CS1112: Foundations of Computer Science I

Exercises
Professor Barry O'Sullivan
(originally by Professor Ken Brown)
Dept of Computer Science, University College Cork
Office: 2-65, Western Gateway Building
Email: b.osullivan@cs.ucc.ie

This document contains the problems and exercises that we will look at in the problem solving classes, and that you will attempt in the weekly assignments. The names of the sheets correspond to the chapters of the course notes.

The fundamental skill you will develop during this module is the ability to express technical concepts clearly and precisely. It is a skill, and you will have to practise in order to get good at it. You will have to learn how to read statements written in formal notation, how to understand those statements, and how to write your own formal statements that express precisely what you want to say. This means you have to become familiar with the language and what its symbols mean. You cannot do this by just watching a lecturer run through some slides or write up some examples on the whiteboard. You can't do this by trying to read the course notes. You will only be able to develop the skill by *doing* it – doing it regularly and repeatedly, gradually trying more complex concepts once you have mastered the simpler ones. This document gives you the exercises you need each week. These exercises also include some definitions, problems and techniques that we will assume in the course.

Problems 0: Introductory Algorithms and Problem Solving

1. Write an algorithm which states how to make a cup of tea. Pass the algorithm to someone else, and ask them to find any ambiguities, missing instructions, and possibilities for misinterpretation.

2. The following algorithm finds the smallest number in a list of numbers (for example, finding the lowest price in a list of prices of items that matched a search). Read the algorithm, and then read the text below it. For the list [4 3 6 5 1 2], write down each change in the value for `n` and `m` as you step through the algorithm.

```
Input: A list of numbers
1. store in n the value 1
2. change m to be equal to the number in position n in list
3. while n is not the last position in the list
4.    update n by adding 1 to its value
5.    if the number in position n < m
6.       change m to be equal to the number in position n
7. return m
```

Note: n and m are *variables*. Think of them as the names of boxes on a form – each time you are told to "`store in n the value ...`" or "`let m be ...`" or "`change m to be ...`", work out the value of the expression, then rub out the previous value in the named box, and write in the value you have just computed. In future, instead of using English phrases like "let n be …" we use a symbol. Sometimes we use ←, or "`:=`", and sometimes we use just "=". So if you see, for example, "`x ← y+1`", it means "change x be equal to y+1", or "compute the value of y+1, and assign it to x".

Lines 5 and 6 are examples of a *conditional* statement. If the test in line 5succeeds, then do line 6; if the test in line 5 fails, skip line 6.

Note the while loop from line 3 to line 6. After line 5, you may have to do line 6. Either way, you then go back to line 3, and do the test there again. If that test succeeds, go to line 4; if it fails, go to line 7.

In line 7, "`return m`" means report the value of m to whoever asked you to run the algorithm.

3. An *algorithm* is an ordered, deterministic, terminating, executable set of instructions.
For each of the following, state why it is not an algorithm:

(i)
```
1. total ← 1
2. while total > 0
3.    x ← total + 1
4.    total ← x
```

(ii)
```
Input: a list of numbers
1. n ← first number in list
2. s ← n
3. while s < 20
4.    n ← next number in list
5.    s ← s+n
6. return s
```

(iii)
```
Input: a standard pack of 52 cards, and a number n of players
1. deal all cards so that each player has same number of cards
2. ask each player for their highest card
3. announce that the player with the highest card wins
```

4. The following is an attempt to write an algorithm for the same task as Q2. What is wrong with it?

```
Input: A list of numbers
m ← first number in list
while there are still numbers after m
    if m > next number after m
        m ← next number after m
return m
```

5. Write an algorithm which takes three separate pieces of input (h,m,s), representing a length of time in hours, minutes and seconds, and produces as output the same length of time in seconds.

6. In the following sum, each letter represents a different digit. Work out a possible solution by giving a unique digit to each letter such that the sum is correct. Describe how you solved the problem.

```
 ABC
+DBE
---------
 FCD
```

7. A farmer returning from market with a dog, a hen and a basket of grain arrives at a river. There is a small boat which can be used to cross the river (and it is on the farmer's side of the river), but the boat is so small it can only hold the farmer plus one other object at any one time. If the dog is left alone with the hen, it will eat it. If the hen is left alone with the grain, it will eat it. Write down a sequence of steps which will allow the farmer to get himself, the dog, the hen and the basket of grain across the river without losing any of it (and no, neither the farmer nor the dog nor the hen can swim).

8. Write an algorithm for playing noughts-and-crosses as the first player, so that, if the algorithm is followed correctly, the first player doesn't lose.

9. An online dictionary consists of many words stored in an array (an ordered sequence of known length), and the words are stored in alphabetical order. Given two words, you can compare them, to see which comes before the other in the alphabet. You know the length of the array, and you can read the word at any individual position. Write an algorithm which, given a word as input, finds out whether or not it is in the dictionary.
Try your algorithm on some random word lists. How many steps does it need? Could you do better?

