CSC 240

BINARY RELATIONS

Predicates: Produce a proposition based on the properties of an object.

Avenger(TonyStark) Relatives(Mario, Luigi)

$3 < 12$ $(3 + 2) = 5$

Binary Relation: A predicate that can be applied to pairs of elements.

$$xRy$$

A binary relation between x and y.

$3 = 3$

$5 > 3$ $\qquad$ xRy

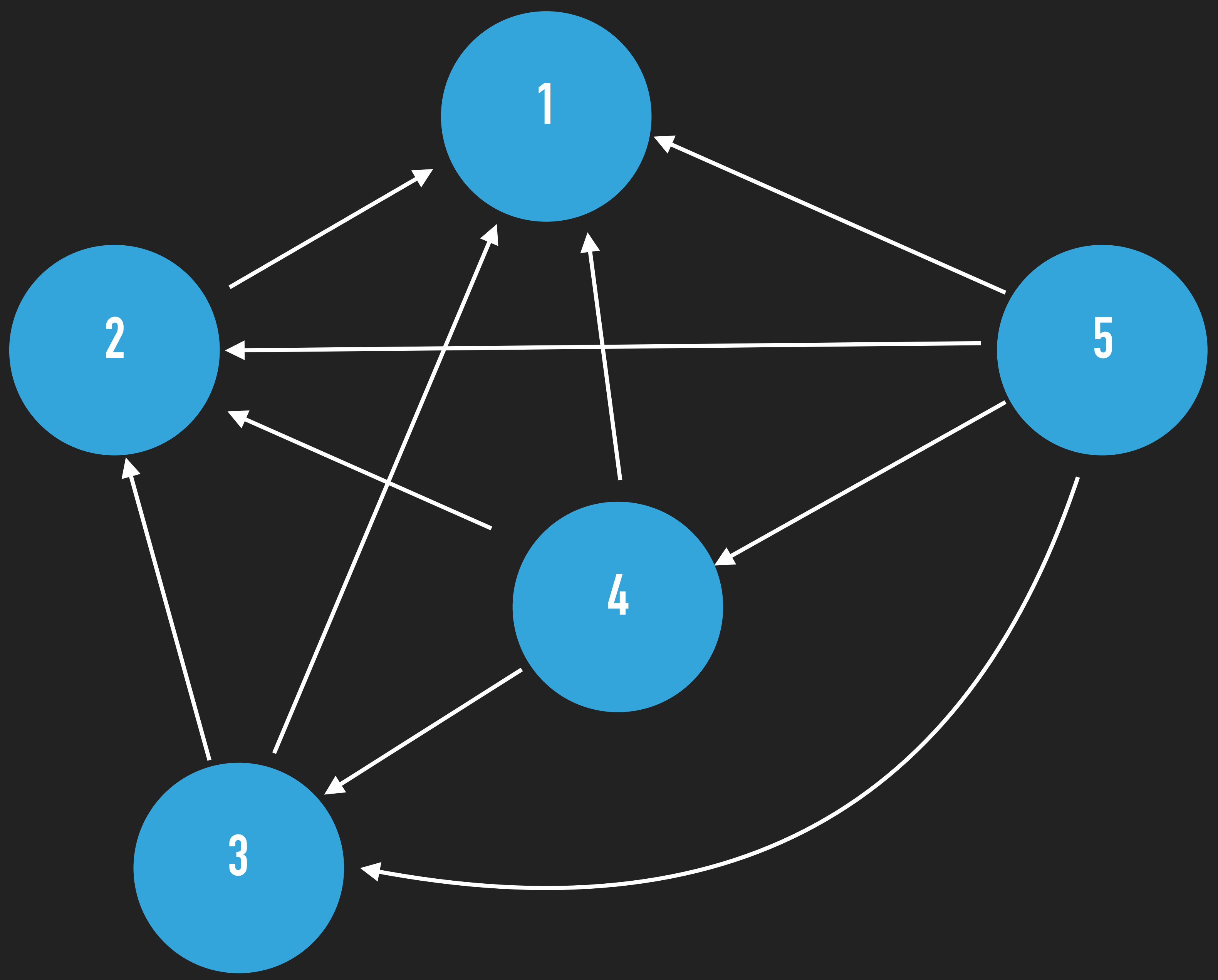
$2 < 4$

$3 = 7$

$5 < 3$ $\qquad$ $x \not R y$

$4 < 2$

VISUALIZING RELATIONS

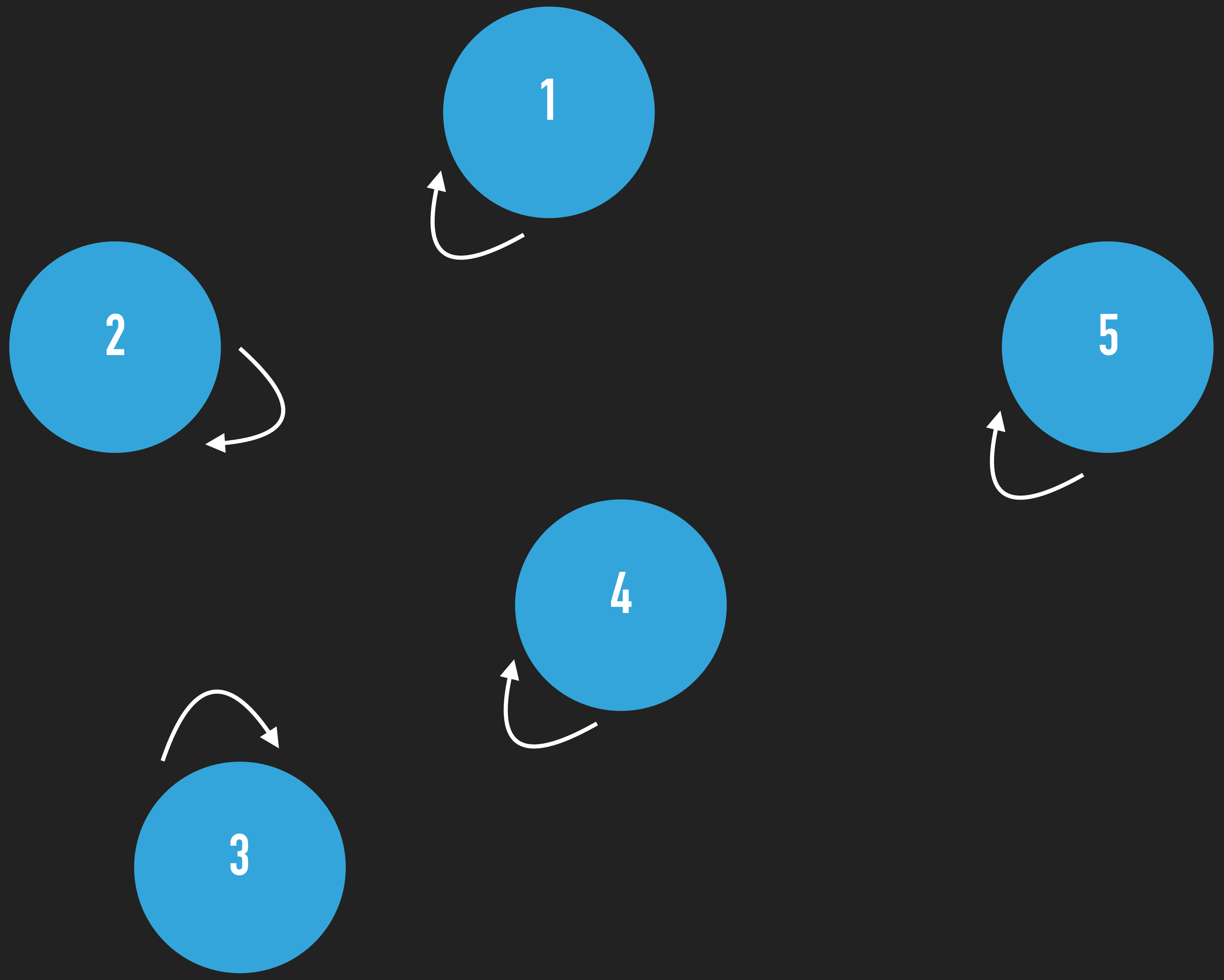


xRy

a > b

binary relations can be visualized as directed graphs

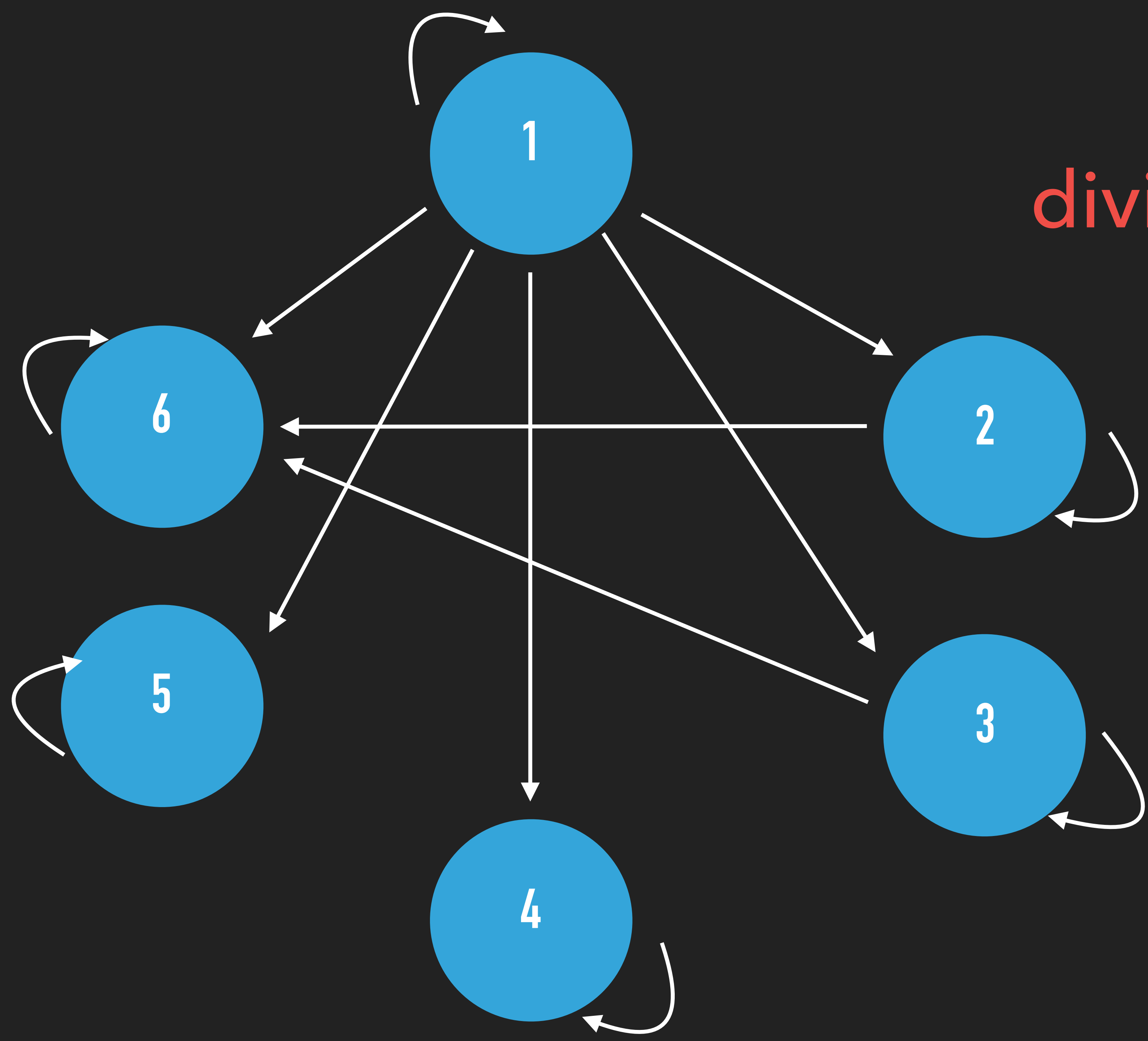
