MAC0329 – Álgebra Booleana e Aplicações

IME/USP, junho/2021

Resumo do que foi coberto até agora

Computadores são máquinas que processam dados

Todos os dados no computador são representados em binário (vários bits 0 e 1)

Qualquer dado ou informação que queremos processar no computador precisa, portanto, ser codificado em binário

No início do semestre discutimos extensivamente a representação de inteiros na base 2. Em seguida discutimos formas de interpretar os binários codificados em um número finito e fixo de bits como números sem sinal ou como números com sinal

O processamento de dados no computador pode ser pensado como uma função do tipo

$$f : \{0,1\}^n \to \{0,1\}^m$$

Isto é, os dados a serem processados são representados em n bits e o resultado de seu processamento é representado em m bits.

Essas funções podem ser também pensadas como m funções binárias, isto é, $f = (f_1, f_2, \ldots, f_m)$ com

$$f_i : \{0,1\}^n \to \{0,1\}, \quad i = 1, 2, \ldots, n$$

No exemplo do somador de bits, temos

$$\begin{aligned} f\colon \{0,1\}^3 &\longrightarrow \{0,1\}^2 \\ (a_i, b_i, c_i) &\longmapsto (s_i, c_{i+1}) \end{aligned}$$

ou, então, $f = (f_1, f_2)$ com

$$s_i = f_1(a_i, b_i, c_i)$$

e

$$c_{i+1} = f_2(a_i, b_i, c_i)$$

Vimos em seguida o que é álgebra booleana e suas propriedades. Pode ser que vocês tenham pensado "O que isso tem a ver com somador de bits?"

Tendo uma álgebra booleana, conseguimos falar em expressões booleanas. Uma expressão booleana em n variáveis constrói-se compondo-se qualquer elemento da álgebra booleana (incluindo o 0 e o 1) e as variáveis $x_1, x_2, \ldots, x_n$, por meio das operações $+$, $\cdot$, $\bar{\ }$, e também compondo-se umas expressões com outras.

Quando atribui-se valores às variáveis $x_1, x_2, \ldots, x_n$ e calcula-se as operações presentes na expressão, obtém-se um elemento da mesma álgebra booleana. Quando calcula-se o valor da expressão para todas as possíveis atribuições de valores às variáveis $x_1, x_2, \ldots, x_n$, temos a tabela-verdade da função definida pela expressão.

Considerando a álgebra booleana B, temos então que uma expressão booleana em n variáveis define uma função $f : B^n \to B$

Podem existir várias expressões que definem uma mesma função $f : B^n \to B$ (basta pensar em expressões equivalentes)

Podemos mostrar que todas as funções binárias do tipo $f : \{0,1\}^n \to \{0,1\}$ (ou $f : B^n \to B$, com $B = \{0,1\}$) são booleanas, isto é, que ela pode ser definida por alguma expressão booleana.

Aqui vale a pena relacionar o processamento de dados que comentamos acima com a álgebra booleana. Ou seja, lembre que o processamento de dados no computador pode ser pensado por meio de funções $f : B^n \to B$ e estas por sua vez são funções booleanas (isto é, podem ser definidas por expressões booleanas). Assim, a álgebra booleana provê os fundamentos que nos permite manipularmos as expressões booleanas (por exemplo, simplificar as expressões ou verificar se duas expressões são equivalentes)

E escrever uma função $f : B^n \to B$ por meio de expressões booleanas é do interesse pois uma vez que as expressões booleanas envolvem apenas as operações $+$, $\cdot$, $\bar{}$, isso significa que podemos implementar qualquer função $f : B^n \to B$ usando as portas lógicas OU, E e NÃO

Outro resultado importante que vimos é que o conjunto das funções booleanas $f : B^n \to B$ (são 2^{2^n} funções dessas) é uma álgebra booleana (que denotamos $B(n)$)

As expressões produto canônico (são 2^n deles) são os átomos de $B(n)$

Baseado no teorema que afirma que todo elemento de uma álgebra booleana pode ser escrito, de forma única, como uma soma de um subconjunto de átomos, podemos afirmar que qualquer função booleana pode ser escrita como soma de produtos canônicos.

Dada a tabela-verdade de uma função $f : B^n \to B$, vimos também como escrever a expressão soma de produtos canônicos diretamente a partir da tabela.

O resumo acima corresponde (mais ou menos) a o que vimos antes de começarmos a falar sobre minimização de funções booleanas. Como já dissemos acima, uma função $f : B^n \to B$ pode ser escrita por diferentes expressões booleanas. O problema de minimização de funções booleanas consiste em encontrar, dentre todas as expressões que definem f, aquela que é "menor" (envolve menos operações $+$, $\cdot$, $\bar{}$) – pois uma expressão "pequena" implica um circuito também pequeno.

Em particular, vamos estudar minimização dois-níveis (isto é, expressões na forma soma de produtos ou produto de somas)