

Дискретная математика

Отношения порядка

Константин Чухарев

Осень 2026

Частичный порядок

Порядок — это расположение вещей, дающее каждой своё место.

— Августин Блаженный

Что раньше?

Одни вещи предшествуют другим:

- Сборка проекта ждёт компиляции зависимостей.
- Версия библиотеки обязана быть совместима с той, что уже используется.
- Документ в системе контроля версий — наследник предыдущего состояния.

Задача А должна быть выполнена раньше задачи Б.

Это и есть отношение порядка.

В отличие от функции, порядок не обязан сопоставлять каждому элементу единственный образ.

Два элемента могут быть несравнимы.

Частичный порядок

Определение 1 (Частичный порядок)

Бинарное отношение $\preceq$ на A — **частичный порядок**, если оно:

- **Рефлексивно**: $a \preceq a$ для всех $a \in A$.
- **Антисимметрично**: из $a \preceq b$ и $b \preceq a$ следует $a = b$ для всех $a, b \in A$.
- **Транзитивно**: из $a \preceq b$ и $b \preceq c$ следует $a \preceq c$ для всех $a, b, c \in A$.

Пара $(A, \preceq)$ называется **ЧУМ** — частично упорядоченное множество.

Пример

Делимость на $\mathbb{N}$: $a \preceq b$, если $a \mid b$.

2 и 3 несравнимы: ни одно не делит другое.

Строгий порядок

Строгий порядок — аналог $<$.

Определение 2 (Строгий порядок)

Бинарное отношение $\prec$ на A — **строгий порядок**, если оно:

- **Иррефлексивно**: $\neg(a \prec a)$.
- **Асимметрично**: из $a \prec b$ следует $\neg(b \prec a)$.
- **Транзитивно**: из $a \prec b$ и $b \prec c$ следует $a \prec c$.

Пример

- $<$ на числах: $1 < 2$ и $2 < 3$ влечёт $1 < 3$. $1 < 1$ ложно.
- Строгое включение: $\{1\} \subset \{1, 2\}$ — тоже строгий порядок.

Связь с нестрогим порядком

$a \prec b$ тогда и только тогда, когда $a \preceq b$ и $a \neq b$.

Антисимметричность — свойство $\preceq$, асимметричность — свойство $\prec$.

Линейный порядок

Частичный порядок допускает несравнимые пары. Линейный порядок несравнимых пар не допускает.

Определение 3 (Линейный порядок)

Частичный порядок, в котором любая пара сравнима: $\forall a, b : (a \preceq b) \vee (b \preceq a)$.

Линейный порядок (*total order*) — это порядок, в котором всё множество сравнимо и само образует цепь.



$(\mathbb{N}, \leq)$ и $(\mathbb{R}, \leq)$ — линейные, делимость — лишь частичная.

Несравнимость — это независимость

Несравнимость часто воспринимают как недостаток. На деле несравнимость — это информация о независимости.

Если a и b несравнимы — между ними нет зависимости.

Их можно выполнять в любом порядке или параллельно.

Пример

- Задачи «написать отчёт» и «собрать данные» независимы.
- Компилятор собирает несравнимые модули параллельно (флаг -j).

Несравнимость — источник параллелизма.

Порядок в коде: PartialOrd

В Rust частичный порядок выражается трейтом PartialOrd.

Метод `partial_cmp` возвращает `Option<Ordering>`.

Значение `None` — ровно математическая несравнимость.

Трейт `Ord` добавляет тотальность: сравнимы всегда.

Пример

Включение подмножеств — частичный порядок: $\{1\}$ и $\{2\}$ несравнимы.

Поэтому включение выражается через `PartialOrd`, а не через `Ord`.

Диаграммы Хассе

Диаграмма Хассе

Диаграмма Хассе — «скелет» частичного порядка.

