Seven Rules for Big-$O$ and $\Theta^*$

Here are seven rules that you can use to solve problems involving big-$O$ and $\Theta$. They will solve the big majority of the big-$O$ and $\Theta$ comparisons you’ll need in this course (and for a long way beyond). Two assumptions are noted in the Fine Print on the back.

\[ \Theta(c \cdot f(x)) = \Theta(f(x)) \] (1)

\[ \Theta(f(x) + g(x)) = \Theta(\max(f(x), g(x))) \] (2)

\[ \Theta(f(x) \cdot h(x)) \leq \Theta(g(x) \cdot h(x)) \quad \text{if and only if} \quad \Theta(f(x)) \leq \Theta(g(x)) \] (3)

\[ \Theta(x^c) \leq \Theta(x^d) \quad \text{if and only if} \quad c \leq d \] (4)

\[ \Theta(\log x) < \Theta(x^c) \quad \text{if and only if} \quad 0 < c \] (5)

Assuming that $c > 0$,

\[ \Theta(x^c) < \Theta(d^c) \quad \text{if and only if} \quad 1 < d \] (6)

Assuming that $1 \leq c$ and $1 \leq d$,

\[ \Theta(c^c) < \Theta(d^c) \quad \text{if and only if} \quad c < d \] (7)

*Joshua Guttman, FL 137, \texttt{mailto:guttman@wpi.edu} with [cs2223] in Subject: field.
**Fine Print.** \( \Theta(f) \) means the set of all functions \( g \) that grow essentially as fast as \( f \). Officially, \( \Theta(f) = \{ g : \text{there exist } N_0, c_1, c_2 \text{ such that, for all } x > N_0, \quad g(x) \leq c_1 \cdot f(x) \text{ and } f(x) \leq c_2 \cdot g(x) \} \).

So \( \Theta(f) = \Theta(g) \), \( g \in \Theta(f) \), and \( f \in \Theta(g) \) all mean the same thing.

Big-\( O \) makes an ordering on the \( \Theta \)-classes. By \( \Theta(f) \leq \Theta(g) \), we mean that \( f \in O(g) \). In fact, when \( f \in O(g) \), either \( f \in \Theta(g) \), or else every function \( g' \in \Theta(g) \) asymptotically dominates every function \( f' \in \Theta(f) \). So this ordering works in a compatible way across whole \( \Theta \)-classes.

\( \Theta(f) < \Theta(g) \) means \( f \in O(g) \) but \( f \not\in \Theta(g) \).

A function \( f \) is non-decreasing if \( x \leq y \) implies \( f(x) \leq f(y) \). It’s eventually non-decreasing if \( N_0 < x \leq y \) implies \( f(x) \leq f(y) \), for some \( N_0 \). A function is eventually positive if, for some \( N_0 \), for all \( x > N_0 \), \( f(x) > 0 \).

In the rules above, assume all the functions \( f, g \) are:

- eventually non-decreasing, and
- eventually positive.