VISUALIZING RELATIONS



xRy

a = b

What is the binary relation visualized here?



xRy

dividesEvenly(x, y)

Predicates: Produce a proposition based on the properties of an object.

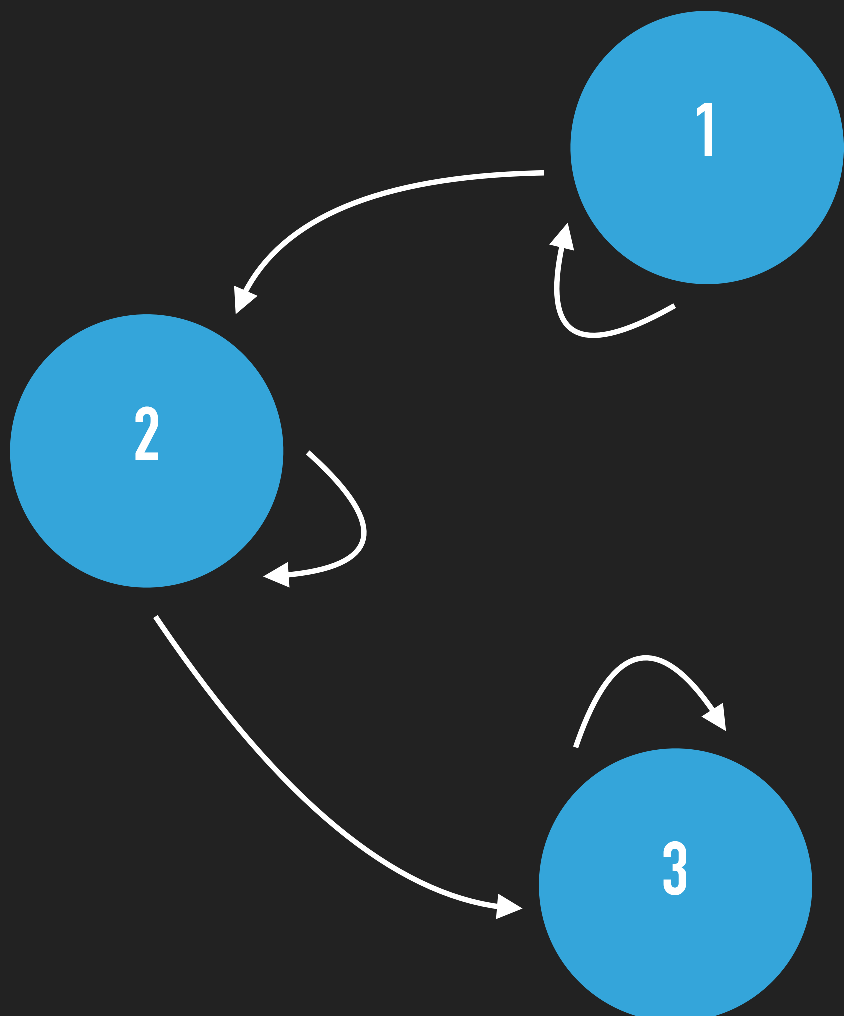
Binary Relation: A predicate that can be applied to pairs of elements.

