A general definition of the big oh notation for algorithm analysis

Kalle Rutanen

Department of Mathematics
Tampere University of Technology

June 3, 2014

Algorithm analysis

Algorithm analysis

Algorithm

An *algorithm* is a finite sequence of instructions for transforming data to another form, a process which can be followed with pen and paper.

Algorithm analysis

Algorithm

An *algorithm* is a finite sequence of instructions for transforming data to another form, a process which can be followed with pen and paper.

Example

An algorithm could provide a way to sort a sequence of integers in increasing order. Then $(0, 4, 2, 7)$ would be transformed to $(0, 2, 4, 7)$.

Algorithm analysis

Algorithm

An *algorithm* is a finite sequence of instructions for transforming data to another form, a process which can be followed with pen and paper.

Example

An algorithm could provide a way to sort a sequence of integers in increasing order. Then $(0, 4, 2, 7)$ would be transformed to $(0, 2, 4, 7)$.

Algorithm analysis

Algorithm analysis studies the correctness and complexity of a given algorithm.

Correctness analysis

Correctness analysis

Correctness analysis

Does the algorithm do what it is claimed to do?

Correctness analysis

Correctness analysis

Does the algorithm do what it is claimed to do?

Example of correctness

Prove that a given algorithm sorts any sequence of integers in increasing order, and does so in finite time for a finite sequence.

Complexity analysis

Complexity analysis

Complexity analysis

How much time / memory does it take to follow the algorithm on a given input in the worst case / best case / average case (etc.)?

Complexity analysis

Complexity analysis

How much time / memory does it take to follow the algorithm on a given input in the worst case / best case / average case (etc.)?

Example of complexity

Prove that a given algorithm never uses more than $n(n-1)/2$ number of order-comparisons to sort any sequence of n integers.

O-notation

O-notation

Motivation

Detailed complexity analysis is not interesting; big-picture scaling behaviour is.

O-notation

Motivation

Detailed complexity analysis is not interesting; big-picture scaling behaviour is.

Example

A given algorithm to sort a sequence of n integers is analyzed to take $6n^2 + 5n + 37$ comparisons in the worst case.

O-notation

Motivation

Detailed complexity analysis is not interesting; big-picture scaling behaviour is.

Example

A given algorithm to sort a sequence of n integers is analyzed to take $6n^2 + 5n + 37$ comparisons in the worst case.

Scaling behavior

This is sometimes interesting. However, more interesting is that the algorithm's complexity scales like n^2.

O-notation

Motivation

Detailed complexity analysis is not interesting; big-picture scaling behaviour is.

Example

A given algorithm to sort a sequence of n integers is analyzed to take $6n^2 + 5n + 37$ comparisons in the worst case.

Scaling behavior

This is sometimes interesting. However, more interesting is that the algorithm's complexity scales like n^2.

Simplification

The O-notation formalizes this scales-like simplification.

Definition

Definition

Definition

An O-notation in a set X is a function $O_X : F_X \to \mathcal{P}(F_X)$, where $F_X = X \to \mathbb{R}^{\geq 0}$, such that it fulfills the primitive properties.

Definition

Definition

An O-notation in a set X is a function $O_X : F_X \to \mathcal{P}(F_X)$, where $F_X = X \to \mathbb{R}^{\geq 0}$, such that it fulfills the primitive properties.

Example

To formalize our earlier example, $(6n^2 + 5n + 37) \in O_{\mathbb{N}}\left(n^2\right)$.

Definition

Definition

An O-notation in a set X is a function $O_X : F_X \to \mathcal{P}(F_X)$, where $F_X = X \to \mathbb{R}^{\geq 0}$, such that it fulfills the primitive properties.

Example

To formalize our earlier example, $(6n^2 + 5n + 37) \in O_{\mathbb{N}}\left(n^2\right)$.

Intuition

The set $O_X(f)$ contains those functions in F_X which *do not scale worse* than f.

Separate concepts

The O-notation used in pure mathematics is a concept separate from the O-notation in algorithm analysis; they have different properties.

Positive scale-invariance

Positive scale-invariance

Definition

$$O_X(\alpha f) = O_X(f) \qquad (\forall \alpha \in \mathbb{R}^{>0}, f \in F_X)$$

Positive scale-invariance

Definition

$$O_X(\alpha f) = O_X(f) \qquad (\forall \alpha \in \mathbb{R}^{>0}, f \in F_X)$$

Intuition

Positive constant factors are not interesting when comparing scaling behaviour.

