Digital Logic Systems

Recitation 5: Propositional Logic & Asymptotics

Guy Even   Moti Medina

School of Electrical Engineering Tel-Aviv Univ.

November 29, 2011

The De Morgan dual

Algorithm 1 DM(φ) - An algorithm for evaluating the De Morgan dual of a Boolean formula $\varphi \in \mathcal{BF}(\{X_1, \ldots, X_n\}, \{\neg, \text{OR}, \text{AND}\})$.

1. Base Cases: (parse tree of size 1 or 2)
   1. If $\varphi = 0$, then return 1.
   2. If $\varphi = 1$, then return 0.
   3. If $\varphi = X_i$, then return $(\neg X_i)$.
   4. If $\varphi = (\neg 0)$, then return 0.
   5. If $\varphi = (\neg 1)$, then return 1.
   6. If $\varphi = (\neg X_i)$, then return X_i.
2. Reduction Rules: (parse tree of size at least 3)
   1. If $\varphi = (\neg \varphi_1)$, then return $(\neg \mathrm{DM}(\varphi_1))$.
   2. If $\varphi = (\varphi_1 \cdot \varphi_2)$, then return $(\mathrm{DM}(\varphi_1) + \mathrm{DM}(\varphi_2))$.
   3. If $\varphi = (\varphi_1 + \varphi_2)$, then return $(\mathrm{DM}(\varphi_1) \cdot \mathrm{DM}(\varphi_2))$.

Example

$\mathrm{DM}(\neg(X + Y)) = ?$

The De Morgan dual (cont.)

Theorem

For every Boolean formula φ, $DM(\varphi)$ is logically equivalent to $(\neg\varphi)$.

Revisiting De Morgan's Law for Sets

Recall that the De-Morgan's law states that for sets A, B,

$$U \setminus (A \cup B) = \bar{A} \cap \bar{B}\,.$$

We depicted this law using Venn diagrams. We now prove this law, formally, using propositional logic.

Proof.

On the whiteboard. $\square$

Negation Normal Form

A formula is in negation normal form if negation is applied only directly to variables.

Definition

A Boolean formula $\varphi \in \mathcal{BF}(\{X_1, \ldots, X_n\}, \{\neg, \text{OR}, \text{AND}\})$ is in *negation normal form* if the parse tree (G, π) of φ satisfies the following condition. If a vertex in G is labeled by negation (i.e., $\pi(v) = \neg$), then v is a parent of a leaf labeled by a variable.

Example

- The formula $(\neg X) \cdot (\neg Y)$ is in negation normal form.
- The formulas $(\neg 0)$, $\neg(A \cdot B)$, $\text{NOT}(\text{NOT}(X))$ are not in negation normal form.

Theorem

Let $\varphi \in \mathcal{BF}(\{X_1, \ldots, X_n\}, \{\neg, \text{OR}, \text{AND}\})$. Then, $NNF(\varphi)$ is logically equivalent to φ and in negation normal form.

Negation Normal Form (cont.)

Algorithm 2 NNF(φ) - An algorithm for computing the negation normal form of a Boolean formula $\varphi \in \mathcal{BF}(\{X_1, \ldots, X_n\}, \{\neg, \text{OR}, \text{AND}\})$.

1. Base Cases: (parse tree of size 1 or 2)
   1. If $\varphi \in \{0, 1, X_i, (\neg X_i)\}$, then return φ.
   2. If $\varphi = (\neg 0)$, then return 1.
   3. If $\varphi = (\neg 1)$, then return 0.
2. Reduction Rules: (parse tree of size at least 3)
   1. If $\varphi = (\neg \varphi_1)$, then return DM(NNF($\varphi_1$)).
   2. If $\varphi = (\varphi_1 \cdot \varphi_2)$, then return (NNF($\varphi_1$) $\cdot$ NNF(φ_2)).
   3. If $\varphi = (\varphi_1 + \varphi_2)$, then return (NNF($\varphi_1$) $+$ NNF(φ_2)).

Example

- $NNF(\neg\neg X) = ?$
- $NNF(\neg\neg\neg X) = ?$.

Order of Growth: Reminder

Definition (7)

Let $f, g : \mathbb{N} \to \mathbb{R}^{\geq}$ denote two functions.

1. We say that $g(n) = O(f(n))$, if there exist constants $c_1, c_2 \in \mathbb{R}^{\geq}$ such that, for every $n \in \mathbb{N}$,
$$g(n) \leq c_1 \cdot f(n) + c_2.$$
2. We say that $g(n) = \Omega(f(n))$, if there exist constants $c_3 \in \mathbb{R}^{>}$, $c_4 \in \mathbb{R}^{\geq}$ such that, for every $n \in \mathbb{N}$,
$$g(n) \geq c_3 \cdot f(n) + c_4.$$
3. We say that $g(n) = \Theta(f(n))$, if $g(n) = O(f(n))$ and $g(n) = \Omega(f(n))$.

Order of Growth: Alternative Definition

Let us consider the following alternative definition of order of growth.

Definition (8)

Let $f, g : \mathbb{N} \to \mathbb{R}^{\geq}$ denote two functions.

1. We say that $g(n) = O(f(n))$, if there exist constants $c \in \mathbb{R}^{\geq}$ and $N \in \mathbb{N}$, such that, for every $n \in \mathbb{N}$,
$$\forall n > N : g(n) \leq c \cdot f(n) .$$
2. We say that $g(n) = \Omega(f(n))$, if there exist constants $d \in \mathbb{R}^{>}$ and $N \in \mathbb{N}$, such that, for every $n \in \mathbb{N}$,
$$\forall n > N : g(n) \geq d \cdot f(n) .$$
3. We say that $g(n) = \Theta(f(n))$, if $g(n) = O(f(n))$ and $g(n) = \Omega(f(n))$.