Определение 4 (Построение)

1. Убрать все петли (рефлексивность подразумевается).
2. Убрать рёбра, следующие из транзитивности. Остаётся только отношение покрытия.
3. Расположить a выше b , когда $b \prec a$.
4. Нарисовать линии между соседними по покрытию элементами.

Каждое сокращение безопасно.

Убранное восстанавливается по аксиомам порядка.

Покрытие

Определение 5

a покрывает b , если $b \prec a$ и между ними нет промежуточного элемента.

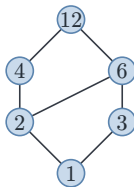
Отношение покрытия — ровно те пары, которые нельзя вывести из транзитивности.

Пример

В делителях числа 12 ребро $1 - 4$ не рисуется.

Хотя $1 \mid 4$ верно, связь следует через 2: $1 \mid 2 \mid 4$.

Делители числа 12



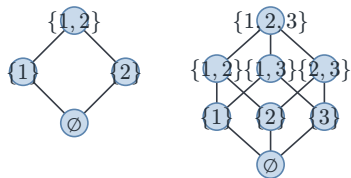
Пример

$D_{12} = \{1, 2, 3, 4, 6, 12\}$ с отношением делимости.

1 — внизу, 12 — вверху.

2 и 3 несравнимы: между ними нет ребра.

Булева решётка



Подмножества $\{1, 2\}$ и $\{1, 2, 3\}$ по включению: слева — квадрат B_2 , справа — куб B_3 .

Ребро диаграммы Хассе — добавить ровно один элемент.

Экстремальные элементы

Минимальные и максимальные

Экстремальные элементы обобщают «первый» и «последний» на случай, когда порядок не линеен.

Определение 6 (Минимальный и максимальный)

m **минимален**, если нет элемента строго меньше:

$$\nexists a \in A : a \prec m.$$

m **максимален**, если нет элемента строго больше:

$$\nexists a \in A : m \prec a.$$

Эти свойства *локальны*.

Минимальный элемент может оставаться несравнимым с остальными.

Наименьший и наибольший

Вторая пара свойств глобальна: элемент сравним со всеми.

Определение 7 (Наименьший и наибольший)

m **наименьший**, если он меньше (нестрого) всех:

$$\forall a \in A : m \preceq a.$$

m **наибольший**, если он больше (нестрого) всех:

$$\forall a \in A : a \preceq m.$$

Пример

В $(\{1, 2, 3\}, \leq)$ наименьший — 1, наибольший — 3.

В ЧУМ $(\{2, 3, 6\}, |)$ наименьшего нет: 2 и 3 несравнимы, хотя наибольший — 6.

Сравнение экстремальных элементов

Понятие	Единственен?	В конечном ЧУМ?	Сравним со всеми?
Минимальный	нет	да	нет
Максимальный	нет	да	нет
Наименьший	да	не гарантирован	да
Наибольший	да	не гарантирован	да

В конечном ЧУМ минимальные и максимальные элементы существуют всегда.

Наименьший и наибольший — не обязательно.

Пример: делители

Рассмотрим $(\{2, 3, 4, 6, 12\}, |)$ — без единицы.

Пример

- Минимальные элементы: 2 и 3.
- Максимальный: 12.
- Наименьшего нет: 2 и 3 несравнимы.
- Наибольший: 12.

Добавление 1 вниз превратит множество в решётку делителей 12 — к этому вернёмся.

**Грани, супремум,
инфимум**

Верхние и нижние грани

Экстремальные элементы характеризуют всё ЧУМ. Грани характеризуют подмножество.

Определение 8 (Грани)

Пусть $S \subseteq A$.

u — **верхняя грань** (*upper bound*) S , если $s \preceq u$ для всех $s \in S$.

l — **нижняя грань** S , если $l \preceq s$ для всех $s \in S$.

Пример

В D_{12} для $S = \{2, 3\}$: верхние грани — 6 и 12, нижние — 1.