10. Amazon.com and other sites provide recommendations based on your browsing history. Suppose each entry in the list of recommendations consists of a title, a recommendation level (a number from 0 to 100), and a price. When you are presented with the list, you should be able to sort it, to see the items ordered by price, or by recommendation, and so on. Write an algorithm which takes a list of any length, and sorts it so that it is ordered from the cheapest to the most expensive entry.
Try your algorithm on some random lists of recommendations. How many steps does it need? Could you do better?

Problems 1: Sets

1. State which of the following statements are true or false.

(i) $b \in \{a,b,c\}$
(ii) $9 \in \{2,3,5,7,11,13,17,19\}$
(iii) $12 \in \{1,2,3\}$
(iv) enda $\notin$ {bertie, brian, john, willie}
(v) "while" $\notin$ {"if", "while", "do", "break"}
(vi) $3 \in \{ x \mid x$ is an integer, and $0 < x < 10\}$
(vii) $5 \in \{z \mid z$ is an integer, and $z < 0\}$
(viii) brazil $\notin$ {y | y is a country in the European Union}
(ix) $\{2,5\} \subseteq \{1,2,3,4,5,6\}$
(x) $\{e,f,g\} \subseteq \{b,a,r,g,e\}$
(xi) {} $\subseteq$ {barry, derek, james, john, ken}
(xii) {bohemians, stpatricks, shamrockrovers, ucd} $\subseteq$ {stpatricks, shamrockrovers, ucd, bohemians}
(xiii) {asfdh, qwerty, mnbvc} $\subseteq$ {qwertyuiop, asfdh, mnbvc}
(xiv) $\{z \mid z$ is a prime number$\} \subseteq \{x \mid x$ is an integer and $0 < x\}$

2. Write out an extensional set definition for the following collections.

(i) the module codes of each module you are registered for this year
(ii) the names of the months of the year which do not have the letter "r" in them
(iii) $\{x \mid x$ is an integer, and $x > -5$ and $x < 5\}$

3. Write out intensional set definitions for the following sets.

(i) $\{1,3,5,7,9\}$
(ii) $\{21, 3, 15, 9, 3, 6, 15, 9, 12, 21, 18\}$

4. Which of the following statements are true?

(i) $\{\} \subseteq A$, for any set A
(ii) $0 \in \{ \}$
(iii) $0 \in ø$
(iv) $\{ \} \in A$, for every set A
(v) $\{\} \subseteq \{\}$
(vi) $\{0\} \subseteq \{\}$

5. Write algorithms which return *true* or *false* for the following tasks, using the command given below:

(i) Given as input an object *x* and a set *S*, return *true* if $x \in S$, and *false* if $x \notin S$
(ii) Given as input a set S and another set T, return *true* if $S \subseteq T$, and *false* if $S \not\subseteq T$

You can assume a command `getNextElement(S)` which reports the next element of a set S (i.e. the first element of S that has not been given to you before). So you can have a line in your algorithm which says, for example, `y ← getNextElement(S)`. If you had a set S = {1,2,3}, and you had the following lines in your algorithm

```
y ← getNextElement(S)
z ← getNextElement(S)
```

then *y* would have the value 1, and *z* would have the value 2. If S has no more elements, then `getNextElement(S)` returns the special value NULL.

6. Consider the 5 sets A, B, C, D and E, taken from the universal set U = {0,1,2,3,4,5,6,7,8,9}
A = {1,2,3,4,5,6}
B = {2,4,6,8}
C = {9,8,7,6,5}
D = {3,6,9}
E = {8,3,5,9,1}
Write down the results of the following operations. Draw Venn diagrams to check your answers.
(i) $A \cup B$
(ii) $C \cap D$
(iii) $C \setminus B$
(iv) $\{\} \cup E$
(v) $B \setminus D$
(vi) E’
(vii) $D \setminus B$
(viii) $D \cap (A \cup E)$

7. Write down the cardinality of the following sets
(i) {1, 2, 3, 4, 5, 6, 7, 8, 9}
(ii) {turnip, potato, parsnip}
(iii) {bush, bush, clinton, clinton, bush, reagan}
(iv) {}
(v) $\{1,2,3,4,5,6\} \cap \{9,8,7,6,5\}$
(vi) $\{1,3,5,7,9\} \cap \{2,4,6,8\}$
(vii) $\{x \mid x \text{ is an integer}, 0 < x < 20\} \setminus \{z \mid z \text{ is an integer and } 5 < z < 15\}$

8. Which of the following statements are true?
(i) $\{\} \subseteq A$, for any set A
(ii) $0 \in \{\ \}$
(iii) $0 \in \emptyset$
(iv) $\{\ \} \in A$, for every set A
(v) $\{\} \subseteq \{\}$
(vi) $\{0\} \subseteq \{\}$
(vii) $|\{\}| = 0$
(viii) $|\{0\}| = 0$
(ix) $|(A \setminus B)| = |A| - |B|$, for all possible pairs of sets A and B

9. (i) Write an algorithm which, when given a set S, returns |S|
(ii) Write an algorithm which, when given two sets S and T, returns the set $S \cap T$
You can assume a command `addToSet(S, x)` which adds the element x to S and returns the new set. So if S = {1,2,3}, then the line
`T ← addToSet(S,8)`
would end up with T = {1,2,3,8}.
You can also use `getNextElement(S)`, `inSet(x,S)` and `subSet(S,T)` from sheet 2.

