MAD 3104

Chapter One stuff:

This is <u>not</u> a topic in formal logic. The formal rules are not important here. What we want is the ability to recognize and make valid proofs.

~~This~~ You can spend the rest of your life trying to get our students to understand this stuff. Face the fact that you will have to review this time and time again. Limit yourself to at <u>most</u> 6 lectures the first time through. A 3 lecture pace is given below.

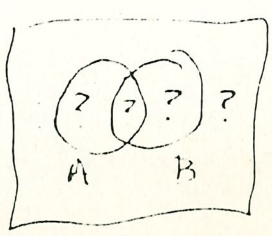
<u>Lecture 1:</u> Review Induction and do recursion §1.11 (Euclidean Algorithm) Assign 1.5, 1.6, 1.8, 1. to be read

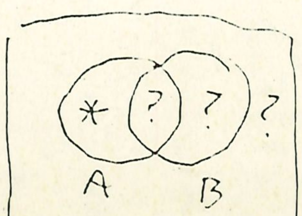
<u>Lecture 2:</u> Changing senteces in logical symbols and venn diagrams (1.6, 1.9). Intro to quantifiers (1.9) Negation, converse and contrapositives (1.9)

<u>Lecture 3:</u> Valid & Invalid inferences (1.10, 1.7) Pigeonhole as an example of proof by contradiction.

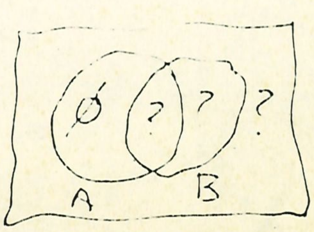
Comments: Bellenot has some stuff written up (rough) for this.

Venn diagrams





Some A is not B



ALL A is B

Additional Exercises for Section 1.2

1. ~~Determine which~~

For each of the properties Trans. Ref. Irref. Sym. Anti-Sym. Asym and each of the relations below either ~~show~~ state the relation has this property or give a counter example to show it does not.

(a) the relation $=$ on the set of integers $\{0, \pm 1, \pm 2 \ldots\}$

(b) the relation $<$ " " " " "

(c) the relation $\leq$ " " " " "

(d) the relation R on the set of integers when

(i) $xRy \iff |x-y| \leq 3$

(ii) $xRy \iff$ remainder of $\frac{y}{x}$ is zero or one

(iii) $xRy \iff xy \geq 100$

(iv) $xRy \iff x-y = 2$

(v) $xRy \iff x \leq y$ and $y \leq x+10$

(vi) $xRy \iff x \leq y$ or $y \leq x+10$

(vii) $xRy \iff x-y$ is even

(viii) $xRy \iff x = 2$

(ix) $xRy \iff |xy| \geq 100$

(x) $xRy \iff x$ can be written in English using strictly less than the number of letters it takes to write y in English.

Breadth First search: BFS

BFS is one way of constructing a spanning tree in a connected graph.

Input: Connected graph G with vertices (and implicit ordering of vertices) $v_1, v_2, \ldots, v_n$

Output: An orientated spanning tree T (i.e. a directed tree such that the children of any node have a position from left to right.)

Recursive Procedure BFS (x : some vertex)

Begin (Here is where you add visit x in BFS-order)

For $i = 1$ to n do

if v_i is not in T and v_i is ~~connected~~ adjacent to x

then add v_i, (x, v_i) to T making v_i the next child of x to the right.

For $i = 1$ to n do

if v_i is a child of x then call BFS (v_i)

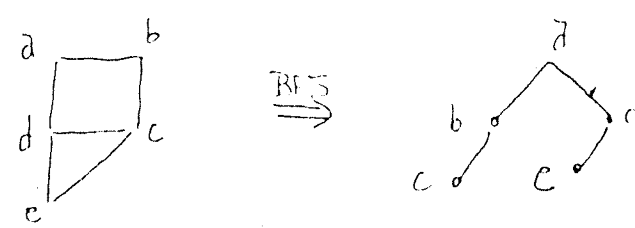
End

Intalization

Put v_1 in T

Call BFS (v_1)

Example:



Comments:

1. The remaing edges G−T can be classified as either cross edges (joining vertices x & y such that subtree(x) ∩ subtree(y) is empty (where subtree(x) is x and everything below)) or back edges (joining vertices x & y such that subtree(x) ⊂ subtree(y)

Comments:

1. If T is constructed by DFS then $G-T$ has no cross edges.

2. This fact can be used to find the articulation points of G (see Tucker §9.4)

3. The traversals in *7 p 442 are based on DFS Preorder ~~is~~ lists the vertices as they are first visited, postorder lists the vertices as they are last visited (i.e when you finally leave the procedure) and inorder catchs the middle time (thus it is only defined for binary trees)

Exercises

1. For a bunch of connected graphs find the spanning trees by BFS & DFS

2. Prove both comments which are *1.

3. For a bunch of trees, list the vertices in BFS order

4. For a bunch of trees, list the vertices in preorder & postorder

5. Do a BFS for all possible games of tic-tac-toe. I.E. the root is #, its 9 children have an x in exactly one square etc.

6. Evalution of expressions: infix, postfix & the use of stacks.