Булева и многозначные логики
Симметричные системы счисления
Троичная информатика

Д. В. Луцив

Кафедра системого программирования СПбГУ

CS103

Содержание

Мощность пространства функций

- Пространство $P = \{f | X \to Y\}$
  - то же самое, что Y^X
  - для конечных пространств:

$$X = X_1 \times X_2 \Rightarrow |X| = |X_1||X_2|$$

$$|P| = |Y|^{|X|} = |Y|^{|X_1||X_2|}$$

Булевы функции

- $B = \{\text{Истина, Ложь}\}$
- $P_n = \{f | B^n \to B\}$ — n-арные функции
  - частичное применение — $f(x_1, ..., x_k) : B^k \to P_{n-k}$

Состав B^n

$$|P_n| = |B|^{|B^n|} = 2^{2^n}$$

$n \leq 2$

1. $n = 0$ — 2 константные функции
2. $n = 1$ — И, Л, $x_1, \neg x_1$
3. $n = 2$ — 16 функций

Базис пространства

Любая функция из P_n выражается при помощи связок И, ИЛИ и НЕ в конъюнктивной или дизъюнктивной нормальных формах исходя из таблицы истинности.

- Стрелка Пирса $x \downarrow y = \neg(x \vee y) = \overline{x} \wedge \overline{y}$
- Штрих Шефера $x|y = \neg(x \wedge y) = \overline{x} \vee \overline{y}$

Базисные функции — основа и для элементной базы. Пример для стрелки Пирса:

- $\neg x = \neg(x \vee x) = x \downarrow x$ — дальше можем пользоваться $\neg x$;
- $x \vee y = \neg\neg(x \vee y) = \neg(x \downarrow y)$ — дальше знаем $x \vee y$;
- $x \wedge y = \neg(\overline{x} \vee \overline{y})$.

Зная выражение для И, ИЛИ и НЕ, можем через нормальную форму выразить любую функцию.

▸ Алгебра логики в Википедии ▸ Булева алгебра в Википедии

Базис пространства

Любая функция из P_n выражается при помощи связок И, ИЛИ и НЕ в конъюнктивной или дизъюнктивной нормальных формах исходя из таблицы истинности.

- Стрелка Пирса $x \downarrow y = \neg(x \vee y) = \overline{x} \wedge \overline{y}$
- Штрих Шефера $x|y = \neg(x \wedge y) = \overline{x} \vee \overline{y}$

Базисные функции — основа и для элементной базы. Пример для стрелки Пирса:

- $\neg x = \neg(x \vee x) = x \downarrow x$ — дальше можем пользоваться $\neg x$;
- $x \vee y = \neg\neg(x \vee y) = \neg(x \downarrow y)$ — дальше знаем $x \vee y$;
- $x \wedge y = \neg(\overline{x} \vee \overline{y})$.

Зная выражение для И, ИЛИ и НЕ, можем через нормальную форму выразить любую функцию.

▸ Алгебра логики в Википедии ▸ Булева алгебра в Википедии

Базис пространства

Любая функция из P_n выражается при помощи связок И, ИЛИ и НЕ в конъюнктивной или дизъюнктивной нормальных формах исходя из таблицы истинности.

- Стрелка Пирса $x \downarrow y = \neg(x \vee y) = \overline{x} \wedge \overline{y}$
- Штрих Шефера $x|y = \neg(x \wedge y) = \overline{x} \vee \overline{y}$

Базисные функции — основа и для элементной базы. Пример для стрелки Пирса:

- $\neg x = \neg(x \vee x) = x \downarrow x$ — дальше можем пользоваться $\neg x$;
- $x \vee y = \neg\neg(x \vee y) = \neg(x \downarrow y)$ — дальше знаем $x \vee y$;
- $x \wedge y = \neg(\overline{x} \vee \overline{y})$.

Зная выражение для И, ИЛИ и НЕ, можем через нормальную форму выразить любую функцию.

