CSC 165

bounds and problems

week 9, lecture 2

Danny Heap

heap@cs.toronto.edu

www.cdf.toronto.edu/~heap/165/F09

A2: — use CDF submission facility early to work out glitches

— Use any application where you can type input + produce PDF

Proposed tutorials:

something from A2

1

Assignment 2 wants you to consider the negation of:

(a) $\exists x \in \mathbb{Q}^{\geq 0}, \exists \epsilon \in \mathbb{Q}^{+}, \forall \delta \in \mathbb{Q}^{+}, \exists y \in \mathbb{Q}^{\geq 0}, |x - y| < \delta \wedge |2x - 2y| \geq \epsilon$

$\forall x \in \mathbb{Q}^{\geq 0}, \forall \varepsilon \in \mathbb{Q}, \exists \delta \in \mathbb{Q}^{+}, \forall y \in \mathbb{Q}^{\geq 0}, |x-y| < \delta \Rightarrow |2x-2y| < \varepsilon$

$$\left[\ |x-y| \geq \delta \vee |2x-2y| < \varepsilon\ \right]$$

There are a couple of ways to negate the conjunction. The one we have a straight-forward proof technique for is implication.

bounded below

recall

Notice that the definition of big-Omega differs in just one character from big-Oh:

$\Omega(g) = \{f : \mathbb{N} \mapsto \mathbb{R}^{\geq 0} \mid \exists c \in \mathbb{R}^{+}, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow f(n) \geq cg(n)\}$

The rôle of B is, as with big-Oh, to act as a breakpoint, so comparisons don't have to start at the origin

The rôle of c is to scale g *down* below f.

If you're proving $f \in \Omega(g)$, you get to choose c and B to suit your proof.

one last bound

It often happens that functions are bounded above *and* below by the same function. In other words, $f \in \mathcal{O}(G) \wedge f \in \Omega(g)$. We combine these two concepts into $f \in \Theta(g)$.

tight bound.

$$\Theta(g) = \{f : \mathbb{N} \mapsto \mathbb{R}^{\geq 0} \mid \exists c_1 \in \mathbb{R}^+, \exists c_2 \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow c_1 g(n) \leq f(n) \leq c_2 g(n)\}$$

You might want to draw pictures, and conjecture about values of c_1, c_2, B for $f = 5n^2 + 15$ and $g = n^2$.

some theorems

$\mathcal{F} = \{f : \mathbb{N} \to \mathbb{R}^{\geq 0}\}$

How about: $g \in \Omega(f) \Leftrightarrow f \in \mathcal{O}(g)$

$\forall f, g \in \mathcal{F},\ g \in \Omega(f) \Rightarrow f \in \mathcal{O}(g)$

Assume $f, g \in \mathcal{F}$ # generic

Assume $g \in \Omega(f)$ # antecedent

Then $\exists c \in \mathbb{R}^+, B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow f(n) \geq c\,g(n)$

from defn of antecedent

Choose $c_1 \in \mathbb{R}^+, B_1 \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B_1 \Rightarrow \underline{g(n) \geq c_1 f(n)}$

Let $c_2 = \underline{\ 1/c_1\ }$. Then $c_2 \in \mathbb{R}^+$

Set $B_2 = \underline{\ B_1\ }$. Then $B_2 \in \mathbb{N}$.

Assume $n \in \mathbb{N}$. ~~assu~~

Assume $n \geq B_2$

Then $g(n) \geq c_1 f(n)$ # from antecedent # $B_2 = B_1$

Thus $c_2 g(n) \geq c_2 c_1 f(n) = f(n)$ # $c_2 = \frac{1}{c_1}$

Then $n \geq B_2 \Rightarrow$

scratch

problem solving

Often problem-solving techniques aren't taught in mathematically-oriented courses. Rather than expect you to have picked them up on your, we offer some approaches that *sometimes* work.

We start from the approach elaborated by George Polya in *How to Solve it*

UNDERSTAND THE PROBLEM: What's given (input), what's required (output)? Can you represent either input or output in ways that seem clearer — possibly symbols or diagrams? Make sure you can state the problem clearly in your own words.

HAVE PLAN(S): *Before* actually diving into an attempt to solve the problem, try to articulate one or more plans to solve it. Perhaps the approach (but not the result) of other problems will work. Perhaps you can work backwards: assume you've solved the problem and think about what the step just *before* solving it must have been.

TRY OUT A PLAN: If it works, try to verify it. If it doesn't work, think about where you got stuck (even write down what blocked you). Did you have a plan B in the previous step?

LOOK BACK: If you get some results, look back on what worked and what didn't. Can you extend or generalize the problem into a new problem?

today's problem

You can get a handout to work on in groups. Here's the gist of the problem:

Suppose you're given the list:

37	93	0	23	79	65	49	81	67	8	32	29	96	76	15	9	51	14	29	69

Can you find one or more longest non-decreasing sequences?

For example, 37, 93, 96 is a sequence that's non-decreasing,

but you can easily find longer ones.

Sequences are ordered, but need not be contiguous