Niklas Königsson

Dv15nkn

Obligatorisk uppgift 4

Dv3: Beräkningar och språk

1

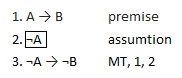
**2) The inclusion problem for context-free grammars is the following. Given two CFG:s G and H, decide whether L(G) ⊆ L(H), i.e., whether the language of G is included in the language of H. The universality problem for context-free grammars is the following. Given a context-free grammar G over terminal alphabet Σ, decide whether L(G) = Σ∗ , i.e., whether G can generate every string over Σ. Describe a reduction from the universality problem for context-free grammars to the inclusion problem for context-free grammars.**

**3) Assume that A ≤m B, i.e., that there is a reduction from A to B. Prove that ¬A ≤m ¬B, where ¬A and ¬B are the compliments of A and B, respectively.**

To prove this I use natural deduction. The pretext “A reduces to B” can, in this context, be seen as A implies B. So we get the statement:

A → B ⊢ ¬A → ¬B

With the Modus Tollens rule we get:



Thus proving the statement.

(This was a joined reasoning with Samuel Bylund Felixon after the 4/11 lesson about natural deduction, wich will explain any resemblance in our answers).

**4. Formally prove each of the following statements**:

**a) 5n4 + 3n2 + 4 = O(n4)**

First I find the constant C when n -> ∞ by using the formula: (f(n) / O(f(n))) + 1.

We get ((5n4 + 3n2 + 4)/ n4) + 1 -> ((5\*100004 + 3\*100002 + 4)/ 100004) + 1 = 6

By testing with n of increasing size we find that C does not take a higher value than 6.

By the definition of big-O: *f(n) <= C\*O(f(n) for all n >= n0,* so we can test this by inserting values for n and at the same time find n0.

F(n) and C \* O(f(n)) meet at n = 2 wich is our n0

And by testing larger n-values we can se that after this point, 5n4 + 3n2 + 4 will never grow faster than n4 wich proves that 5n4 + 3n2 + 4 = O(n4).

**b) 2n = O(n!)**

In this function we have something that grows in the following manner: 2 \* 2 \* 2 …n number of times and a bigger n value will add that many 2’s.

The factorial n will add another n if increased.

This means that after an n-value of 2 the factorial function will grow faster than 2n thus proving that

2n <= C \* O(n!) for all n > 2. Where the constant C is at least 2.

**c) 22n != O(2n)**

In this function we have something that grows in the following manner: 2 \* (2)2 \* (2)2 …n number of times and a bigger n value will add that many (2)2’s.

The O-notation grows with n number of single 2’s. This clearly shows that 22n always grow twice as fast as 2n with a positive n. And this is true for any positive constant C.

This proves that22n != O(2n)

**d) For any real constant a we have (n + a)3 = O(n3).**

Instictively we can say that only the exponent change a graphs acceleration. And since the formulas exponent number is the same in both (n + a)3 and n3 we know that these graphs accelerate at the same rate after a certain n-value. There will always be a certain constant C to make f(x) <= C \* O(f(x)). This constant will perhaps be of a higher value if a is incremented but there will always be a C to make this true for any a.