Asymptotic Properties

This document enumerates several of the asymptotic properties we’ve seen in class this semester.

**Relational Properties.** We saw analogies between asymptotic comparisons between two functions \( f, g \) and the comparisons between two reals \( x, y \). The analogies are as follows:

- \( f = O(g) \) is similar to \( x \leq y \)
- \( f = \Omega(g) \) is similar to \( x \geq y \)
- \( f = \Theta(g) \) is similar to \( x = y \)
- \( f = o(g) \) is similar to \( x < y \)
- \( f = \omega(g) \) is similar to \( x > y \)

We derived many properties that chained together asymptotic relationships by appealing to this metaphor. The ones we saw in class are listed below:

1. If \( f = O(g) \) and \( g = O(h) \) then \( f = O(h) \).
2. If \( f = \Omega(g) \) and \( g = \Omega(h) \) then \( f = \Omega(h) \).
3. If \( f = \Theta(g) \) and \( g = \Theta(h) \) then \( f = \Theta(h) \).
4. If \( f = o(g) \) and \( g = o(h) \) then \( f = o(h) \).
5. If \( f = \omega(g) \) and \( g = \omega(h) \) then \( f = \omega(h) \).

The facts above all demonstrate the *transitivity* of asymptotic notation. It’s also possible to derive transitive properties that mix different asymptotic relationships. Examples we saw in class include

6. If \( f = O(g) \) and \( g = o(h) \) then \( f = o(h) \).
7. If \( f = o(g) \) and \( g = O(h) \) then \( f = o(h) \).

It is easy to generate new properties in this way, by appealing to the analogy w/real numbers. e.g. we know that if \( x < y \) and \( y = z \), we must have \( x < z \). In the same way, we get:

8. If \( f = o(g) \) and \( g = \Theta(h) \) then \( f = o(h) \).

It’s important to understand that we don’t automatically get these properties just because they hold for real numbers. For example, one relational property that applies to reals but not asymptotic functions is *trichotomy*:

**Fact.** For every \( x, y \in \mathbb{R} \), either \( x < y \), \( x = y \), or \( x > y \).

It is not true that either \( f = o(g) \), \( f = \Theta(g) \), or \( f = \omega(g) \). In summary, while the analogy is intuitive and goes along way, always remember that we proved these properties using first principles.
Combining Functions

9. If \( f = O(h) \) and \( g = O(h) \) then \( f + g = O(h) \).

10. For any constant integer \( k > 0 \), \( f_1 + \cdots + f_k = O(k) \).

11. If \( f_1 = O(g_1) \) and \( f_2 = O(g_2) \) then \( f_1 + f_2 = O(g_1 + g_2) \).

12. **Product Rule.** If \( f_1 = O(g_1) \) and \( f_2 = O(g_2) \) then \( f_1 f_2 = O(g_1 g_2) \).

Log, Polynomial, Exponent Rules

Let \( f, g \) be functions such that \( f = O(g) \).

13. **Polynomial Rule.** For any constant integer \( k > 0 \), \( f^k = O(g^k) \).

14. **Log Rule.** If \( g = \omega(1) \) then \( \log(f(n)) = O(\log(g(n))) \).

15. **Exponent Rule.** If \( g(n) = f(n) + \omega(1) \), then \( 2^{f(n)} = O(2^{g(n)}) \).

Comparing Polynomials, Logarithmic, and Exponential Functions.

The most common functions we’ll see in this class (and in computer science in general) are polynomial, logarithmic, or exponential functions.

- A **polynomial** function is a function of the form e.g. \( f(n) = a_d n^d + a_{d-1} n^{d-1} + \cdots + a_1 n + a_0 \). The degree of this function is \( d \), i.e. the largest exponent of any term. In this class, we’ll always assume that the coefficient of the largest term is positive, e.g., \( a_d > 0 \).

- A **logarithmic** functions is of the form \( f(n) = \log_b(n) \).

- An **exponential** function is of the form \( f(n) = r^n \) for some \( r \).

In class, we derived properties for comparing different polynomial functions, different logarithmic functions, and different exponential functions:

16. If \( f, g \) are polynomials of the same degree then \( f = \Theta(g) \).

17. If \( f = n^a \) and \( g = n^b \) for constants \( b > a \) then \( f = o(g) \).

18. If \( f = \log_a(n) \) and \( g = \log_b(n) \) for constants \( a, b > 1 \) then \( f = \Theta(g) \).

19. If \( f = r^n \) and \( g = s^n \) for constants \( r > s > 1 \), then \( f = \omega(g) \).

Finally, we saw different rules for comparing polynomials vs logarithmic functions and polynomial vs exponential functions:

20. \( n^k = O(2^n) \) for any \( k > 0 \).

21. \( \log(n) = O(n^\epsilon) \) for any \( \epsilon > 0 \).
We also discussed that in fact, property 21 can be tightened:

**Lemma 1.** *For any* $\epsilon > 0$, $\log(n) = o(n^\epsilon)$.

This development in class was impromptu and not polished. I promised a \LaTeX’d proof, which lies below.

*Proof.* Fix any $\epsilon > 0$, and define $\epsilon' := \epsilon/2$. Then, $\epsilon' > 0$ as well, so we know $\log n = O(n^{\epsilon'})$. By the product rule, we get $(\log(n)^2) = O((n^{\epsilon'})^2) = O(n^\epsilon)$. From property 17 (setting $N := \log(n)$), we get $\log(n) = o((\log n)^2)$. So, we have

$$\log(n) = o((\log(n))^2), \text{ and } (\log(n))^2 = O(n^\epsilon).$$

By property 7, we get $\log(n) = o(n^\epsilon)$. \hfill \square