Positive scale-invariance

Definition

$$O_X(\alpha f) = O_X(f) \qquad (\forall \alpha \in \mathbb{R}^{>0},\ f \in F_X)$$

Intuition

Positive constant factors are not interesting when comparing scaling behaviour.

Example

$$O_{\mathbb{N}}\left(6n^2\right) = O_{\mathbb{N}}\left(n^2\right)$$

Reflexivity

Reflexivity

Definition

$$f \in O_X(f) \qquad \forall f \in F_X$$

Reflexivity

Definition

$f \in O_X(f) \qquad \forall f \in F_X$

Intuition

A function does not scale worse than itself.

Reflexivity

Definition

$f \in O_X(f) \qquad \forall f \in F_X$

Intuition

A function does not scale worse than itself.

Example

$n \in O_{\mathbb{N}}(n)$

Transitivity

Transitivity

Definition

$$f \in O_X(g) \text{ and } g \in O_X(h) \Rightarrow f \in O_X(h) \qquad \forall f, g, h \in F_X$$

Transitivity

Definition

$$f \in O_X(g) \text{ and } g \in O_X(h) \Rightarrow f \in O_X(h) \qquad \forall f, g, h \in F_X$$

Intuition

If f does not scale worse than g, and g does not scale worse than h, then f does not scale worse than h.

Transitivity

Definition

$$f \in O_X(g) \text{ and } g \in O_X(h) \Rightarrow f \in O_X(h) \qquad \forall f, g, h \in F_X$$

Intuition

If f does not scale worse than g, and g does not scale worse than h, then f does not scale worse than h.

Example

$$n \in O_{\mathbb{N}}\left(n^2\right) \text{ and } n^2 \in O_{\mathbb{N}}\left(n^3\right) \Rightarrow n \in O_{\mathbb{N}}\left(n^3\right).$$

Order-consistency

Order-consistency

Definition

$$f \leq g \Rightarrow O_X(f) \subset O_X(g) \qquad \forall f, g \in F_X$$

Order-consistency

Definition

$$f \leq g \Rightarrow O_X(f) \subset O_X(g) \qquad \forall f, g \in F_X$$

Intuition

If all values of f are at most that of g, then those functions which do not scale worse than f do not scale worse than g either

Order-consistency

Definition

$f \leq g \Rightarrow O_X(f) \subset O_X(g) \qquad \forall f, g \in F_X$

Intuition

If all values of f are at most that of g, then those functions which do not scale worse than f do not scale worse than g either

Examples

Order-consistency

Definition

$$f \leq g \Rightarrow O_X(f) \subset O_X(g) \qquad \forall f, g \in F_X$$

Intuition

If all values of f are at most that of g, then those functions which do not scale worse than f do not scale worse than g either

Examples

$$O_X(0) \subset O_X(f) \qquad \forall f \in F_X$$

Order-consistency

Definition

$$f \leq g \Rightarrow O_X(f) \subset O_X(g) \qquad \forall f, g \in F_X$$

Intuition

If all values of f are at most that of g, then those functions which do not scale worse than f do not scale worse than g either

Examples

$$O_X(0) \subset O_X(f) \qquad \forall f \in F_X$$
$$O_{\mathbb{R}^{\geq 1}}(x^\alpha) \subset O_{\mathbb{R}^{\geq 1}}(x^\beta) \qquad \forall \alpha, \beta \in \mathbb{R}^{\geq 0} : \alpha \leq \beta$$

Order-consistency

Definition

$$f \leq g \Rightarrow O_X(f) \subset O_X(g) \qquad \forall f, g \in F_X$$

Intuition

If all values of f are at most that of g, then those functions which do not scale worse than f do not scale worse than g either

Examples

$$O_X(0) \subset O_X(f) \qquad \forall f \in F_X$$

$$O_{\mathbb{R}^{\geq 1}}(x^\alpha) \subset O_{\mathbb{R}^{\geq 1}}(x^\beta) \qquad \forall \alpha, \beta \in \mathbb{R}^{\geq 0} : \alpha \leq \beta$$

$$O_{\mathbb{R}^{\geq 1}}\left(\frac{1}{\alpha e \log_e(\beta)} \log_\beta(x)\right) \subset O_{\mathbb{R}^{\geq 1}}(x^\alpha) \qquad \forall \alpha, \beta \in \mathbb{R}^{>0}$$

Multiplicativity

Multiplicativity

Definition

$$O_X(f)O_X(g) = O_X(fg) \qquad \forall f, g \in F_X$$

Multiplicativity

Definition

$$O_X(f)O_X(g) = O_X(fg) \qquad \forall f, g \in F_X$$

Intuition

The product of a function which does not scale worse than f and a function which does not scale worse than g does not scale worse than fg. Every function in $O_X(fg)$ is a product of such functions.

