

Set Relation Language

Tom Rochette <tom.rochette@coreteks.org>

December 21, 2025 — 77e1b28a

0.1 Notation

- $A = \{1, 2, 3\}$
- $\{x \mid x = 1 \text{ or } x = 2 \text{ or } x = 3\}$
- $\{x \mid P(x)\}$ (where P is a predicate)

0.2 Tests

Results in a boolean value.

- $x \text{ memberOf } A$
- $x \text{ containedIn } A$
- $x \text{ includedIn } A$
- $x \text{ elementOf } A, x \in A, x \in A$
- $A \text{ contains } x$
- $A \text{ includes } x, A \text{ has } x$
- $A \text{ subsetOf } B, A \subseteq B$
- $A \text{ properSubsetOf } B, A < B$
- $B \text{ supersetOf } A, B \supseteq A$
- $B \text{ properSupersetOf } A, B > A$

0.3 Queries

- $\text{cardinality}(A), \text{card}(A), |A| \rightarrow \text{int}$ (set is seen as a collection of elements)
- $\text{subsetCardinality}(A), \text{sscard}(A) \rightarrow \text{int}$ (set is seen as a collection of elements AND sets)

0.4 Operations/Transformations

Results in a Set.

- $A \cup B, \text{union}(A, B), A + B, A \mid B, A \cup B$
- $A \cap B, \text{intersection}(A, B), A \& B, A \cap B$
- $A \setminus B, \text{difference}(A, B), A - B, A \setminus B, A \setminus B$
- $A \text{ symmetricDifference } B, \text{symmetricDifference}(A, B), A \text{ xor } B, A \wedge B, A \text{ sd } B$
- $A \text{ cartesianProduct } B, \text{cartesianProduct}(A, B), A \text{ cartesian } B, A \times B, A * B, A \text{ cp } B$
- $A^k, \text{power}(A), p A, A^{**}, A^{\wedge}, A^{\wedge n}$

0.5 Tests on relations

Results in a boolean value.

Consider f a function that maps items from set A to set B .

- $\text{surjective}(f), \text{sur}(f)$
- $\text{injective}(f), \text{inj}(f)$

- $\text{bijective}(f)$, $\text{bij}(f)$

0.6 Uncategorized

- Partial function
- Total function
- Reflexive
- Symmetric
- Antisymmetric
- Transitive
- Surjective
- Injective
- Bijective
- Composition
 - Cartesian product
- Membership
- Identity
- Domain
- Range
- Union - Field
- Inverse
- Image
- Preimage

0.7 Ideas

- $x \text{ Relation } y$
 - Tom isA human
 - Tom knows programming
 - Tom knows agi? (how do we determine the NOT operation based on relations alone? if there's no relation, then it implies the NOT operator)

1 References

- https://en.wikibooks.org/wiki/Set_Theory/Relations
- https://en.wikibooks.org/wiki/Set_Theory/Sets
- https://en.wikipedia.org/wiki/Set_theory
- https://en.wikipedia.org/wiki/Set-builder_notation