10. For the student database

$U = \{alice, bob, carol, darragh, eileen, frank, oliver, patsy, ronan, susan, ted, una\}$

with registrations for new modules as follows:

Alice: CS200, CS201
Bob: CS201, EC204
Carol: EE101, FR105, MG201
Darragh: CS201
Eileen: EC204, FR105
Frank: CS201, MG201
Oliver: EC204, EE101
Patsy: MG201
Ronan: CS200
Susan: EE101, EC204
Ted: FR105, CS201, CS200
Una: MG201

Write down the sets of students registered for each module.

11. For the student database in Q10, write down the result of the following operations:

(i) $CS200 \cup CS201$
(ii) $MG201 \cap EE101$
(iii) $EC204 \cap CS200$
(iv) $CS200 \setminus CS201$
(v) $CS201 \setminus CS200$

12. Write down the following

(i) $P\{a,b,c,d\}$
(ii) $P\{apple\}$
(iii) $P\{\}$
(iv) $P(CS200)$, where CS200 is the set you created in Q10
(v) $|P\{a,b,c,d\}|$
(vi) $|P\{apple\}|$
(vii) $|P\{\}|$

13. State whether the following statements are true for every possible set A, B and C. If the statement is false, give an example of sets that demonstrate it is false.

(i) $A \cup B = B \cup A$
(ii) $A \setminus B = B \setminus A$
(iii) $(A \cup B) \cap C = (A \cap B) \cup C$
(iv) $(A \setminus B) \setminus C = A \setminus (B \setminus C)$
(v) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
(vi) $A \cup \{\} = A$
(vii) $A \cap \{\} = A$
(viii) $|(A \cap B)| = |(A \cup B)| - |A| - |B|$
(ix) $(A \cap B)' = A' \cup B'$

Problems 2: Functions

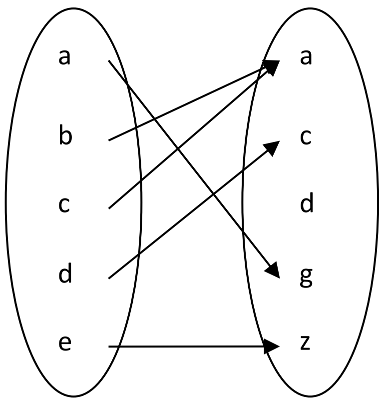
1. Sketch an arrow diagram for the following functions:

(i) A = {alice, carol, eileen, susan}, B = { bob, darragh, oliver, padraig, ronan, ted}
f:A→B = {(alice, ted), (carol, ted), (eileen, ronan), (susan, bob) }
representing the function "wants_to_be_with"

(ii) N = {-3,-2,-1,0,1,2,3}, S = {0,1,2,3,4,5,6,7,8,9},
sq:N→S is the function where the 2nd element is the square of the 1st

(iii) T = {toulouse, cork, galway, london, paris, wexford}, C = {Ireland, Britain, France},
g:T→C is the relation where the 2nd element is the country of the first

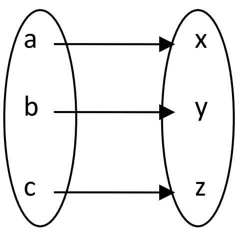
2. Write out the set notation to describe the following functions (i.e. specify two sets and the mapping between them as a set of ordered pairs:



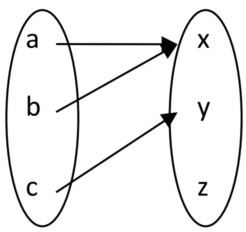
(i)

(ii)"is the lecturer for" between the set of modules you are taking, and a set of lecturers from UCC

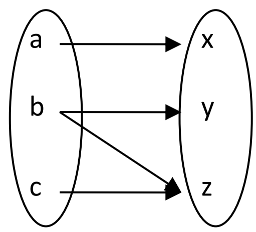
3. Which of the following are functions? For those that aren't, state why not.



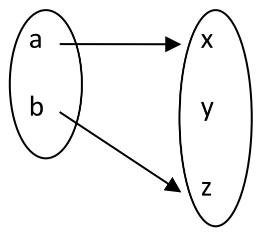
(i)



(ii)



(iii)



(iv)

(v) "Place of birth", relating passport holders to locations
(vi) the relationship linking student ID numbers to names in the student records database
(vii) the relationship "is registered for" linking student ID numbers to modules in UCC
(viii) the relationship "has price" linking books to prices in Waterstones
(ix) the relationship "is author of" linking people to books on the Amazon database
(x) the relationship "is a friend of" linking people to people on Facebook

4. State the domain and codomain of the following functions:

(i) $f : \mathsf{Z} \to \mathsf{Z} : x \mapsto x+1$

(ii) $g : \mathsf{Z} \to \mathsf{R} : x \mapsto x/2$

(iii) *manager* : {alice,bob,carol} → {alice, ted} : *manager*(*x*) = the line manager of employee *x*