▸ Алгебра логики в Википедии ▸ Булева алгебра в Википедии

Многозначная троичная (одна из)

Числовой эквивалент

$$n : T = \{\text{Л, Н, И}\} \to \{-1, 0, 1\}$$

$$B \subset T$$

Значение (t)	$n(t)$	Смысл
И	1	Истина
Н	0	Неизвестность/спорность
Л	-1	Ложь

Континуальная нечёткая (одна из)

Числовой эквивалент

$$n : F = [\text{Л}, \text{И}] \to [-1, 1]$$

$$T \subset F$$

Значение (t)	$n(t)$	Смысл
И	1	Истина
	$(0, 1)$	Истина с разной степенью уверенности
Н	0	Неизвестность/спорность
	$(-1, 0)$	Ложь с разной степенью уверенности
Л	-1	Ложь

Операция		Семантика
$t_1 \wedge t_2$	$=$	$n^{-1}(\min(n(t_1), n(t_2))$
$t_1 \vee t_2$	$=$	$n^{-1}(\max(n(t_1), n(t_2))$
$\neg t$	$=$	$n^{-1}(-n(t))$

Консервативность

$$B \subset T \subset F$$

$$\forall x \in B \; \forall y \in B \quad x \wedge_{F,T} y = x \wedge_B y \in B$$

$$\forall x \in T \; \forall y \in T \quad x \wedge_F y = x \wedge_T y \in T$$

Аналогично, $\vee_F, \neg_F : F \to F$, $T \to T$ и $B \to B$

Описанные логики является консервативными расширениями булевой.

Консервативность

$$B \subset T \subset F$$

$$\forall x \in B \; \forall y \in B \quad x \wedge_{F,T} y = x \wedge_B y \in B$$

$$\forall x \in T \; \forall y \in T \quad x \wedge_F y = x \wedge_T y \in T$$

Аналогично, $\vee_F, \neg_F : F \to F \; , T \to T$ и $B \to B$

Описанные логики является консервативными расширениями булевой.

Полнота

Через $\wedge, \neg, \vee$ можно, как и в классической булевой логике, выразить все нужные функции. Однако, в зависимости от применения, эти функции могут расширяться по-разному.

Например, импликация может быть не только «материальной» ($t_1 \Rightarrow t_2 \leftrightarrow \neg t_1 \vee t_2$), но и ещё много какой. Например, м.б. импликации Лукасевича (часть т.н. модальной логики), Гейтинга.

См. ▸ Троичная логика ▸ Тринари ▸ Троичные ЭВМ

Полнота

Через $\wedge, \neg, \vee$ можно, как и в классической булевой логике, выразить все нужные функции. Однако, в зависимости от применения, эти функции могут расширяться по-разному. Например, импликация может быть не только «материальной» ($t_1 \Rightarrow t_2 \leftrightarrow \neg t_1 \vee t_2$), но и ещё много какой. Например, м.б. импликации Лукасевича (часть т.н. модальной логики), Гейтинга.

См. ▸ Троичная логика ▸ Тринари ▸ Троичные ЭВМ

Основание систем нечетное, одна цифра изображает числа $c_k \in [-T, T], B = 2T + 1$. Число представляется так же, как и для неотрицательных систем.

$$v = \sum_{k=1}^{N} c_k B^{k-1}$$

Ценные свойства:

- отбрасыванием ненужных цифр происходит округление в сторону ближайшего, а не меньшего, целого;
- отрицательные числа представляются естественным образом, так же, как и положительные.

Основание систем нечетное, одна цифра изображает числа $c_k \in [-T, T], B = 2T + 1$. Число представляется так же, как и для неотрицательных систем.

$$v = \sum_{k=1}^{N} c_k B^{k-1}$$

Ценные свойства:

- отбрасыванием ненужных цифр происходит округление в сторону ближайшего, а не меньшего, целого;
- отрицательные числа представляются естественным образом, так же, как и положительные.