Equivalence Relation: A special type of binary relation that is…

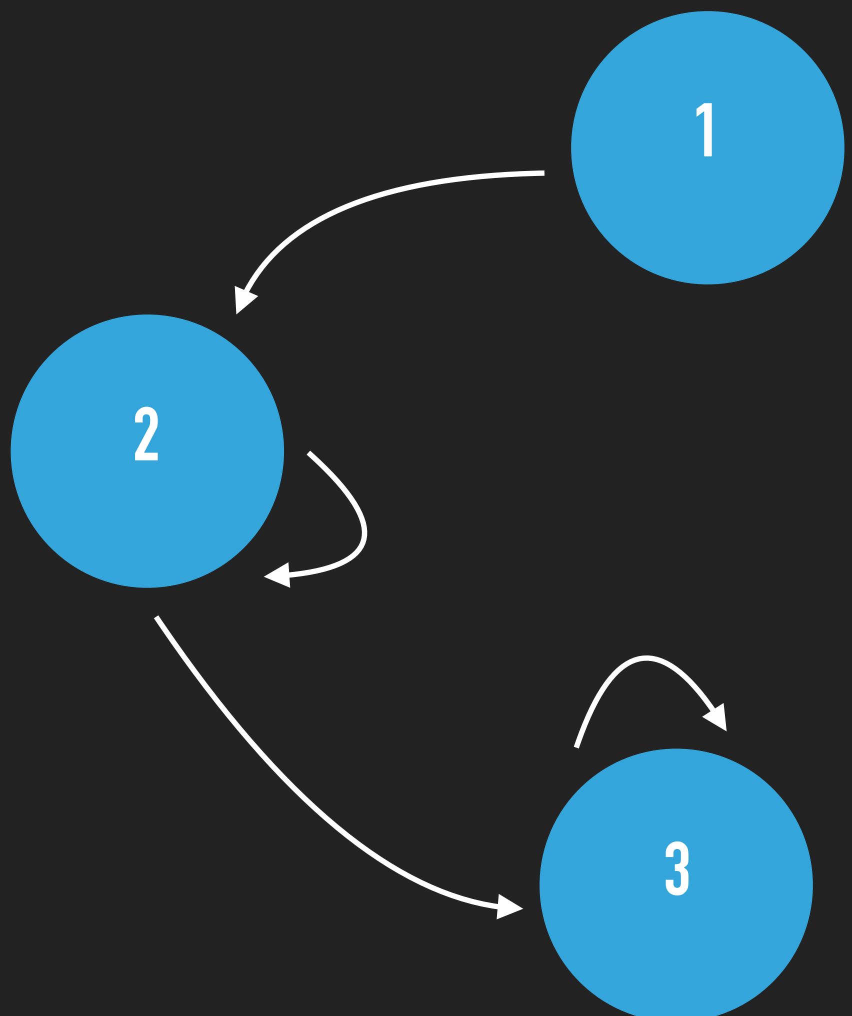
You won’t believe what happened to this binary relation….