5. State the range of the following functions:

(i) $f : \mathsf{Z} \to \mathsf{Z} : x \mapsto 2*x$

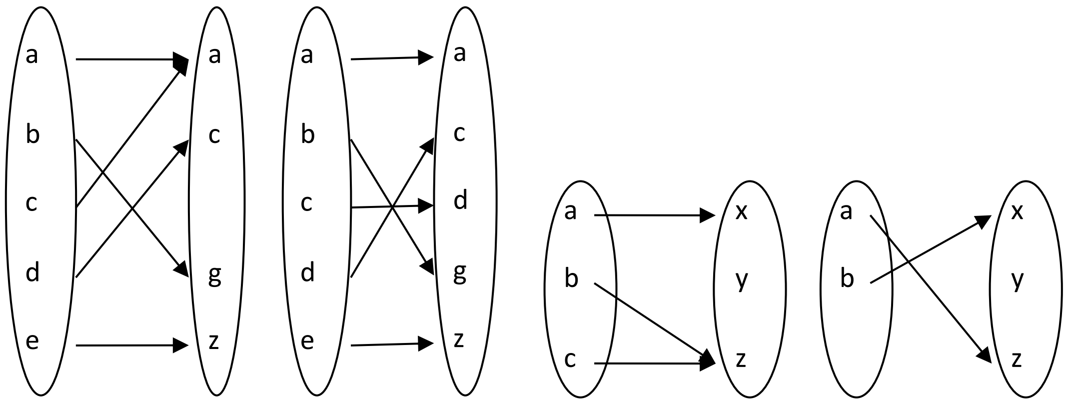
(ii) $h : \mathsf{N} \to \mathsf{N} : x \mapsto$ the remainder when x is divided by 3 [Note: N = {0,1,2,3,...}]

(iii) {(1,3), (6,8), (2,4), (3,5)}

6. Define a set of students at UCC you know, a set of degree programmes, a set of second level schools, a set of PPS numbers, and functions "is_registered_for", "attended_school" and "has PPS number". What do the following expressions mean, in English?

(i) domain(is_registered_for)
(ii) range(attend_school)
(iii) has_PPS_number $^{-1}$

7. State whether the following functions are bijective, injective, surjective, or just a plain function:



(i) (ii) (iii) (iv)

(v) "was first registered in year" between all students in the UCC records database and all calendar years
(vi) "has ID number" between all current students in theUCC records database and all ID numbers ever issued by UCC (Note: we don't re-use ID numbers)
(vii) "has ID number" between all current students in the UCC database and all ID numbers of current students in UCC
(viii) "has rank in sales figures" between all books on Amazon's database and all integers (Note: if two books have the same sales, they have equal rank)
(ix) f : all non-empty strings of characters $\rightarrow$ all non-empty strings of characters : any string s is mapped onto its reversed string [for example, "apple" would be mapped onto "elppa"]
(x) g: all non-empty strings of characters $\rightarrow$ all non-empty strings of characters : any string s is mapped onto a new string, which is s with every letter replaced by its corresponding capital [for example, "Ken_Brown_126" would be mapped to "KEN_BROWN_126"]

8. When data is transmitted across networks, in order to be able to check whether the data was received successfully without errors, we sometimes place a "check digit" at the end of the data sequence. For example, when we are transmitting a number $x=x_1x_2x_3...x_n$, we might add another digit at the end which is equal to the rightmost digit of $x_1+x_2+x_3+...+x_n$. When we receive the data, we strip off the last digit, and check to see if it is the digit we should have received; if it isn't, we know there has been an error in the transmission, and we can ask for the data to be sent again. For example, (i) if we are transmitting the number 423871, we add all the digits to get 4+2+3+8+7+1=25, and so we add "5" onto the end, and transmit 4238715; (ii) if we receive a transmission of 523146, we take the number 6 off the end, add the remaining digits 5+2+3+1+4 = 15, take the rightmost digit, which is 5, and so we know there has been an error in transmission, since 5 =/= 6.

(i) What is the check digit to be added to the end of 238613?
(ii) Is the procedure defined above a function from non-empty strings of digits to the set {0,1,2,...,9}? If so, is it bijective, injective, surjective, or just a plain function?
(iii) If we receive the transmission 427159, can we be sure that we have received it with no errors? Explain your answer by referring to your answer for (ii).

9. Specify the inverse functions for the following functions

(i) f: X→ Y, where f is given by {(1,2), (2,4), (3,6),(4,8)}, X = {1,2,3,4}, Y = {2,4,6,8}

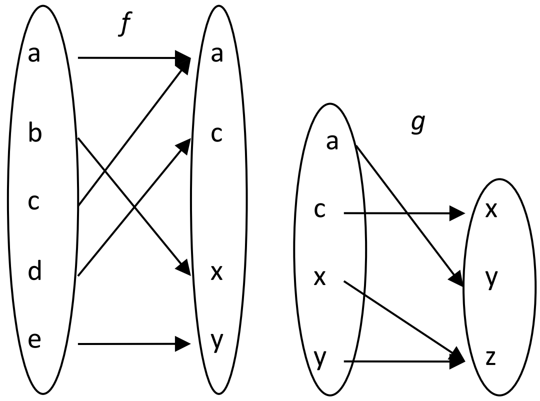
(ii) $g : \mathrm{Z} \to \mathrm{Z} : x \mapsto x+7$