Multiplicativity

Definition

$$O_X(f)O_X(g) = O_X(fg) \qquad \forall f, g \in F_X$$

Intuition

The product of a function which does not scale worse than f and a function which does not scale worse than g does not scale worse than fg. Every function in $O_X(fg)$ is a product of such functions.

Examples

Multiplicativity

Definition

$$O_X(f)O_X(g) = O_X(fg) \qquad \forall f, g \in F_X$$

Intuition

The product of a function which does not scale worse than f and a function which does not scale worse than g does not scale worse than fg. Every function in $O_X(fg)$ is a product of such functions.

Examples

$$O_{\mathbb{N}}(n)O_{\mathbb{N}}(n^2) = O_{\mathbb{N}}(n^3)$$

Multiplicativity

Definition

$$O_X(f)O_X(g) = O_X(fg) \qquad \forall f, g \in F_X$$

Intuition

The product of a function which does not scale worse than f and a function which does not scale worse than g does not scale worse than fg. Every function in $O_X(fg)$ is a product of such functions.

Examples

$$O_{\mathbb{N}}(n)O_{\mathbb{N}}(n^2) = O_{\mathbb{N}}(n^3)$$
$$O_{\mathbb{R}>0}(1/x)O_{\mathbb{R}>0}(x) = O_{\mathbb{R}>0}(1)$$

Locality

Locality

Definition

$f \in O_X(g) \Leftrightarrow \forall D \in C : (f|D) \in O_D(g|D)$
$\forall f, g \in F_X, C \subset \mathcal{P}(X)$: C is a finite cover of X.

Locality

Definition

$f \in O_X(g) \Leftrightarrow \forall D \in C : (f|D) \in O_D(g|D)$
$\forall f, g \in F_X, C \subset \mathcal{P}(X)$: C is a finite cover of X.

Intuition

An f does not scale worse than g if and only if that holds when restricted to a set of a finite cover, for all such sets.

Zero-separation

Zero-separation

Definition

$O_X(1) \not\subset O_X(0)$

Zero-separation

Definition

$O_X(1) \not\subset O_X(0)$

Intuition

There exists a function which scales worse than 0, but does not scale worse than 1.

One-separation

One-separation

Definition

$O_{\mathbb{N}>0}(n) \not\subset O_{\mathbb{N}>0}(1)$

One-separation

Definition

$O_{\mathbb{N}>0}(n) \not\subset O_{\mathbb{N}>0}(1)$

Intuition

There exists a function which scales worse than 1, but does not scale worse than n.

Composition rule

Composition rule

Definition

$$O_X(f) \circ s \subset O_Y(f \circ s) \qquad \forall f \in F_X, s : Y \to X.$$

Composition rule

Definition

$$O_X(f) \circ s \subset O_Y(f \circ s) \qquad \forall f \in F_X, s : Y \to X.$$

Intuition

A function which does not scale worse than f, mapped through s, does not scale worse than f mapped through s.

Uniqueness and existence

Uniqueness and existence

Primitive properties

The given properties are called the *primitive properties*.

Uniqueness and existence

Primitive properties

The given properties are called the *primitive properties*.

Existence

There exists a function O with the primitive properties.

Uniqueness and existence

Primitive properties

The given properties are called the *primitive properties*.

Existence

There exists a function O with the primitive properties.

Uniqueness

There exists at most one function O with the primitive properties.

Uniqueness and existence

Primitive properties

The given properties are called the *primitive properties*.

Existence

There exists a function O with the primitive properties.

Uniqueness

There exists at most one function O with the primitive properties.

Explicit definition

$f \in O_X(g) :\Leftrightarrow \exists c \in \mathbb{R}^{>0} : f \leq cg$

Prevailing definition

Prevailing definition

Definition

$f \in O_X(g) :\Leftrightarrow \exists c \in \mathbb{R}^{>0}, y \in X : (f|X^{\geq y}) \leq c(g|X^{\geq y})$
where $X \subset \mathbb{R}^d$ and $d \in \mathbb{N}$.

Prevailing definition

Definition

$f \in O_X(g) :\Leftrightarrow \exists c \in \mathbb{R}^{>0}, y \in X : (f|X^{\geq y}) \leq c(g|X^{\geq y})$
where $X \subset \mathbb{R}^d$ and $d \in \mathbb{N}$.