Однозначность положительных чисел

- По аналогии с неотрицательными системами, покажем, что

$$1 \times B^N > \sum_{k=1}^{N} \frac{B-1}{2} \times B^{k-1}$$

- Это верно, так как ранее уже было доказано более сильное утверждение.
- Аналогично, если две записи одного и того же числа отличаются, то отличия проявляются вплоть до какой-то цифры. Отличия в самой старшей отличающейся цифре должны компенсироваться младшими.
- Но это, как мы доказали, невозможно, т.к. никакой набор младших цифр не будет давать значение, равное значению единицы в старшей цифре.

Однозначность положительных чисел

- По аналогии с неотрицательными системами, покажем, что

$$1 \times B^N > \sum_{k=1}^{N} \frac{B-1}{2} \times B^{k-1}$$

- Это верно, так как ранее уже было доказано более сильное утверждение.
- Аналогично, если две записи одного и того же числа отличаются, то отличия проявляются вплоть до какой-то цифры. Отличия в самой старшей отличающейся цифре должны компенсироваться младшими.
- Но это, как мы доказали, невозможно, т.к. никакой набор младших цифр не будет давать значение, равное значению единицы в старшей цифре.

Однозначность положительных чисел

- По аналогии с неотрицательными системами, покажем, что

$$1 \times B^N > \sum_{k=1}^{N} \frac{B-1}{2} \times B^{k-1}$$

- Это верно, так как ранее уже было доказано более сильное утверждение.
- Аналогично, если две записи одного и того же числа отличаются, то отличия проявляются вплоть до какой-то цифры. Отличия в самой старшей отличающейся цифре должны компенсироваться младшими.
- Но это, как мы доказали, невозможно, т.к. никакой набор младших цифр не будет давать значение, равное значению единицы в старшей цифре.

Однозначность представления

Положительных

доказали

Отрицательных

аналогично

0

аналогично

Однозначность представления

Положительных

доказали

Отрицательных

аналогично

0

аналогично

Однозначность представления

Положительных

доказали

Отрицательных

аналогично

0

аналогично

```
(define (val->symn value base)
  (let ((half (/ (- base 1) 2)))
    (if (= value 0)
        '() ; no digits for 0
        (let* (
                    (predigit (modulo value base))
                    (digit
                      (if (> predigit half)
                          (- predigit base)
                          predigit
                    )))
; When digit is out of range, normalize it
; It will converge despite of negative digits
          (append
           (val->symn (/ (- value digit) base) base)
           (list digit)
           )
          )
        )
    )
  )
```

Система счисления

- Система строится на цифрах -1, 0, 1. Для записи -1 одним знаком пишут $\overline{1}$.
- Система применялась в машине «Сетунь» (Н. П. Брусенцов, 1959 г., ВЦ МГУ)
- Сложение поразрядное с переносом 0, 1 или -1.
- Умножение производится тоже по классической схеме, путем сдвига и сложения. Деление — путем сдвига и вычитания.
- По мнению проф. Д. Э. Кнута, троичные информатика и математика должны быть более изящными, чем двоичные.

Система счисления

- Система строится на цифрах -1, 0, 1. Для записи -1 одним знаком пишут $\overline{1}$.
- Система применялась в машине «Сетунь» (Н. П. Брусенцов, 1959 г., ВЦ МГУ)
- Сложение поразрядное с переносом 0, 1 или -1.
- Умножение производится тоже по классической схеме, путем сдвига и сложения. Деление — путем сдвига и вычитания.
- По мнению проф. Д. Э. Кнута, троичные информатика и математика должны быть более изящными, чем двоичные.

Система счисления

- Система строится на цифрах -1, 0, 1. Для записи -1 одним знаком пишут $\overline{1}$.
- Система применялась в машине «Сетунь» (Н. П. Брусенцов, 1959 г., ВЦ МГУ)
- Сложение поразрядное с переносом 0, 1 или -1.
- Умножение производится тоже по классической схеме, путем сдвига и сложения. Деление — путем сдвига и вычитания.
- По мнению проф. Д. Э. Кнута, троичные информатика и математика должны быть более изящными, чем двоичные.