(iii) $h : \mathrm{R} \to \mathrm{R} : x \mapsto x/3$

(iv) f: non-empty strings of characters → non-empty strings of binary numbers between 0 and 127, such that each character in the string is replaced by its ASCII character code

10. Specify the composition $g \circ f$ for the following functions f and g.

(i)



(ii) $f : \mathrm{Z} \to \mathrm{R} : x \mapsto x/2$ $\quad$ g: $\mathrm{R} \to \mathrm{R} : y \mapsto y+3$

(iii) f: {0,1,2,3,4,5} → {1,2,3}, where f is defined by {(0,2), (3,2), (4,1), (5,1), (1,2), (2,3)}
g: {1,2,3} → $\{x,y,z\}$, where g is defined by $\{(1,z), (2,x), (3,z)\}$

11. Suppose we have a database table for a music collection, where the columns are for name of artiste or group, name of track, and a unique identifier. The identifier is a key for the table – that is, once we know the id, we can identify exactly which artiste/group and which track we are talking about. If A is the set of artiste/band names, T is the set of track titles, and K is the set of keys, describe this in terms of functions. Still talking about functions, explain why the track name is unlikely to be a key.

12. Specify the following functions:

(i) *minus*, which takes two numbers and returns the result of subtracting the second from the first

(ii) *county*, which takes longitude and latitude representing locations in Ireland, and returns the county that that location is in

(iii) *best_of_three*, which takes three numbers as input, and returns the largest

13. For a function $f : A_1 \times A_2 \times \ldots \times A_n \to B$, define the following:

(i) range

(ii) image

and state what the following terms would mean

(iii) injective

(iii) surjective

(iii) bijective

Problems 3: Relations

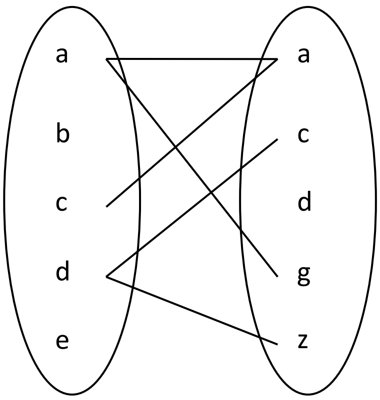
1. Sketch an arrow diagram for the following relations:

(i) A = {alice, bob, carol, darragh, eileen}, B = {oliver, padraig, ronan, susan, ted}
$R \subseteq AxB$ = {(alice, ted), (alice, susan), (bob, ronan), (carol,ted), (eileen, oliver)}

(ii) N = {-4,-3,-2,-1,0,1,2,3,4}, $R \subseteq NxN$ is the relation where the 2nd element is the square of the 1st

(iii) L={Boehme, Brown, Herley, Manning, Morrison},
M={CS1106, CS1110, CS1112, CS1115, CS1117}
$R \subseteq LxM$ is the relation where the first element is the lecturer on the second

2. Write out the set notation to describe the following relations:



(i)

(ii) "has as lecture room" between the set of computer science modules for which you are registered, and the set of lecture rooms on your timetable

(iii) "is at least 2 more than, but less than 5 more than" from the set {0,1,2,3,4,5,6} to itself.

3. If *R* is the relation of Q1(iii), and *S* is the relation of Q2(ii) above, sketch the arrow diagram for S^oR.

4. State the sets result from the following operations (note the set may contain ordered pairs):

(i) select * from the relation R in **Q1**(i), where the element of A is "alice" or "carol"
(ii) select * from the relation R in **Q1**(i), where the element of B is in{padraig,ronan,ted}
(iii) select * from the relation in **Q2**(ii) where the module is CS1112
(iv) project the relation R in **Q1**(i) onto A
(v) project the answer to **Q4**(i) onto B
(vi) select * from R^oS where the first element of the tuple is 2, 3 or 4,
where S:{1,2,3,4,5}x{1,2,3,4,5} = {(a,b)| a = 2*b}
and R:{1,2,3,4,5}x{0,1,2,3,4} = {(a,b)| b = a-1}

5. Let A be a set of soccer players , B be a set of soccer teams, and C a set of countries. Define the relations "is_the_country_of" ($\subseteq C\times A$) and "plays_for" ($\subseteq A \times B$). What do the following expressions mean, in English?

(i) project *is_the country_of* onto C
(ii) project *plays_for* onto B
(iii) is_the_country_of^{-1}
(iv) plays_for^o is_the_country_of
(v) project the result of (select * from *plays_for^o is_the_country_of* where country=ireland) onto B
[assuming Ireland was one of your countries]

6. For the following homogeneous relations on the set $\{0,1,2,3,4,5\}$, state whether they are (a) reflexive, anti-reflexive or neither, (b) symmetric, anti-symmetric or neither, and (c) transitive or not

(i) $\{(0,1),(1,2),(4,3),(1,0),(4,5),(5,4),(3,4),(2,1)\}$
(ii) $\{(0,0),(0,2),(0,4),(1,1),(1,3),(1,5),(2,2),(2,4),(3,3),(3,5),(4,4,),(5,5)\}$
(iii) $\{(0,0),(1,1),(2,4)\}$

7. Which of the following are equivalence relations? For each one that is an equivalence relation, write out its equivalence classes.