Для $S = \{4, 6\}$: верхняя грань — 12, нижние — 1 и 2.

Наилучшие грани

Граней может быть много.

Интересны наилучшие грани: наименьшая среди верхних и наибольшая среди нижних.

Супремум и инфимум

Определение 9 (Супремум и инфимум)

Супремум S — наименьшая верхняя грань, обозначается $\sup S$.

Инфимум S — наибольшая нижняя грань, обозначается $\inf S$.

Для пары $\{a, b\}$ супремум называют join, инфимум — meet.

Пример

В делителях 12 с делимостью для пары $\{2, 3\}$: верхние грани — 6 и 12, нижняя — 1.
Значит, супремум — 6, инфимум — 1.

Супремум и инфимум единственны, когда существуют.

Но в произвольном ЧУМ они могут отсутствовать.

Инфимум без наименьшего

У бесконечного множества наименьшего элемента может не быть.

Пример

$S = \{1, \frac{1}{2}, \frac{1}{3}, \dots\}$ — числа $\frac{1}{n}$ при $n \geq 1$.

Наименьшего элемента нет: всегда есть число меньше.

Но инфимум есть — это 0.

- В $\mathbb{R}$ супремум множества $\{q \in \mathbb{Q} : q^2 < 2\}$ равен $\sqrt{2}$, хотя $\sqrt{2} \notin \mathbb{Q}$ (супремум не обязан принадлежать множеству).
- В самом $\mathbb{Q}$ супремума этого множества не существует вовсе — вот чем полнота $\mathbb{R}$ отличается от плотности $\mathbb{Q}$.

Супремум — не то же самое, что максимум: максимум обязан принадлежать множеству, супремум — лишь наилучшая граница.

Пример: делители 12

В $D_{12} = \{1, 2, 3, 4, 6, 12\}$ возьмём $S = \{2, 3\}$.

Пример

- Верхние грани S : 6 и 12.
- Супремум: 6 — наименьшая из верхних граней.
- Нижние грани S : только 1.
- Инфимум: 1.

Для пары $\{4, 6\}$: верхняя грань — 12, супремум — 12. Нижние грани — 1 и 2, инфимум — 2.

Решётки

Решётка

Супремум и инфимум есть не у всякой пары. Решётка — это ЧУМ, где они есть всегда.

Определение 10 (Решётка)

ЧУМ называется **решёткой**, если у каждой пары элементов существуют и супремум (join), и инфимум (meet).

Пример

- Булеан с включением: join — объединение, meet — пересечение.
- Делители с НОД и НОК: join — НОК, meet — НОД.
- Линейный порядок: join — максимум, meet — минимум.

Решётка не требует линейности.

Несравнимость по-прежнему допускается — но каждая пара имеет точные грани.

ЧУМ, не являющееся решёткой

Пример

$(\{2, 3, 4, 6, 12\}, |)$ — не решётка.

У пары $\{2, 3\}$ нет нижней грани в этом множестве.

Инфимум был бы 1, но единицы в множестве нет.

Добавление 1 превращает множество в D_{12} — решётку делителей 12.

Одного недостающего элемента хватило, чтобы структура «дотянулась» до решётки.

Дистрибутивная решётка и булева алгебра

Определение 11 (Дистрибутивная решётка)

Решётка **дистрибутивна**, если $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ для всех a, b, c . Здесь $\wedge$ — meet, $\vee$ — join.

Определение 12 (Булева алгебра)

Булева алгебра — дистрибутивная решётка с наименьшим 0 , наибольшим 1 и дополнением: для каждого a выполнено $a \wedge \bar{a} = 0$ и $a \vee \bar{a} = 1$.

Логика, множества и булева алгебра — одна дистрибутивная решётка с дополнением в разных записях.

Аксиомы и двойственность — в лекции о булевой алгебре (lec-boolean).