Problem

Our definition is different to the prevailing definition, which has been used for decades. Is there something wrong with the prevailing definition?

Prevailing definition

Definition

$f \in O_X(g) :\Leftrightarrow \exists c \in \mathbb{R}^{>0}, y \in X : (f|X^{\geq y}) \leq c(g|X^{\geq y})$
where $X \subset \mathbb{R}^d$ and $d \in \mathbb{N}$.

Problem

Our definition is different to the prevailing definition, which has been used for decades. Is there something wrong with the prevailing definition?

Solution

The prevailing definition fulfills all of the primitive properties, except for the composition rule!

Fundamental

The composition rule is fundamental; without it the complexity analysis of an algorithm cannot be approached by dividing it into parts, and studying the complexity of each part.

An example of failure

An example of failure

Algorithm 2 An algorithm which takes as input $(m, n) \in \mathbb{N}^2$, and has time-complexity $O_{\mathbb{N}^2}(1)$ according to the prevailing definition.

```
procedure ConstantComplexity(m, n)
    j ← 0
    if m = 0 then
        for i ∈ [0, n] do
            j ← j + 1
        end for
    end if
    return j
end procedure
```

Algorithm 3 An algorithm which takes as input $n \in \mathbb{N}$, and calls another $O_{\mathbb{N}^2}(1)$ algorithm n times with varying arguments.

```
procedure BASICANALYSIS(n)
    for i ∈ [0, n) do
        CONSTANTCOMPLEXITY(0, i)
    end for
end procedure
```

Algorithm 4 An algorithm which takes as input $n \in \mathbb{N}$, and calls another $O_{\mathbb{N}^2}(1)$ algorithm n times with varying arguments.

```
procedure BASICANALYSIS(n)
    for i ∈ [0, n) do
        CONSTANTCOMPLEXITY(0, i)
    end for
end procedure
```

Composition

Computed via the composition rule, the complexity of this algorithm is $O_{\mathbb{N}}(n)$.

Algorithm 5 An algorithm which takes as input $n \in \mathbb{N}$, and calls another $O_{\mathbb{N}^2}(1)$ algorithm n times with varying arguments.

```
procedure BASICANALYSIS(n)
    for i ∈ [0, n) do
        CONSTANTCOMPLEXITY(0, i)
    end for
end procedure
```

Composition

Computed via the composition rule, the complexity of this algorithm is $O_{\mathbb{N}}(n)$.

Substitution

Computed via substitution, the complexity of this algorithm is $O_{\mathbb{N}}\left(n^2\right)$. A contradiction!

Characterization of failure

Characterization of failure

Theorem (Asymptotic composition rule)

Let $X \subset \mathbb{R}^{d_1}$, $Y \subset \mathbb{R}^{d_2}$, and $s : Y \to X$. The composition rule holds for s under the prevailing definition if and only if

$$\forall x^* \in X : \exists y^* \in Y : s(Y^{\geq y^*}) \subset X^{\geq x^*}. \tag{1}$$

Characterization of failure

Theorem (Asymptotic composition rule)

Let $X \subset \mathbb{R}^{d_1}$, $Y \subset \mathbb{R}^{d_2}$, and $s : Y \to X$. The composition rule holds for s under the prevailing definition if and only if

$$\forall x^* \in X : \exists y^* \in Y : s(Y^{\geq y^*}) \subset X^{\geq x^*}. \tag{1}$$

Subset-sum

Since the subset-sum rule implies the composition rule, the former does not hold for the prevailing definition either.

Characterization of failure

Theorem (Asymptotic composition rule)

Let $X \subset \mathbb{R}^{d_1}$, $Y \subset \mathbb{R}^{d_2}$, and $s : Y \to X$. The composition rule holds for s under the prevailing definition if and only if

$$\forall x^* \in X : \exists y^* \in Y : s(Y^{\geq y^*}) \subset X^{\geq x^*}. \tag{1}$$

Subset-sum

Since the subset-sum rule implies the composition rule, the former does not hold for the prevailing definition either.

Subset-sum or composition

Actually, assuming the other properties, the composition rule and the subset-sum rule are equivalent.

Implied properties

Implied properties

More properties

The following properties are implied by the primitive properties.

Implied properties

More properties

The following properties are implied by the primitive properties.

Implied properties

Implied properties

More properties

The following properties are implied by the primitive properties.