(i) A = {a,b,c,d}, R = {(a,a),(b,b),(b,c),(b,d),(c,c),(c,d),(d,d)}
(ii) A = {a,b,c,d}, R = {(a,a),(b,b),(b,c),(b,d),(c,b),(c,c),(c,d),(d,b),(d,c),(d,d)}
(iii) A= {0,1,2,3}, R is the relation "is_the_square_of"
(iv) A = {beef, pork, carrot, potato, onion, lamb, haddock}
R = {(beef,pork), (haddock,haddock),(lamb,beef), (carrot,onion), (onion,potato),(pork,beef), (carrot,carrot), (beef, lamb), (onion,carrot), (potato,carrot), (beef,beef), (onion,onion), (carrot,potato),(potato,onion), (potato,potato), (pork,lamb), (pork,pork), (lamb,lamb), (lamb,pork)}

8. State the transitive closure of the following relations:

(i) A = {1,2,3,4,5}, R = {(1,2), (2,4), (1,3), (4,5)}
(ii) A = {cork, shannon, heathrow, madrid, glasgow, frankfurt}
R = {(cork, glasgow), (cork,heathrow), (heathrow,madrid), (heathrow, frankfurt), (shannon,glasgow)}
(iii) R = "is_a_child_of" (describe the transitive closure for this in English)

9. For each of the following relations on a single set, state whether it is an order relation or not, and if it is an order, state whether it is strict or not strict, and state whether it is total. If it is not an order relation, state why it is not.

(i) A = $\{a,b,c,d\}$, R = $\{(a,b)$,(a,d),(b,d), $(c,d)\}$
(ii) A = $\{a,b,c\}$, R = $\{(a,a)$, (a,b),(a,c),(b,c),$(b,a)\}$
(iii) A = $\{1,2,3,4\}$, R = $\{(3,2)$, $(3,4))$,$(3,1)$,$(2,4)$,$(2,1)$,$(4,1)\}$
(iv) A = $\{a,b,c,d\}$, R = $\{(a,b)$,(b,d),(c,d),$(b,c)\}$
(v) A = {*apple, banana, orange*}, R = {(*apple,apple*), (*apple,orange*)}
(vi) A = $\mathcal{P}\{1,2,3\}$, R is the relation where B R C if and only if $B \subset C$
(vii) A is a set of people, R is the relation "is an ancestor of"
(viii) A is a set of people, R is the relation "is a blood relative of"
(ix) A = $\mathcal{P}\{1,2,3\}$, R is such that B R C if and only if $|B| \leq |C|$

10. Sketch the Hasse diagrams for the following relations:

(i) A = $\{1,2,3,4,5\}$, R = $\{(1,2),(1,3),(1,4),(1,5),(2,3),(2,4),(2,5),(3,4),(3,5),(4,5)\}$
(ii) A = $\{a,b,c,d,e,f,g\}$, R = $\{(a,c),(a,d),(a,f),(a,g),(b,d),(b,f),(b,e),(b,g),(c,f),(d,f),(d,g),(e,g)\}$
(iii) A = $\mathcal{P}\{1,2,3\}$, R = $\subset$

11. A *topological sort* for a partially-ordered set is a sequence of all the elements of the set such that all the order relations are obeyed. For example, for the partial order which is the solution to **9**(i) above, <*a,b,c,d*> is a topological sort; so is <*a,c,b,d*>; but <*a,d,b,c*> is not, because *d* appears before *b* in the sequence, while the relation says *b* must be before *d*.

A software engineering project requires the following tasks:
{ *document_A, write_global_tests, develop_module_A, test_A, test_B, , document_system document_B, develop_module_C, determine_requirements, test_C, document_C, develop_module_B, integrate_modules, test_integration, design_architecture* }.
The order relation on the tasks specifies which tasks must be completed before which other tasks, and has the following links in its Hasse Diagram:
{*(determine_requirements, design_architecture), (design_architecture, write_global_tests), (design_architecture, develop_module_A), (design_architecture, develop_module_B), (design_architecture, develop_module_C), (develop_module_A, test_A), (develop_module_B, test_B), (develop_module_C, test_C), (test_A, document_A), (test_B, document_B), (test_C, document_C), (document_A, integrate_modules), (document_B, integrate_modules), (document_C, integrate_modules), (integrate_modules, test_integration), (write_global_tests, test_integration), (test_integration, document_system)*}
Find a topological sort for the set of tasks.

Note: this is part of the problem of project scheduling: the Hasse diagram states the ordering constraints on the tasks, and the topological sort is a possible sequence in which you could complete the tasks. A more realistic version would include estimates of how long each task takes, and statements of when each task could possibly be executed (for example, test B might require a particular machine, and that machine might only be available at limited times of the day). Some topological sorts might not be compatible with the time windows, so the problem is then to find a topological sort which is compatible with the time windows, and which completes all the tasks in the shortest total time.

