Языки описания схем

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

Блок 27

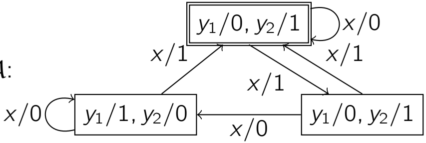
Пара слов о символьных автоматах

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

ВМК МГУ, 2024/2025, осенний семестр

Вступление

$\mathcal{A}$:



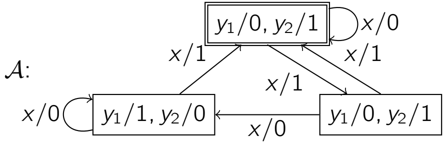
Разработка автомата для последующей схемной реализации начинается с того, что заданы:

- 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 суммарной шириной всех входных шин (n):

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

Вступление



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

То есть даже в автоматах, относительно небольших по числу состояний и размеру входов, оказывается столько переходов, что явно «рисовать» каждый переход оказывается бессмысленно

На практике при разработке автоматов для схемной реализации используются «сжатые» представления автоматов, в которых наглядно и коротко обозначаются большие семейства схожих переходов

Вступление

Начнём с примера: предположим, что в автомате со входами x, y, u, v ширины 1 содержатся такие 8 переходов (оценка $[x/\alpha, y/\beta, u/\gamma, v/\delta]$ изображена как набор $(\alpha, \beta, \gamma, \delta)$):

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

Выпишем множество наборов значений, помечающих эти переходы:

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

S можно понимать как характеристическое множество булевой функции местности 4: $f_S(x, y, u, v) = 1 \Leftrightarrow (x, y, u, v) \in S$

Функцию f_S можно представить, например, формулой $x \oplus y \oplus u$

Тогда рассматриваемые 8 переходов можно «сжато» изобразить в виде одного, помеченного этой формулой:

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

Это символьный переход, задающий множество переходов для всех меток-оценок, выполняющих формулу, которой он помечен

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, $\{0, 1, \ldots, 2^n - 1\}$ 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 его легко осознать для практической разработки

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
- $\mathbb{f} = \emptyset$

Кроме того, в примерах будем использовать естественные арифметические предикаты, такие как:

- $x_i = d$, 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

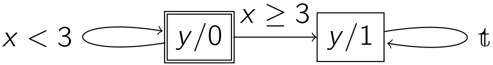
$$[(x_1 = x_2) \,\&(x_2 < x_3)] = \{(0,0,1),(0,0,2),(1,1,2)\} \subseteq \{0,1,2\}^3$$

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

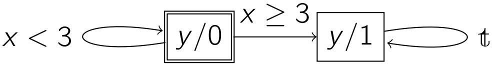
Семантику символьного автомата можно задать как соответствующий недетерминированный «обычный» автомат

В связи с соответствием схем и автоматов полное определение недетерминированного автомата не понадобится, и достаточно только понимать, что в таком автомате функция переходов многозначна

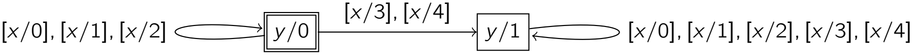
Символьному автомату $(Q, q_0, B, \mathcal{T})$ соответствует автомат (Q, q_0, B, T) над тем же выходным алфавитом и над входным алфавитом всех оценок переменных $x_1, \ldots, x_n$, где функция переходов T задаётся соотношением

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

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



соответствует «обычному» автомату



Детерминированные символьные автоматы

Символьный автомат $\mathcal{A}$ назовём детерминированным, если для него справедливо следующее:

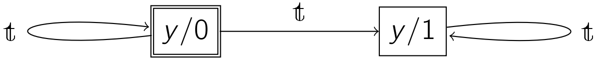
- если $q \xrightarrow{\varphi_1} q_1$ и $q \xrightarrow{\varphi_2} q_2$ — два различных перехода в $\mathcal{A}$, то $[\varphi_1 \,\&\, \varphi_2] = \mathbb{f}$
  - это означает, что из каждого состояния по каждому символу можно перейти **не более чем по одной** дуге
- если $q \xrightarrow{\varphi_1} q_1, \quad \dots, \quad q \xrightarrow{\varphi_k} q_k$ — все переходы в $\mathcal{A}$, исходящие из произвольно выбранного состояния q, то $[\varphi_1 \vee \dots \vee \varphi_k] = \mathbb{t}$
  - это означает, что из каждого состояния по каждому символу можно перейти **хотя бы по одной** дуге

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

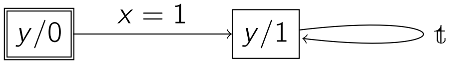
Утверждение. Любому детерминированному конечному символьному автомату соответствует детерминированный конечный «обычный» автомат

Детерминированные символьные автоматы

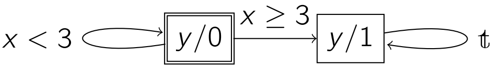
Примеры:



Этот автомат не является детерминированным:
из левого состояния исходят различные ду́ги, помеченные формулой $\mathbb{t}$,
и $[\mathbb{t}\,\&\,\mathbb{t}] = \mathbb{t} \neq \mathbb{f}$



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



А 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:

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 (или «почти» булевых) выражений в языке