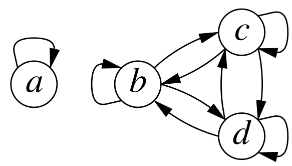
EQUIVALENCE RELATIONS

Equivalence Relation.

- A (binary) relation R on a set X is an equivalence relation if it is reflexive (R), symmetric (S), and transitive (T).
- We refer to these properties together, in short, as RST-properties.

Example.

- An equivalence relation on $X = \{a, b, c, d\}$. A simple way to represent this equivalence relation is to give its equivalence classes (defined below): $\{a\}$, $\{b, c, d\}$.



Equivalence Class $[x]$ of x in An Equivalence Relation R.

- $[x] = \{y$: x related to y in R, i.e., $(x, y) \in R\}$. (Note: $x \in [x]$, by reflexivity of R.)

  We can also say $[x] = \{y$: y related to x, i.e., $(y, x) \in R\}$, because R is symmetric.

Two Properties of Equivalence Classes.

(P.1) If $y \in [x]$, i.e., $(x, y), (y, x) \in R$ then $[x] = [y]$. (We say x and y are equivalent.)

(P.2) Two equivalence classes $[x]$ and $[y]$ are either disjoint ($[x] \cap [y] = \varnothing$) or equal.

Three Steps in Proving Property (P.1).

(1) Suppose $y \in [x]$. For each $z \in [y]$, we have $(z, y) \in R$ and this together with $(y, x) \in R$ imply $(z, x) \in R$, by transitivity of R.

(2) This means $z \in [x]$, and hence $[y] \subseteq [x]$.

(3) Also, $(y, x) \in R$ implies $x \in [y]$ and hence $[x] \subseteq [y]$ as in (1)-(2). We now have $[x] = [y]$ because $[y] \subseteq [x]$ and $[x] \subseteq [y]$.

Proof of Property (P.2).

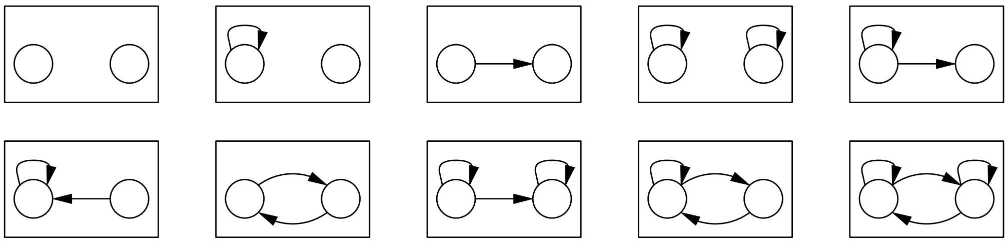
- If $[x] \cap [z] \notin \varnothing$ and $y \in [x] \cap [z]$, then $[x] = [y] = [z]$.

Disjoint Equivalence Classes Form a Partition of X.

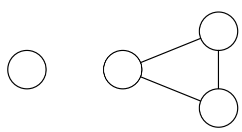
- A partition of X is decomposition of X into 1 or more non-empty disjoint subsets.
- #(Partitions of X) = #(equivalence relations on X).

Practice Questions.

1. Shown below are the structures of all relations on $|X| = 2$. Mark those which correspond to equivalence relations.



2. Show the digraph of structures of equivalence relations on $n = 3$ items. Also, for each digraph show #(equivalence relations) for that structure.
3. Show #(equivalence relations on n items with exactly 2 equivalence classes). Verify your answer for $n = 2$ and 3 from your solutions of Problems 1 and 2.
4. Shown below is the graph (undirected links and without loops to simplify the diagram) for the structure of the equivalence relation given in the previous page; show all other structures of equivalence relations on $\{a, b, c, d\}$ items. In each case, show (a) #(equivalence classes of size k) for each $k \geq 1$, and (b) #(equivalence relations for that structure).



(a) 1 equiv. class of size 1;
1 equiv. class of size 3
(b) #(equiv. rels.) = 4;

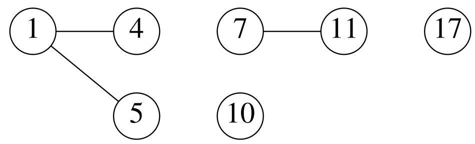
5. Complete the code below for printing the equivalence classes of an $n \times n$ equivalence relation-matrix R. Show the output for the equivalence relation with equivalence classes {0, 1, 3}, {2, 5}, {4, 6}.

```
int i = 0, //i = smallest item in an equiv. class that is found next
    j, printed[] = new int[n];
do { for (j = 0; j < n; j++) //find items in equiv. class of i
         if (1 == R[i][j])
         {  printed[j] = 1;
            System.out.println(j + " is equivalent to " + i);
         }
     //find smallest item not printed yet
     for (j = 0; j < n; j++)
         if (0 == printed[j]) break; //don't use 0 == R[i][j];
     if (j < n) //there is an equiv. class with smallest item j
        ...................
     else break;
} while (...................);
```

How do you avoid starting from `j = 0` every time to find an item not printed yet?

How would you modify the above code to count #(equivalence classes)?

6. Let $0 \le p \le q$ be two fixed numbers and X a non-empty set of numbers. We define the relation $R_{p,q}$ on X by $\{(x, y)$: $x, y \in X$ and $p \le |x - y| \le q\}$. Show the missing links in the following graph of the anti-reflexive and symmetric relation $R_{3,5}$ for $X = \{1, 4, 5, 7, 10, 11, 17\}$. Which of the transitive, non-transitive, and anti-transitive properties hold for $R_{3,5}$?



7. Consider the codes (a)-(n) below. We define ES-relation below two codes C and C' as follows: $(C, C') \in ES$ if the flowcharts of C and C' have the same structure when we ignore the contents of tests and assignments. First, argue that ES-relation satisfies RST-properties and hence it is an equivalence relation. Show the ES-equivalent classes of codes. Also, show the flowchart for each equivalence class.

```
(a) max = first;
    if (second > max) max = second;

(b) if (first > second) max = first;
    else max = second;

(c) max = first;
    if (second >= max) max = second;

(d) if (first >= second) max = first;
    else max = second;

(e) max = first;
    if (second > first) max = second;

(f) if (first > second) max = first;
    if (first <= second) max = second;

(g) max = first;
    if (second >= first) max = second;

(h) if (first >= second) max = first;
    if (first < second) max = second;

(i) max = second;
    if (first > max) max = first;

(j) if (first >= second) max = first;
    if (first <= second) max = second;

(k) max = second;
    if (first >= max) max = first;

(l) if (first < second) max = second;
    else max = first;

(m) max = second;
    if (first > second) max = first;

(n) if (first <= second) max = second;
    else max = first;
```