## Forge: Relational Expressions 2

Some forge relational operators:

- + set union
- - set difference
- & set intersection
- ~ transpose
- -> cross product
- ^ transitive closure
- some nonempty

- no empty
- lone one or zero
- iden relation from all atoms to themselves
- univ set of all atoms in the universe

- Evaluator
  - Will evaluate expressions in the current universe being shown
  - Super helpful for finding bugs!
  - Use it to break down constraints part by part
  - Evaluator doesn't change the instance, it just evaluates whatever you write in it

You can think of relations as database tables:

- Visualize as columns containing elements e.g. Node is a set containing 4 elements
- A relation like edges has two columns (start, end). We can also have relations with more columns

no g1 & g2 : checks if it's true that there's nothing in the intersection between these two relations

> Set of formulas constrains set of instances that Forge will find

Transpose works only on binary relations; it simply flips all of its tuples (see 2/3 lecture for more information)

Why is it useful that we can do this? What's the use of a language of sets and booleans?

• Ability to tell Forge what world we want to see

Join operator changes number of columns in the evaluator -- how?

- Similar to database join
- Matches middle columns up
- Let's try edges edges for a graph of 3 Nodes:



- Interpretation of result: there are 3 reachability paths of exactly 2 'hops'
  - $\circ$   $\;$  Paths of length 1 in the initial edges relation, meaning result has paths of length 2  $\;$
- Look for matches on innermost columns and match corresponding values in outer columns