} \subseteq \{0,1,2\}^3$$

Символьные автоматы

Рассматривающийся здесь вариант символьного автомата над переменными $x_1, \dots, x_n$, доменами $D_1, \dots, D_n$, множеством базовых предикатов $\mathcal{P}$ и выходным алфавитом O — это система $(Q, q_0, B, \mathcal{T})$, где:

- Q, q_0 и B — те же компоненты, что и в «обычном» автомате: конечное множество состояний, начальное состояние ($q_0 \in Q$) и функция выхода ($B : Q \to O$)
- $\mathcal{T} \subseteq Q \times \mathfrak{P} \times Q$ — множество символьных переходов
    - Переход (q_1, φ, q_2) изображается как помеченная дуга $q_1 \xrightarrow{\varphi} q_2$

Пример символьного автомата
над базовыми предикатами $x < 3$, $x \geq 3$ и $\mathbb{t}$:



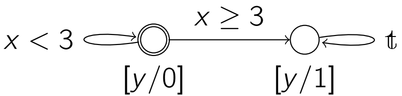
Символьные автоматы

Семантику символьного автомата можно задать через соответствующий недетерминированный «обычный» автомат: такой, в котором функция переходов может быть и многозначна

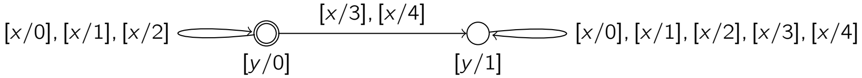
Символьному автомату $(Q, q_0, B, \mathcal{T})$ соответствует автомат (Q, q_0, B, T) над тем же выходным алфавитом и над входным алфавитом всех оценок переменных $x_1, \ldots, x_n$, где функция переходов T задаётся соотношением

$$q_2 \in T(q_1, \xi) \Leftrightarrow \exists \varphi : (q_1, \varphi, q_2) \in \mathcal{T} \text{ и } \xi \in [\varphi]$$

Например, если $x = x_1$ — единственная переменная и $D_1 = \{0, 1, 2, 3, 4\}$, то символьный автомат



соответствует «обычному» автомату



Детерминированные символьные автоматы

«Обычный» автомат называется детерминированным (*точнее, детерминированным полным*), если для любых состояния q и входного символа a в нём из q исходит ровно один переход, помеченный a

Символьный автомат называется детерминированным, если ему соответствует детерминированный «обычный» автомат

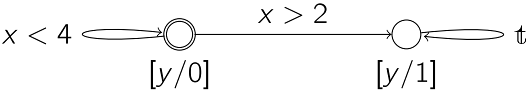
Определение детерминированности можно переформулировать так:

1. Для любых формул φ_1, φ_2, помечающих два разных перехода из одного состояния, верно $[\varphi_1 \,\&\, \varphi_2] = \mathbb{f}$
   - «... исходит не более одного перехода ...»
2. Если $\varphi_1, \dots, \varphi_n$ — все формулы, помечающие переходы из одного произвольно выбранного состояния, то $[\varphi_1 \vee \dots \vee \varphi_n] = \mathbb{t}$
   - «... исходит не менее одного перехода ...»

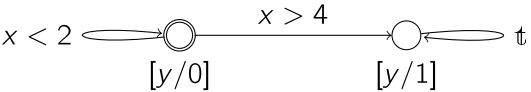
Символьный автомат конечен, если конечны домены всех переменных, выходной алфавит и множества состояний и переходов

Детерминированные символьные автоматы

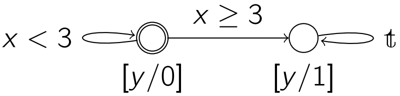
Примеры:



Этот автомат не является детерминированным: $[(x < 4)\,\&(x > 2)] \neq fa$



Этот автомат не является детерминированным: $[(x < 2) \vee (x > 4)] \neq \mathbb{t}$



А это детерминированный автомат

Заключение

Ранее в лекциях встречался такой тезис с пометкой, что он будет уточнён:

«Наиболее часто встречающийся в схемотехнике вид автоматов — это конечные детерминированные автоматы-преобразователи Мура»

Уточнение этого тезиса:

Наиболее часто встречающийся в схемотехнике вид автоматов — это конечные детерминированные символьные автоматы-преобразователи Мура

В таких автоматах переменные — это входные порты схемы, домены переменных задаются их типами в языке проектирования, и базовые предикаты отвечают возможностям построения булевых выражений в языке