Lemma

Definitions 7 and 8 are equivalent if $f(n) \geq 1$ *and* $g(n) \geq 1$*, for every* n.

Reminder: Is it enough to solve for powers of 2?

In the following lemma we show that, under reasonable conditions, it suffices to consider powers of two when bounding the rate of growth.

Lemma (10)

Assume that:

1. *The functions $f(n)$ and $g(n)$ are both monotonically nondecreasing.*
2. *The constant ρ satisfies, for every $k \in \mathbb{N}$,*

$$\rho \geq \frac{g(2^{k+1})}{g(2^k)}.$$

If $f(2^k) = O(g(2^k))$, then $f(n) = O(g(n))$.

Revisiting: Is it enough to solve for powers of 2? Yes!

An analogous lemma that states that $f(n) = \Omega(g(n))$ can be proved if $\frac{g(2^{k+1})}{g(2^k)} \geq \rho$, for a constant ρ The lemma is as follows.

Lemma

Assume that:

1. *The functions $f(n)$ and $g(n)$ are both monotonically nondecreasing.*
2. *The constant ρ satisfies, for every $k \in \mathbb{N}$,*

$$\rho \leq \frac{g(2^{k+1})}{g(2^k)}.$$

If $f(2^k) = \Omega(g(2^k))$, then $f(n) = \Omega(g(n))$.

Example - 1

Let $g(n) \triangleq \log_3 n$. We claim that $g(n) = \Theta(\log_2 n)$

Proof.

Recall that for every $a, b, c \in \mathbb{R}$, $a, c \neq 1$,

$$\log_a b = \frac{\log_c b}{\log_c a} \,. \tag{1}$$

Hence, $\log_3 n = \frac{\log_2 n}{\log_2 3}$. Since, $3/2 < \log_2 3 < 8/5$ is a constant, then $c_1 = 2/3, c_2 = 0, c_3 = 5/8, c_4 = 0$ satisfy the conditions in Definition 7. □

Hence, when considering the order of growth of log functions with a constant base, that is $\log_c n$ and $\log_d n$ where c, d are constants, we may omit the base and simply refer the order of growth of these functions as $O(\log n)$, $\Omega(\log n)$ and $\Theta(\log n)$.

Example - 2

Let $g(n) \triangleq n^{\log_2 c}$. We claim that $g(n) = \Theta(c^{\log_2 n})$.

Proof.

We prove the following stronger claim.

$$n^{\log_2 c} = c^{\log_2 n} \,. \tag{2}$$

That will conclude the proof, since for every two functions $f, g : \mathbb{N} \to \mathbb{R}^{\geq}$, if $f = g$ then $f(n) = \Theta(g(n))$ and $g(n) = \Theta(f(n))$ (look for it in the text book!).
Let us apply the $\log_2$ function on the left-hand side and the right-hand side of Eq. 2. We get

$$\begin{aligned} \log_2(n^{\log_2 c}) &\stackrel{?}{=} \log_2(c^{\log_2 n}) \Leftrightarrow \\ \log_2 c \cdot \log_2 n &= \log_2 n \cdot \log_2 c \,, \end{aligned} \tag{3}$$

where the second line follows from the fact that $\log(a^b) = b \cdot \log(a)$. Since Eq. 3 holds with equality, and since the log function is one-to-one, then their arguments are equal as well,

Example - 3: Recurrence 3.

Consider the recurrence

$$f(n) \triangleq \begin{cases} 1 & \text{if } n = 1 \\ n + 3 \cdot f(\lfloor \frac{n}{2} \rfloor) & \text{if } n > 1. \end{cases} \qquad (4)$$

Lemma

The rate of growth of the function $f(n)$ defined in Eq. 4 is $\Theta(n^{\log_2 3})$.

Proof.

On the whiteboard. $\square$

Example - 4

Consider the recurrence

$$f(n) \triangleq \begin{cases} c & \text{if } n = 1 \\ a \cdot n + b + f\left(\left\lfloor \frac{n}{2} \right\rfloor\right) & \text{if } n > 1, \end{cases} \tag{5}$$

where a, b, c are constants.

Lemma

The rate of growth of the function $f(n)$ is $\Theta(n)$.

proof: $f(2^k) = 2a \cdot 2^k + b \cdot k + c - 2a...$

Example -5

Consider the recurrence

$$f(n) \triangleq \begin{cases} c & \text{if } n = 1 \\ a \cdot n + b + 2 \cdot f\left(\left\lfloor \frac{n}{2} \right\rfloor\right) & \text{if } n > 1, \end{cases} \tag{6}$$

where $a, b, c = O(1)$.

Lemma

The rate of growth of the function $f(n)$ is $\Theta(n \log n)$.

proof: We claim that $f(2^k) = a \cdot k2^k + (b + c) \cdot 2^k - b$....

Examples with floor and ceiling

1. $$f(n) \triangleq \begin{cases} 1 & \text{if } n = 1 \\ 1 + f\left(\left\lfloor \frac{n}{2} \right\rfloor\right) & \text{if } n > 1, \end{cases}$$

2. $$f(n) \triangleq \begin{cases} 1 & \text{if } n = 1 \\ n + f\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + f\left(\left\lceil \frac{n}{2} \right\rceil\right) & \text{if } n > 1, \end{cases}$$