Equivalence Relation: A special type of binary relation that is…

Reflexive: $\forall x \in S. (xRx)$



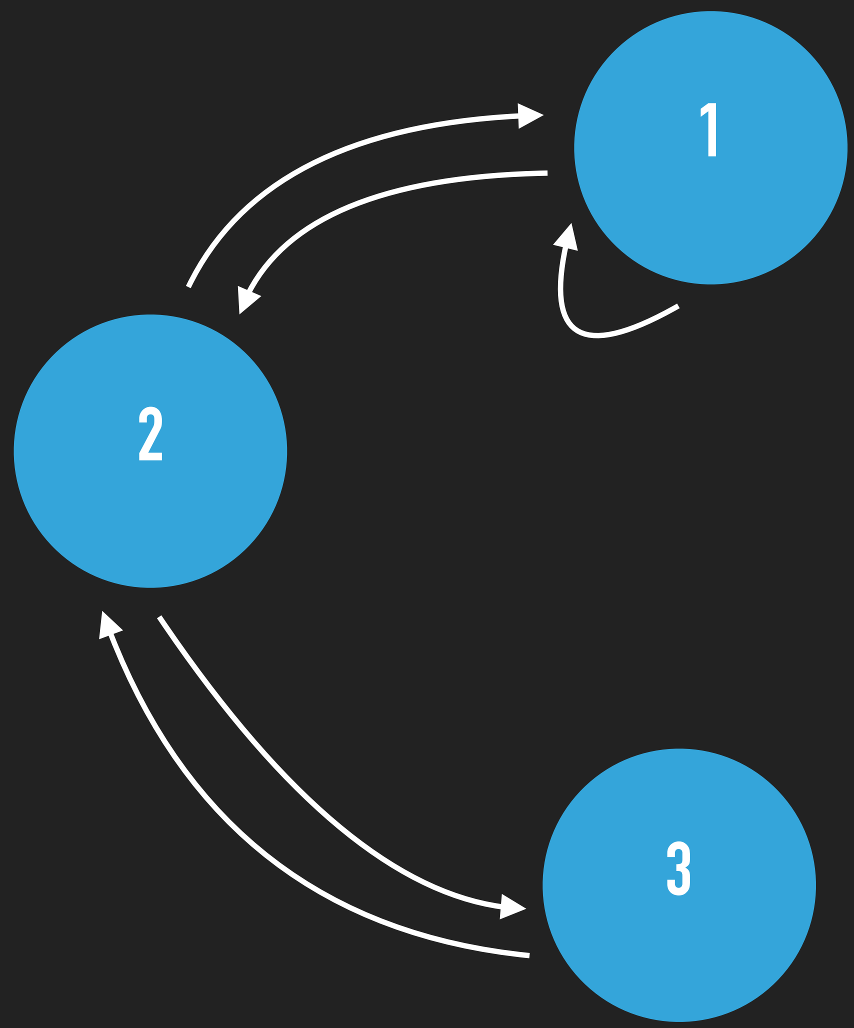
reflexive



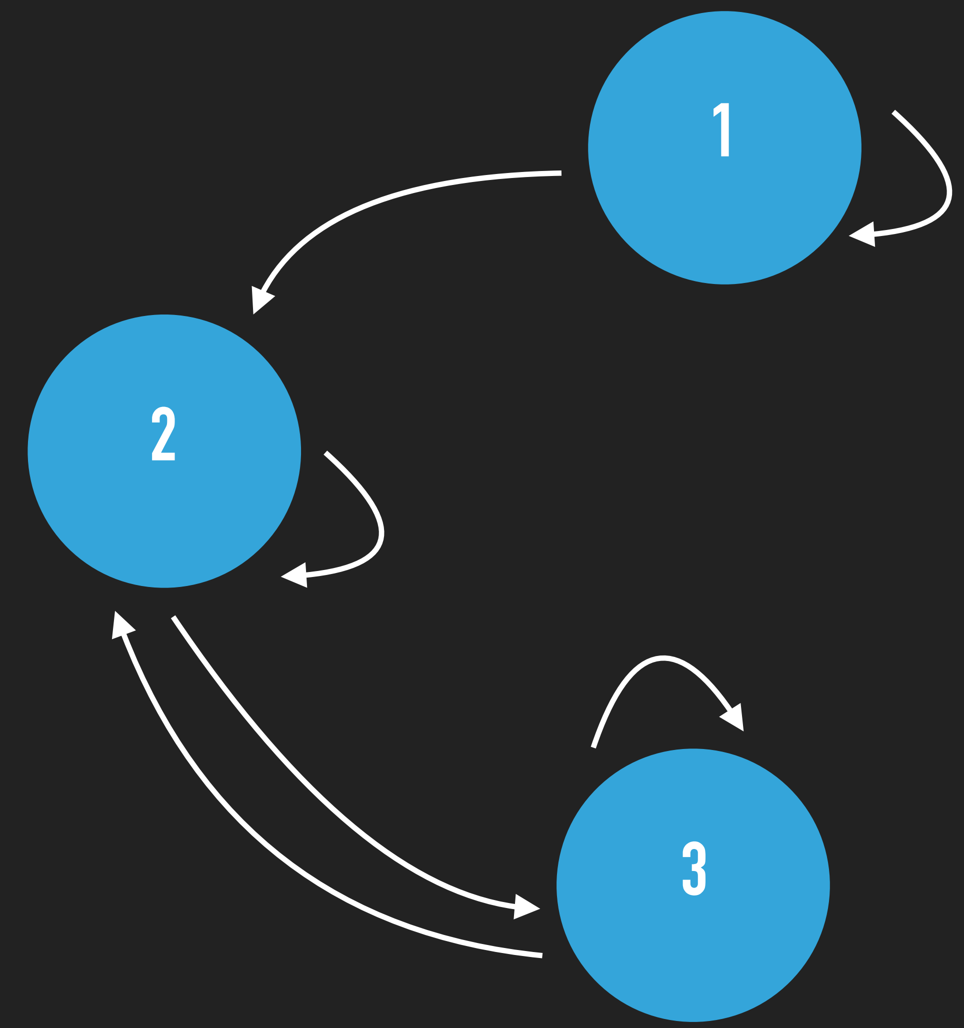
not reflexive

Equivalence Relation: A special type of binary relation that is…

Symmetric: $\forall x \in S.\ \forall y \in S.\ (xRy \rightarrow yRx)$



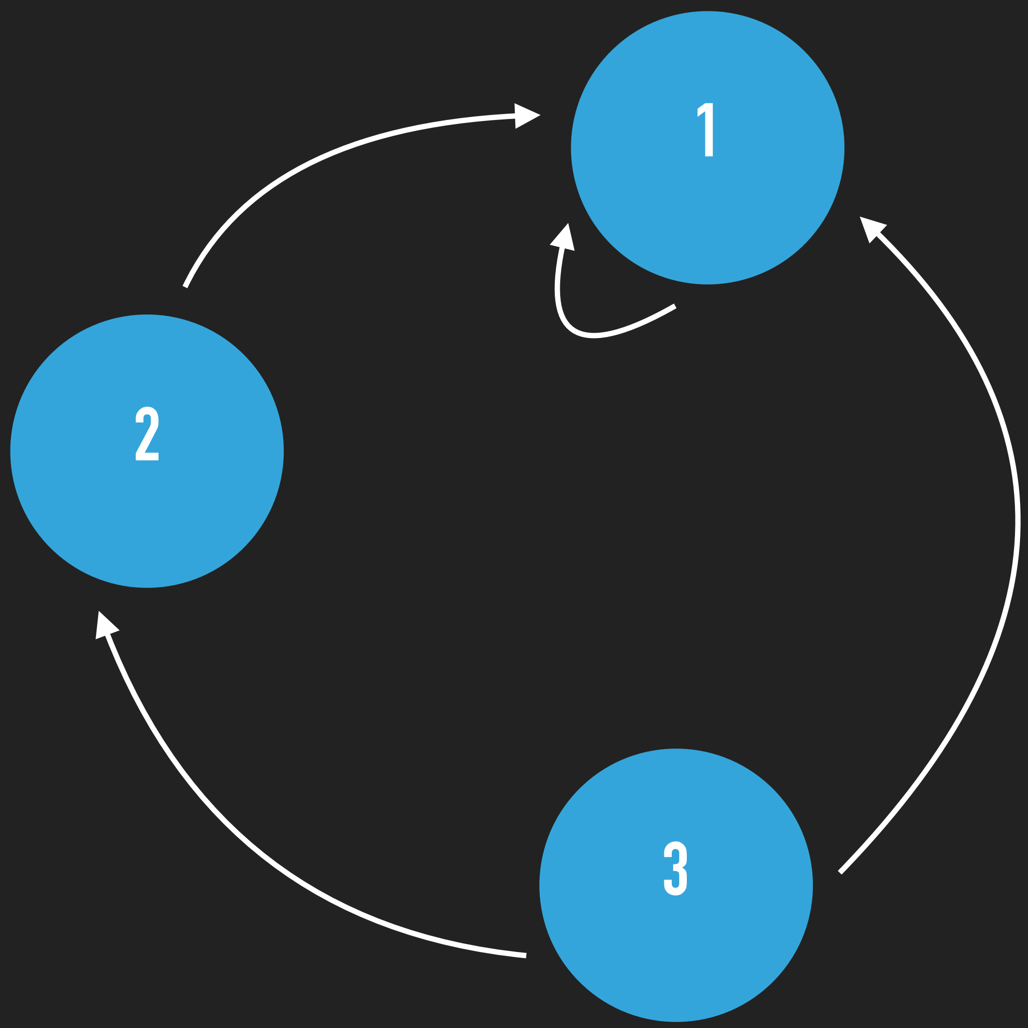
symmetric



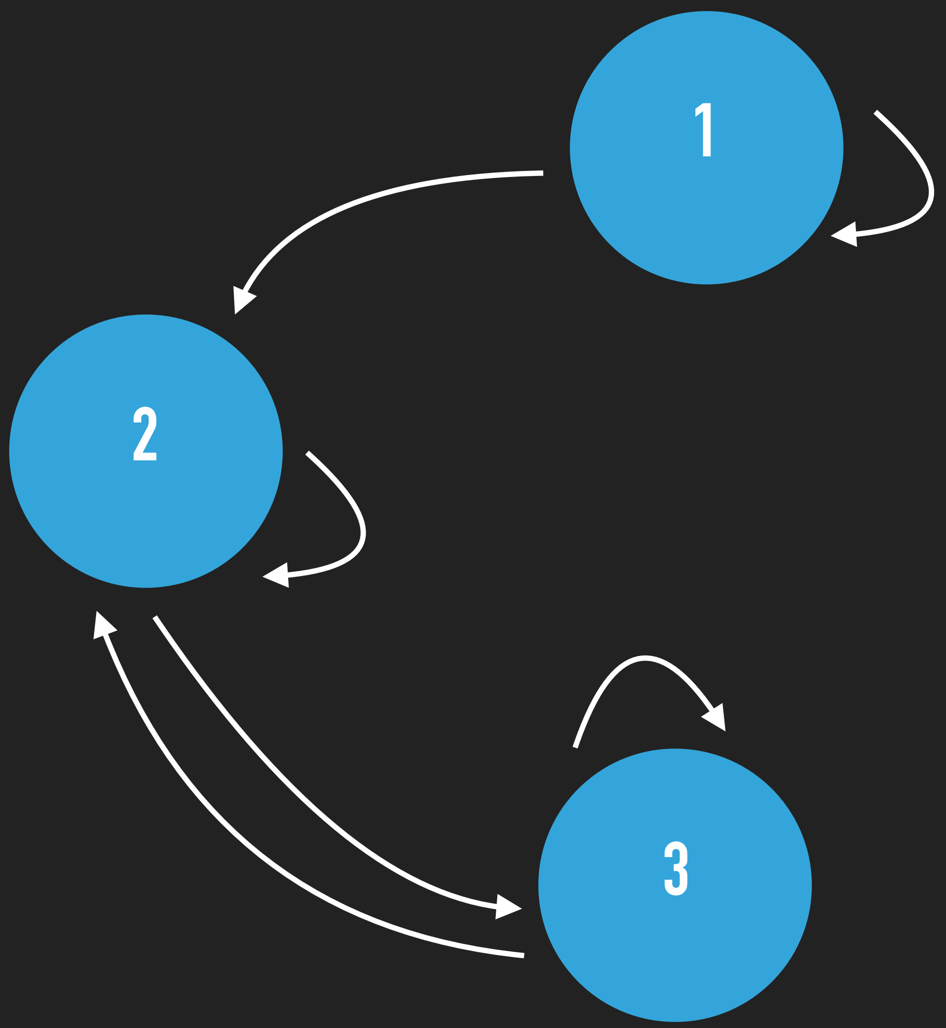
not symmetric

Equivalence Relation: A special type of binary relation that is…

Transitive: $\forall x \in S.\ \forall y \in S.\ \forall z \in S.\ (xRy \land yRz \rightarrow xRz)$



transitive



not transitive

Predicates: Produce a proposition based on the properties of an object.

Binary Relation: A predicate that can be applied to pairs of elements.

Equivalence Relation: A special type of binary relation that is…

Reflexive: $\forall x \in S.\ (xRx)$

Reflexive: every item is related to itself.

Symmetric: $\forall x \in S.\ \forall y \in S.\ (xRy \rightarrow yRx)$

Symmetric: if x is related to y, then y is related to x.

Transitive: $\forall x \in S.\ \forall y \in S.\ \forall z \in S.\ (xRy \land yRz \rightarrow xRz)$

Transitive: if x is related to y, and y is related to z, then x is related to z.

Equivalence Relations in Programming: dictionaries, maps, hash tables, sorting, etc…

User-defined classes that customize their comparison behavior should follow some consistency rules, if possible:

- Equality comparison should be reflexive. In other words, identical objects should compare equal:

  `x is y` implies `x == y`

- Comparison should be symmetric. In other words, the following expressions should have the same result:

  `x == y` and `y == x`

  `x != y` and `y != x`

  `x < y` and `y > x`

  `x <= y` and `y >= x`

- Comparison should be transitive. The following (non-exhaustive) examples illustrate that:

  `x > y and y > z` implies `x > z`

  `x < y and y <= z` implies `x < z`

python

Equivalence Relations in Programming: dictionaries, maps, hash tables, sorting, etc…

```
public boolean equals(Object obj)
```

Indicates whether some other object is "equal to" this one.