Implied properties

Monotonicity: $O_X(O_X(f)) \supset O_X(f)$

Implied properties

More properties

The following properties are implied by the primitive properties.

Implied properties

Monotonicity: $O_X(O_X(f)) \supset O_X(f)$
Idempotence: $O_X(O_X(f)) = O_X(f)$

Implied properties

More properties

The following properties are implied by the primitive properties.

Implied properties

Monotonicity: $O_X(O_X(f)) \supset O_X(f)$
Idempotence: $O_X(O_X(f)) = O_X(f)$
Membership rule: $f \in O_X(g) \Leftrightarrow O_X(f) \subset O_X(g)$

Implied properties

More properties

The following properties are implied by the primitive properties.

Implied properties

Monotonicity: $O_X(O_X(f)) \supset O_X(f)$
Idempotence: $O_X(O_X(f)) = O_X(f)$
Membership rule: $f \in O_X(g) \Leftrightarrow O_X(f) \subset O_X(g)$
Bounded translation-invariance:
$(\exists \beta \in \mathbb{R}^{>0} : f \geq \beta) \Rightarrow O_X(f + \alpha) = O_X(f)$

Implied properties

More properties

The following properties are implied by the primitive properties.

Implied properties

Monotonicity: $O_X(O_X(f)) \supset O_X(f)$
Idempotence: $O_X(O_X(f)) = O_X(f)$
Membership rule: $f \in O_X(g) \Leftrightarrow O_X(f) \subset O_X(g)$
Bounded translation-invariance:
$(\exists \beta \in \mathbb{R}^{>0} : f \geq \beta) \Rightarrow O_X(f + \alpha) = O_X(f)$
Positive homogenuity: $\alpha O_X(f) = O_X(\alpha f)$

Implied properties

More properties

The following properties are implied by the primitive properties.

Implied properties

Monotonicity: $O_X(O_X(f)) \supset O_X(f)$
Idempotence: $O_X(O_X(f)) = O_X(f)$
Membership rule: $f \in O_X(g) \Leftrightarrow O_X(f) \subset O_X(g)$
Bounded translation-invariance:
$(\exists \beta \in \mathbb{R}^{>0} : f \geq \beta) \Rightarrow O_X(f + \alpha) = O_X(f)$
Positive homogenuity: $\alpha O_X(f) = O_X(\alpha f)$
Positive power-homogenuity: $O_X(f)^\alpha = O_X(f^\alpha)$

Implied properties

More properties

The following properties are implied by the primitive properties.

Implied properties

Monotonicity: $O_X(O_X(f)) \supset O_X(f)$
Idempotence: $O_X(O_X(f)) = O_X(f)$
Membership rule: $f \in O_X(g) \Leftrightarrow O_X(f) \subset O_X(g)$
Bounded translation-invariance:
$(\exists \beta \in \mathbb{R}^{>0} : f \geq \beta) \Rightarrow O_X(f + \alpha) = O_X(f)$
Positive homogenuity: $\alpha O_X(f) = O_X(\alpha f)$
Positive power-homogenuity: $O_X(f)^\alpha = O_X(f^\alpha)$
Additive consistency: $uO_X(f) + vO_X(f) = (u + v)O_X(f)$

Implied properties

More properties

The following properties are implied by the primitive properties.

Implied properties

Monotonicity: $O_X(O_X(f)) \supset O_X(f)$
Idempotence: $O_X(O_X(f)) = O_X(f)$
Membership rule: $f \in O_X(g) \Leftrightarrow O_X(f) \subset O_X(g)$
Bounded translation-invariance:
$(\exists \beta \in \mathbb{R}^{>0} : f \geq \beta) \Rightarrow O_X(f + \alpha) = O_X(f)$
Positive homogenuity: $\alpha O_X(f) = O_X(\alpha f)$
Positive power-homogenuity: $O_X(f)^{\alpha} = O_X(f^{\alpha})$
Additive consistency: $uO_X(f) + vO_X(f) = (u + v)O_X(f)$
Multiplicative consistency: $O_X(f)^u O_X(f)^v = O_X(f)^{u+v}$

Implied properties

More implied properties

Implied properties

More implied properties

Maximum-consistency: $\max(O_X(f), O_X(f)) = O_X(f)$

Implied properties

More implied properties

Maximum-consistency: $\max(O_X(f), O_X(f)) = O_X(f)$
Restriction rule: $(O_X(f)|D) = O_D(f|D)$