Ёмкость

Если считать стоимость ячейки равной $C_B = B$ фантиков (пропорционально количеству состояний), то для записи числа x потребуется:

$$C_x(B) = B \log_B x = B \frac{\log_{C_2} x}{\log_{C_2} B}$$

фантиков.

$$\frac{dC_x(B)}{dB} = \log_{C_2} x \left(\frac{\log_{C_2} B - 1}{\log^2_{C_2} B} \right) = 0$$

$$B = e!$$

самая ёмкая система — e-ичная.
Для целых оснований достаточно $\forall x \; C_x(3) < C_x(2)$.
К сожалению, электронные бинарные ячейки *существенно* дешевле тернарных и прочих.

Ёмкость

Если считать стоимость ячейки равной $C_B = B$ фантиков (пропорционально количеству состояний), то для записи числа x потребуется:

$$C_x(B) = B \log_B x = B \frac{\log_{C_2} x}{\log_{C_2} B}$$

фантиков.

$$\frac{dC_x(B)}{dB} = \log_{C_2} x \left(\frac{\log_{C_2} B - 1}{\log^2_{C_2} B} \right) = 0$$

$$B = e!$$

самая ёмкая система — e-ичная.

Для целых оснований достаточно $\forall x \; C_x(3) < C_x(2)$.

К сожалению, электронные бинарные ячейки *существенно* дешевле тернарных и прочих.

Ёмкость

Если считать стоимость ячейки равной $C_B = B$ фантиков (пропорционально количеству состояний), то для записи числа x потребуется:

$$C_x(B) = B \log_B x = B \frac{\log_{C_2} x}{\log_{C_2} B}$$

фантиков.

$$\frac{dC_x(B)}{dB} = \log_{C_2} x \left(\frac{\log_{C_2} B - 1}{\log^2_{C_2} B} \right) = 0$$

$$B = e!$$

самая ёмкая система — e-ичная.
Для целых оснований достаточно $\forall x \; C_x(3) < C_x(2)$.
К сожалению, электронные бинарные ячейки *существенно* дешевле тернарных и прочих.

Ёмкость

Если считать стоимость ячейки равной $C_B = B$ фантиков (пропорционально количеству состояний), то для записи числа x потребуется:

$$C_x(B) = B \log_B x = B \frac{\log_{C_2} x}{\log_{C_2} B}$$

фантиков.

$$\frac{dC_x(B)}{dB} = \log_{C_2} x \left(\frac{\log_{C_2} B - 1}{\log^2_{C_2} B} \right) = 0$$

$$B = e!$$

самая ёмкая система — e-ичная.
Для целых оснований достаточно $\forall x \; C_x(3) < C_x(2)$.
К сожалению, электронные бинарные ячейки *существенно* дешевле тернарных и прочих.

▸ Словарь Хаффмана

Упражнение

В приведённом генераторе словаря Хаффмана на приведённом прямо на сайте примере показать, что метод Хаффмана для тернарных деревьев требует модификации (а авторы генератора этим манкировали =P).

Упражнение

Модифицировать алгоритм Хаффмана для построения оптимального словаря.

Словарь Хаффмана

Упражнение

В приведённом генераторе словаря Хаффмана на приведённом прямо на сайте примере показать, что метод Хаффмана для тернарных деревьев требует модификации (а авторы генератора этим манкировали =P).

Упражнение

Модифицировать алгоритм Хаффмана для построения оптимального словаря.

Хэш-таблицы

По сути без изменений

Двоичные алгоритмы делят данные по принципу «$\leq, >$» или «$<, \geq$» на два примерно равных по мощности класса. Троичное деление по принципу «$<, =, >$» будет неравномерным. Нужны другие подходы или готовая информация о распределении данных.

Вопросы

EDU.DLUCIV.NAME