The `equals` method implements an equivalence relation on non-null object references:

- It is *reflexive*: for any non-null reference value `x`, `x.equals(x)` should return `true`.
- It is *symmetric*: for any non-null reference values `x` and `y`, `x.equals(y)` should return `true` if and only if `y.equals(x)` returns `true`.
- It is *transitive*: for any non-null reference values `x`, `y`, and `z`, if `x.equals(y)` returns `true` and `y.equals(z)` returns `true`, then `x.equals(z)` should return `true`.

java

Proofs and equivalence relations…

~ means "equivalence relation"

Given: a ~ b if a + b is even

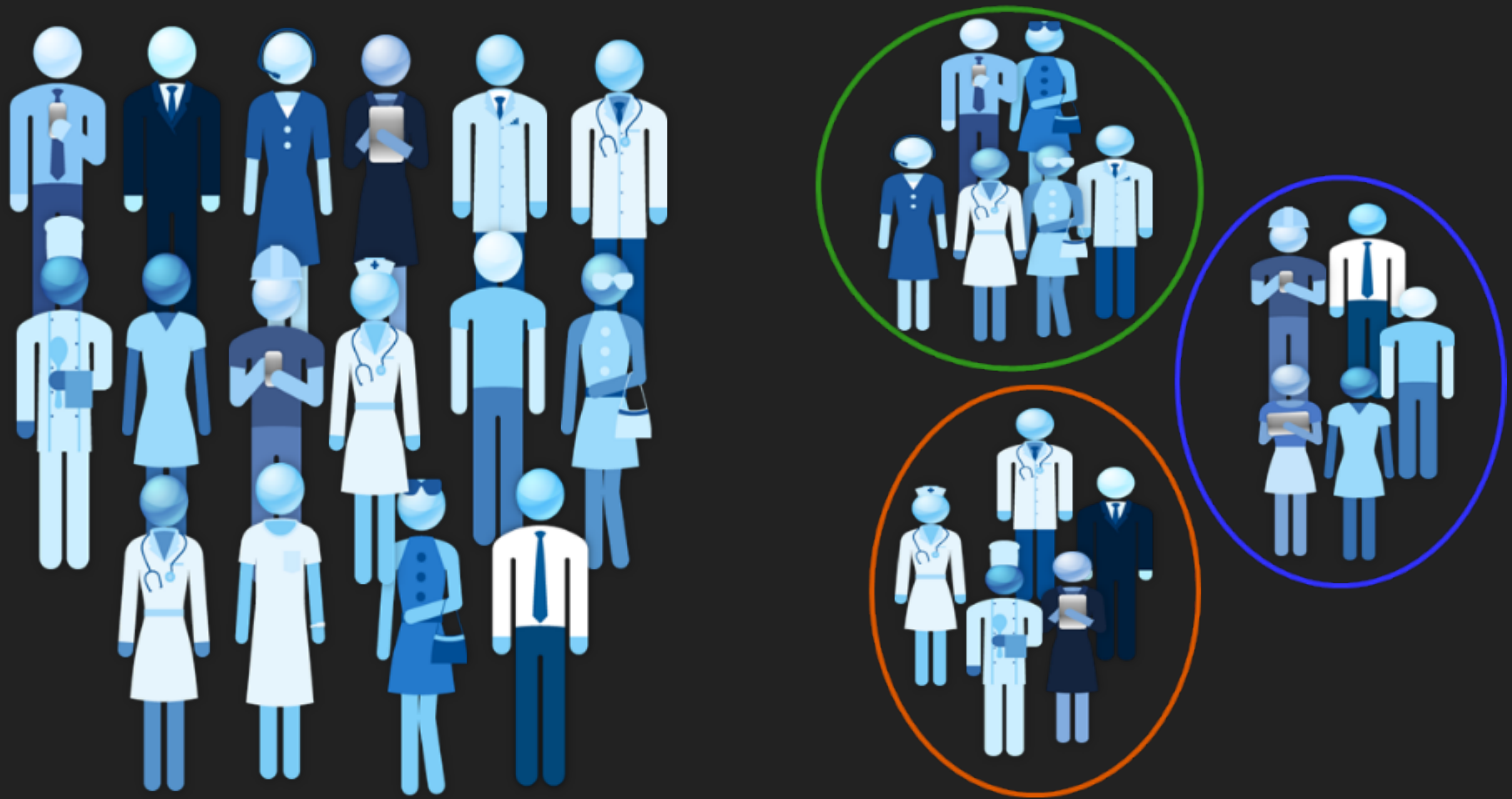
Prove:

