BİL 133 Combinatorics and Graph Theory

HOMEWORK 4 (25 Points)

Due Date: June 21, 2018

1 [4 POINTS] FREE/BOUND VARIABLES AND SUBSTITUTION OF TERMS FOR VARIABLES

Let ϕ be $\exists x(P(y,z) \land (\forall y(\neg Q(y,x) \lor P(y,z))))$, where P and Q are predicates with two arguments.

- [$0.5 Points$] Draw the parse tree of ϕ.
- [$0.5 Points$] Identify those variable leaves which occur free and those which occur bound in ϕ.
- [$0.5 Points$] Is there a variable in ϕ which has free and bound occurrences? Explain.
- [$1.5 Points$] Consider the terms w (w is a variable), $f(x)$ and $g(y,z)$, where f and g are function symbols with one, respectively two, arguments.

  [$0.5 Points$] Compute $\phi[w/x]$, $\phi[w/y]$, $\phi[f(x)/y]$, and $\phi[g(y,z)/z]$.

  [$0.5 Points$] Which of w, $f(x)$ and $g(y,z)$ are free for x in ϕ?

  [$0.5 Points$] Which of w, $f(x)$ and $g(y,z)$ are free for y in ϕ?
- [$0.5 Points$] What is the scope of $\exists x$ in ϕ?
- [$0.5 Points$] Suppose that we change ϕ to $\exists x(P(y,z) \land (\forall x(\neg Q(x,x) \lor P(x,z))))$. What is the scope of $\exists x$ now?

2 [8 Points] Logic Proofs

Notice that the rules for $\forall$ are very similar to those for $\wedge$ and those for $\exists$ are just like those for $\vee$. Let's elaborate this statement in this problem.

- $[1 Point]$ Find a (propositional) proof for $\phi \to (q_1 \wedge q_2) \vdash (\phi \to q_1) \wedge (\phi \to q_2)$.
- $[1 Point]$ Find a (predicate) proof for $\phi \to \forall x Q(x) \vdash \forall x(\phi \to Q(x))$, provided that x is not free in ϕ. (Hint: Whenever you used $\wedge$ rules in the propositional proof of the first item, use $\forall$ rules for this item).
- $[1 Point]$ Find a proof for $\forall x(P(x) \to Q(x)) \vdash \forall x P(x) \to \forall x Q(x)$.
- $[1 Point]$ Prove $\forall x(P(x) \wedge Q(x)) \vdash \forall x P(x) \wedge \forall x Q(x)$.
- $[1 Point]$ Prove $\forall x P(x) \vee \forall x Q(x) \vdash \forall x(P(x) \vee Q(x))$.
- $[1 Point]$ Prove $\exists x(P(x) \wedge Q(x)) \vdash \exists x P(x) \wedge \exists x Q(x)$.
- $[1 Point]$ Prove $\exists x F(x) \vee \exists x G(x) \vdash \exists x(F(x) \vee G(x))$
- $[1 Point]$ Prove $\forall x \forall y(S(y) \to F(x)) \vdash \exists y S(y) \to \forall x F(x)$.

3 [4 Points] UpperBound for a Variation of Tower of Hanoi

Consider the variation of the Tower of Hanoi Problem, where we have 4 pegs instead of 3. Let $T(n)$ denote the minimum number of moves sufficient to carry n disks from one peg to another while obeying Lucas' rules when there are only 3 pegs as we defined in class. Now, let $S(n)$ denote the minimum number of moves sufficient to carry n disks from one peg to another while obeying Lucas' rules when there are 4 pegs.

First, determine the values of $S(0), S(1)$ and $S(2)$.

Then, show that $S(n) \le 2S(k) + 2T(n-k-1) + 1$ for any $k \in \{1, 2, \ldots, n-1\}$.

4 [4 Points] Induction Proof for Geometry

You learned in high school that the sum of the internal angles in an n-sided polygon (where $n \ge 3$) is $(n-2)\pi$. In that problem, you are asked to prove that fact by using mathematical induction on the number of sides of polygons.
You can take the fact that the sum of the internal angles of any triangle is π as a premise.

5 [5 POINTS] ANOTHER VARIATION OF TOWER OF HANOI

Consider the variation of the Tower of Hanoi problem, where we have $2n$ disks of n different sizes, two of each size. As in the classical version, there are 3 pegs, and all the $2n$ disks are initially stacked in one of the pegs. The goal is to transfer the pile of the disks to another peg. As usual, only one disk can be moved at a time, and a disk cannot be placed onto a smaller one. Notice that a disk can be placed on the other identical disk (since otherwise the initial configuration would be problematic.)
How many moves are necessary and sufficient to transfer the entire tower from one peg to another?

- Look at small cases first and try to see a pattern.
- $[1 Point]$ Define a function $T(n)$ that corresponds to the quantity of interest. (Make sure that your definition is unambiguous.)
- $[1 Point]$ Find an algorithm to solve the problem. Use the performance of your algorithm to obtain a recurrence **inequality** for $T(n)$.
- $[1 Point]$ Prove that the upperbound you proved in the previous step is tight, by proving a matching lowerbound for $T(n)$. In order to do that the algorithm you found in the previous step has to be the optimal one!
- $[1 Point]$ By combining your matching upper and lower bounds, you can obtain a recurrence equality that defines $T(n)$. Please state your recurrence equality.
- $[1 Point]$ Solve your recurrence.