Implied properties

More implied properties

Maximum-consistency: $\max(O_X(f), O_X(f)) = O_X(f)$
Restriction rule: $(O_X(f)|D) = O_D(f|D)$
Additivity: $O_X(f) + O_X(g) = O_X(f+g)$

Implied properties

More implied properties

Maximum-consistency: $\max(O_X(f), O_X(f)) = O_X(f)$
Restriction rule: $(O_X(f)|D) = O_D(f|D)$
Additivity: $O_X(f) + O_X(g) = O_X(f + g)$
Maximum rule: $\max(O_X(f), O_X(g)) = O_X(\max(f, g))$

Implied properties

More implied properties

Maximum-consistency: $\max(O_X(f), O_X(f)) = O_X(f)$
Restriction rule: $(O_X(f)|D) = O_D(f|D)$
Additivity: $O_X(f) + O_X(g) = O_X(f + g)$
Maximum rule: $\max(O_X(f), O_X(g)) = O_X(\max(f, g))$
Summation rule: $O_X(f + g) = O_X(\max(f, g))$

Implied properties

More implied properties

Maximum-consistency: $\max(O_X(f), O_X(f)) = O_X(f)$
Restriction rule: $(O_X(f)|D) = O_D(f|D)$
Additivity: $O_X(f) + O_X(g) = O_X(f+g)$
Maximum rule: $\max(O_X(f), O_X(g)) = O_X(\max(f,g))$
Summation rule: $O_X(f+g) = O_X(\max(f,g))$
Maximum-sum rule: $\max(O_X(f), O_X(g)) = O_X(f) + O_X(g)$

Implied properties

More implied properties

Maximum-consistency: $\max(O_X(f), O_X(f)) = O_X(f)$
Restriction rule: $(O_X(f)|D) = O_D(f|D)$
Additivity: $O_X(f) + O_X(g) = O_X(f + g)$
Maximum rule: $\max(O_X(f), O_X(g)) = O_X(\max(f, g))$
Summation rule: $O_X(f + g) = O_X(\max(f, g))$
Maximum-sum rule: $\max(O_X(f), O_X(g)) = O_X(f) + O_X(g)$
Injective composition rule: $O_X(f) \circ s = O_Y(f \circ s)$ (s injective)

Implied properties

More implied properties

Maximum-consistency: $\max(O_X(f), O_X(f)) = O_X(f)$

Restriction rule: $(O_X(f)|D) = O_D(f|D)$

Additivity: $O_X(f) + O_X(g) = O_X(f+g)$

Maximum rule: $\max(O_X(f), O_X(g)) = O_X(\max(f,g))$

Summation rule: $O_X(f+g) = O_X(\max(f,g))$

Maximum-sum rule: $\max(O_X(f), O_X(g)) = O_X(f) + O_X(g)$

Injective composition rule: $O_X(f) \circ s = O_Y(f \circ s)$ (s injective)

Subset-sum rule:
$f \in O_Y(g) \Rightarrow \left(x \mapsto \sum_{y \in S(x)} f(y)\right) \in O_X\left(\sum_{y \in S(x)} g(y)\right)$

Prevailing definition

The prevailing definition fulfills all of the implied properties, except for the subset-sum rule, and the injective composition rule.

Relation definitions

Given the O-notation $O_X : F_X \to \mathcal{P}(F_X)$, we define the relations $\preceq, \succeq, \prec, \succ, \approx \subset (F_X \times F_X)$ such that

$$
\begin{aligned}
g \preceq f &\Leftrightarrow g \in O_X(f), \\
g \succeq f &\Leftrightarrow f \in O_X(g), \\
g \prec f &\Leftrightarrow f \notin O_X(g) \text{ and } g \in O_X(f), \\
g \succ f &\Leftrightarrow f \in O_X(g) \text{ and } g \notin O_X(f), \\
g \approx f &\Leftrightarrow f \in O_X(g) \text{ and } g \in O_X(f),
\end{aligned}
\tag{2}
$$

for all $f, g \in F_X$.

Relation definitions

Using these relations, we define the traditional related notations $\Omega_X, o_X, \omega_X, \Theta_X : F_X \to \mathcal{P}(F_X)$ such that

$$\begin{aligned} \Omega_X(f) &:= \{g \in F_X : g \succeq f\}, \\ o_X(f) &:= \{g \in F_X : g \prec f\}, \\ \omega_X(f) &:= \{g \in F_X : g \succ f\}, \\ \Theta_X(f) &:= \{g \in F_X : g \approx f\}, \end{aligned} \tag{3}$$

for all $f \in F_X$.

The end. Questions? :)