12. Write an algorithm for generating a topological sort for any given partial order relation. You can assume that: you can select an arbitrary element of the set; you can remove a named element from the set; for any element *y*, you can ask for the name of another element *x* such that (*x*,*y*) is in the relation, or NULL if there is no such element *x*; you can create an empty sequence S; you can add any element *z* to either the front or the back of the sequence S; you can detect when the set is empty. Write the algorithm first in English, and then try to write it in a formal style.

Problems 4: Propositional logic

1. Which of the following are propositions?
(i) Cork is a city in Ireland
(ii) Dublin is a beautiful city
(iii) Why does it always rain on me?
(iv) Put the cash in this brown envelope
(v) 3*7 = 21
(vi) x = 7 + 1
(vii) 25 / x = 5
(viii) Enter "OK" to proceed to the next level.
(ix) The entered password matches the password stored on file
(x) Do you have a registered account on this system?

2. State whether the following statements are using an inclusive or, an exclusive or, or something else:
(i) If you are a CS student or a staff member, you can enter B10b
(ii) You can have that laptop in silver or black
(iii) You must have a very good CV or a friend in high places to get that job
(iv) You can have a free cup of coffee or a free cup of tea
(v) If you use this ticket, you can sit in either the balcony or the stalls.

3. Let p = "The user has paid his subscription fee" and q = "The user gets access to the database". Express the following statements as compound propositions.
(i) The user has not paid his subscription fee
(ii)The user has paid his subscription fee and gets access to the database
(iii) If the user has paid his subscription fee, then the user gets access to the database
(iv) The user gets access to the database if and only if the user has paid his subscription fee

4. Let p = "I have collected all five weapons" and q = "I can defeat the enemy in level 4" and r = "I have less than 2 units of energy left". Express the following propositional formulae in English.
(i) $\neg q$
(ii) $p \vee q$
(iii) $q \rightarrow p$
(iv) $q \wedge p$
(v) $q \leftrightarrow (p \wedge \neg r)$

5. Express the following statements as propositional formula using the conditional or biconditional. NOTE: you must state explicitly what 'p' and 'q' represent (or any other propositional symbol)
(i) If you go down to the woods today, you are in for a big surprise
(ii) If you donate money to the governing party, you will get special favours
(iii) In order to pass CS1112, you must read all the lecture notes
(iv) You need to pay your subscription before you can watch that channel
(v) It is sunny when the exams are on, and the exams are on when it is sunny
(vi) For the user to save files to this disk, all you need is enough free space
(vii) To enter this website, it is a necessary and sufficient condition that you have an account.
(viii) Only when you have responded to this email can you join the mailing list
(ix) Whenever I visit this coffeeshop, I find you there
(x) On the days I drive to work, there is a traffic jam

6. If p, q and r are propositional symbols, which of the following are well-formed formulae of propositional logic (using the symbols presented in lectures)?

(i) p
(ii) $\neg r$
(iii) $(q \neg\rightarrow p)$
(iv) $p)$
(v) $p \vee q$
(vi) $p\ (\rightarrow)\ r$
(vii) $r \leftrightarrow (p \wedge q)$
(viii) $((q \vee (r \rightarrow p))$
(ix) p *and* $q = T$
(x) $((p \wedge (q \rightarrow (\neg p \vee r))) \leftrightarrow p)$

7. Rewrite the following simplified formulae by adding brackets according to the rules of precedence and association

(i) $p \vee q \rightarrow r$
(ii) $p \wedge r \rightarrow q \vee \neg r$
(iii) $p \vee r \wedge p \leftrightarrow q$
(iv) $p \vee q \wedge r \vee t \wedge w \vee s \wedge x$
(v) $q \rightarrow r \leftrightarrow \neg t$

8. Sketch the structure trees for the following propositional logic formulae:

(i) $((p \vee q) \rightarrow r)$
(ii) $(p \vee (q \rightarrow r))$
(iii) $((q \wedge r) \rightarrow (\neg p \wedge q))$

9. Write truth tables for the following propositional formulae.

(i) $p \rightarrow q$
(ii) $\neg q \wedge (p \rightarrow r)$
(iii) $r \leftrightarrow (\neg q \vee r)$
(iv) $(p \rightarrow r) \wedge (\neg p \rightarrow r)$
(v) $(q \rightarrow p) \vee (\neg r \rightarrow q)$

10. Construct the truth tables for the following statements

(i) $p \vee (q \rightarrow p)$
(ii) $r \wedge (p \rightarrow \neg q)$
(iii) $\neg(p \wedge \neg r) \rightarrow (r \vee (p \rightarrow q))$

11. Show using truth tables that the following logical equivalences are correct.

(i) $p \vee p \equiv p$
(ii) $\neg\neg p \equiv p$
(iii) $p \vee (p \wedge q) \equiv p$
(iv) $p \vee (q \vee r) \equiv (p \vee q) \vee r$
(v) $p \rightarrow (q \vee r) \equiv \neg q \rightarrow (p \rightarrow r)$

12. Show using truth tables that the following pairs of statements are **not** logically equivalent.

(i) $(p \wedge q) \rightarrow r$ and $(p \rightarrow r) \wedge (q \rightarrow r)$

(ii) $p \rightarrow (q \rightarrow r)$ and $(p \rightarrow q) \rightarrow r$

13. Represent the following statement using propositional logic, find an equivalent logical statement, and then reword the original sentence to make it easier to understand.