a ~ b is reflexive

a ~ b is symmetric

a ~ b is transitive

EQUIVALENCE RELATIONS AND CLUSTERING

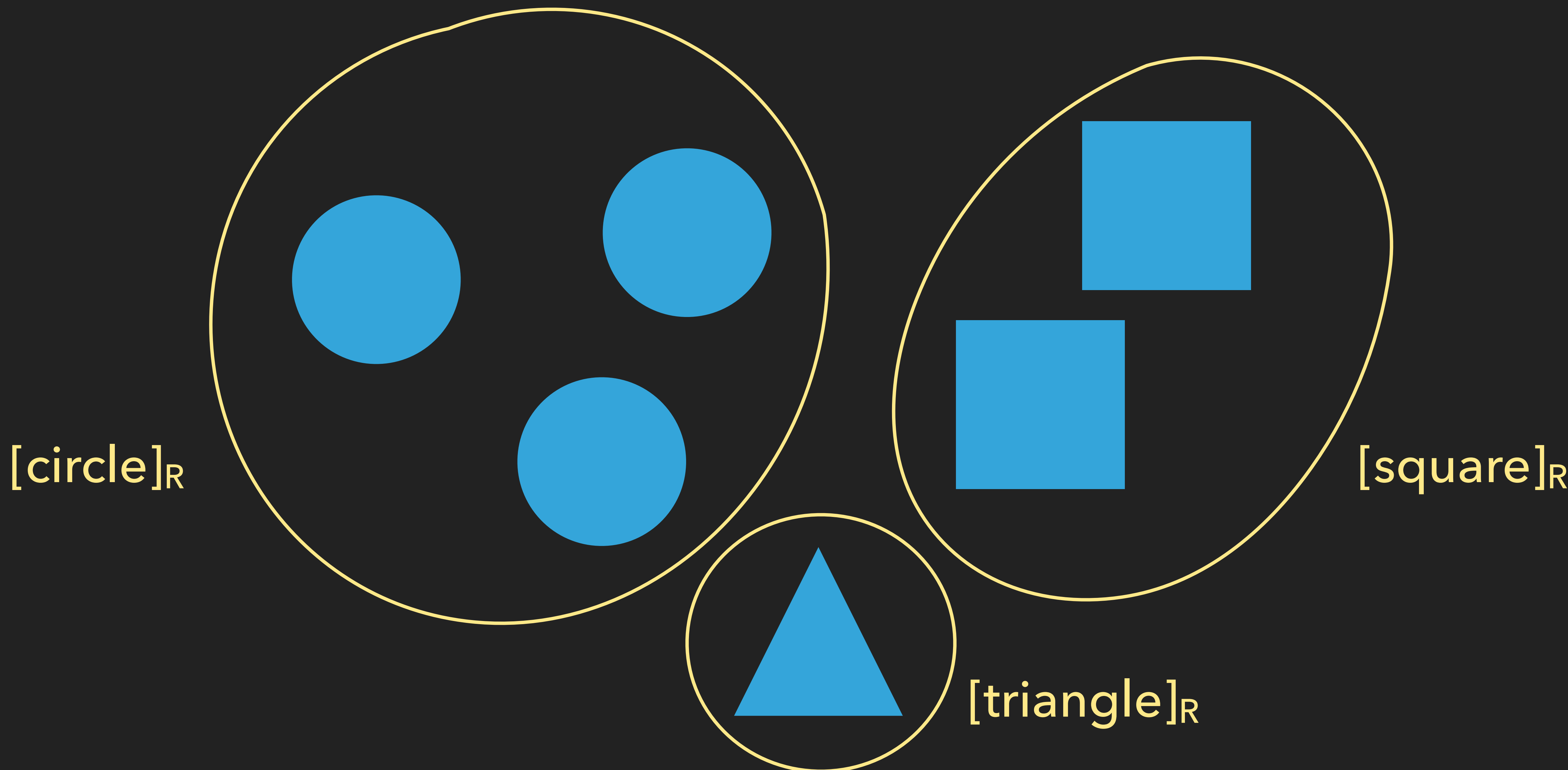


https://blogs.sas.com/content/subconsciousmusings/2016/05/26/data-mining-clustering/

EQUIVALENCE CLASSES

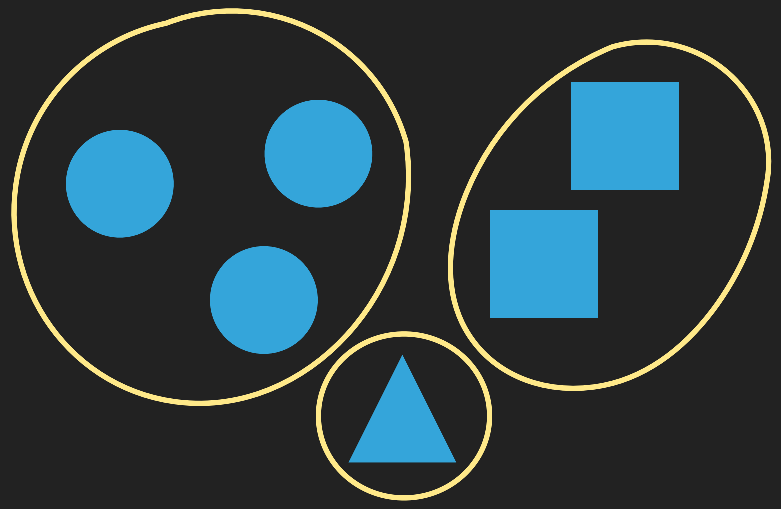
$$[x]_R = \{\ y \in S \mid xRy\ \}$$

The set of things related to x by equivalence relation R.



xRy if x and y have the same shape

THE FUNDAMENTAL THEOREM OF EQUIVALENCE RELATIONS



Partition: a grouping of a set's elements into non-empty subsets, in such a way that every element is included in one and only one of the subsets.

An equivalence relation R on a set S forms a partition of S.

xRy if x and y have the same shape

Every partition of a set S has an equivalence relation R.