# Set Relation Language

Tom Rochette <tom.rochette@coreteks.org>

December 12, 2016 — f270899f

## 0.1 Notation

- A = {1, 2, 3}
- {x | x = 1 or x = 2 or x = 3}
- {x | P(x)} (where P is a predicate)

## 0.2 Tests

Results in a boolean value.

- x memberOf A
- x containedIn A
- x includedIn A
- x elementOf A, x in A, x eo A
- A contains x
- A includes x, A has x
- A subsetOf B, A <= B
- A properSubsetOf B, A < B
- B supersetOf A, B >= A
- B properSupersetOf A, B > A

## 0.3 Queries

- cardinality(A), card(A), |A| -> int (set is seen as a collection of elements)
- subsetCardinality(A), sscard(A) -> int (set is seen as a collection of elements AND sets)

## 0.4 Operations/Transformations

Results in a Set.

- A union B, union(A, B), A + B, A | B, A u B
- A intersection B, intersection(A, B), A & B, A i B
- A difference B, difference(A, B), A - B, A  B, A d B
- A symmetricDifference B, symmetricDifference(A, B) A xor B, A ^ B, A sd B
- A cartesianProduct B, cartesianProduct(A, B), A cartesian B, A x B, A * B, A cp B
- power A, power(A), p A, A**, A^, A^n

## 0.5 Tests on relations

Results in a boolean value.

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

- surjective(f), sur(f)
- injective(f), inj(f)

- bijective(f), 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 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