If the database is opened, the input window is locked, if the database is not being installed

14. Represent the following statements using propositional logic, and then apply De Morgan's laws to rewrite them, and then translate back into English:

(i) *it is not true that the system is secure and open*

(ii) *it is not true that the data will be copied to drive E and drive F*

(iii) *it is not true that both x=3 and y=false*

(iv) *it is not true that either we have reached the end of the list or we have found the required element*

15. By transforming these statements using series of logical equivalences, show they are always true.

(i) $s \vee (s \rightarrow t)$

(ii) $\neg(p \wedge \neg r) \rightarrow (r \vee (p \rightarrow q))$

16. State whether the following formulae are tautologies, contradictions, or satisfiable statements. If they are satisfiable, show at least one situation in which they are true and one in which they are false.

(i) $(p \wedge (p \rightarrow q)) \rightarrow q$

(ii) $\neg(p \rightarrow q) \rightarrow \neg q$

(iii) $(p \vee q) \rightarrow (p \vee r)$

(iv) $(p \leftrightarrow q) \leftrightarrow (\neg p \leftrightarrow q)$

17. Using truth tables, show that the following rules of inference are always valid (i.e. show that whenever the premises are true, the conclusion is also true):

(i)
$$\begin{array}{l} p \rightarrow q \\ \underline{q \rightarrow r} \\ p \rightarrow r \end{array}$$

(ii)
$$\begin{array}{l} p \rightarrow q \\ \underline{\neg q \quad\quad} \\ \neg p \end{array}$$

(iii)
$$\begin{array}{l} p \leftrightarrow q \\ \underline{p \quad\quad} \\ q \end{array}$$

18. What rules of inference are being applied in the following arguments?

(i) *Either there is at least 5 MB of free space or the operation cannot be completed. There is only 4MB of free space. Therefore the operation cannot be completed.*

(ii) *If the user clicks the reset button, all data is erased. The data has not been erased. Therefore the user did not click the reset button.*

19. Using rules of inference, prove (using the format of 4.24 in the course notes) that the statement must follow from the initial facts:

(i) Initial facts: $p \to \neg q$
$r \to q$
r
statement: $\neg p$

(ii) Initial facts: $\neg(p \wedge q) \to (r \wedge s)$
$s \to t$
$\neg t$
statement: p

20. (i) Translate the following sentences into propositional logic

The database is not locked.
Either I can save the result in the database, or the database is locked.
If I can save the result in the database then the process will be completed.

(ii) Using a formal logical argument (4.24), show from the above three statements we can prove

The process will be completed.

21. (i) Translate the following sentences into propositional logic

If I can access the web, then I will play the game
If I can't access the web, then I will complete the exercises.
If I complete the exercises, I will understand the material.

(ii) Using a formal logical argument (4.24), show from the above three statements we can prove

If I do not play the game, then I will understand the material.

22. **(i)** Translate the following sentences into propositional logic

Either the north door was opened or the south door was opened
If the intruder is in the north wing then the intruder was spotted at 15:00
If the north door was opened then the intruder is in the north wing
If the south door was opened then the intruder is in the south wing
The intruder was not spotted at 15:00

(ii) Using a formal logical argument (4.24), show from the above statements we can prove

The intruder is in the south wing

CHALLENGE QUESTIONS

23. (i) Show that we do not need either of the connectives "$\to$" or "$\leftrightarrow$". That is, show that any statement which uses either of those connectives can be rewritten as a logically equivalent statement using only "$\neg$" , "$\vee$"and "$\wedge$". (Hint: $(p \leftrightarrow q) \equiv ((p \to q) \wedge (q \to p))$)

(ii) Show that we do not need any of the connectives "$\wedge$", "$\to$" or "$\leftrightarrow$". That is, show that any statement which uses any of those connectives can be rewritten as a logically equivalent statement using only "$\neg$" and "$\vee$". (Hint: de Morgan's laws)

24. A propositional statement is in *conjunctive normal form* if it has the form $C_1 \wedge C_2 \wedge \ldots \wedge C_n$ where each C_i has the form $(L_1 \vee L_2 \vee \ldots L_t)$ and each L_i is either an atomic symbol (*p, q, r,* etc.) or the negation of an atomic symbol ($\neg p$, $\neg q$, $\neg r$, etc.). Any propositional logic statement can be translated into an equivalent conjunctive normal form statement.
So $(p \vee \neg q \vee r) \wedge (q \vee \neg p) \wedge (\neg s \vee r \vee \neg t) \wedge (t \vee p)$ is in conjunctive normal form.

Convert the following statements into conjunctive normal form, by using logical equivalences

(i) $q \wedge (p \to r)$

(ii) $(s \vee t) \to (\neg(p \wedge s) \to q)$.