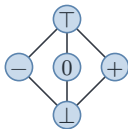
Примеры

Пример

- Булеан: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.
- Делители 12 с НОД и НОК — дистрибутивны.
- Булеан с включением: $0 = \emptyset$, $1 = A$, дополнение — $\bar{S}$.
- Двухэлементная: $\{0, 1\}$ с $\wedge$, $\vee$, $\neg$.

Решётки в коде: знаки

Решётка из статического анализа — знаки чисел.



Каждое значение заменяется знаком: $+$, $-$, 0 .

Слияние двух ветвей условного оператора берёт супремум их знаков.

Если после ветвей знак x — только $+$ или $-$, деление на x безопасно.

Абстрактная интерпретация — свойства программы как элементы решётки.

Топологическая сортировка

Топологическая сортировка

Частичный порядок моделирует зависимости. Чтобы составить расписание, его разворачивают в линейный.

Определение 13 (Топологическая сортировка)

Линейный порядок $\leq$ на ЧУМ, такой что $a \preceq b$ влечёт $a \leq b$, называется **топологической сортировкой**.

Пример

«ОП» $\preceq$ «Алгоритмы» и «Структуры данных». «СД» $\preceq$ «Базы данных».

Допустимый порядок: ОП, СД, Алг, БД.

Несравнимые элементы получают произвольное упорядочение.

Разные сортировки — разные способы разрешить свободу.

Существование

Теорема 1

Любое конечное ЧУМ имеет хотя бы одну топологическую сортировку.

Доказательство

В конечном ЧУМ всегда есть минимальный элемент.

Выбираем минимальный элемент, выводим его и удаляем из множества, затем повторяем шаг для оставшихся элементов.

Когда выводим элемент, все его предшественники уже удалены.

«Снять минимальный» никогда не застревает.

В цикле сортировки нет: у циклического графа топологической сортировки не существует.

Применение

- **Сборка:** если модуль A зависит от B, то B компилируется раньше.
- Несравнимые модули собираются параллельно (флаг -j в Make).
- **Расписания:** метод PERT вычисляет ранние и поздние сроки обходом в топологическом порядке.
- **CI/CD пайплайны** и планировщики задач — та же идея.

Топологическая сортировка — мост от частичного порядка к последовательности действий.

Алгоритм Кана

Алгоритм Кана — жадная реализация той же идеи.

1. Посчитать число входящих рёбер у каждой вершины.
2. Положить в очередь все вершины с нулевым входом.
3. Пока очередь не пуста:
 - Извлечь вершину и вывести её.
 - Уменьшить вход каждого соседа.
 - Если вход соседа стал нулевым, добавить его в очередь.
4. Если вывели не все вершины, в графе есть цикл.

Сложность — $O(V + E)$.

Результат — порядок сборки или отказ при цикле.

Лексикографический порядок

Лексикографический порядок

Определение 14 (На парах)

Пусть порядки заданы на A и B . Тогда на $A \times B$:

$$(a_1, b_1) \prec (a_2, b_2)$$

если $a_1 \prec a_2$, или $a_1 = a_2$ и $b_1 \prec b_2$.

Пример

На $\{0, 1\}^2$: $(0, 0) \prec (0, 1) \prec (1, 0) \prec (1, 1)$.

Первая компонента имеет приоритет.

Так работают словари: сначала по первой букве, потом по второй.

Строки: словарный и shortlex

Со строками разной длины нужен выбор.

Определение 15 (Словарный порядок)

В словарном порядке:

- префикс меньше любой своей непустой надстройкой: $a \prec aa$;
- если ни одна строка не является префиксом другой, порядок задаёт первая позиция, в которой они различаются.

Определение 16 (Shortlex)

Shortlex сначала сравнивает строки по длине, а при равной длине — лексикографически, поэтому $b \prec aa$.

Вполне упорядоченность

Shortlex вполне упорядочен над конечным алфавитом: любое непустое множество строк имеет наименьший элемент.

Поэтому он удобен для перебора строк в алгоритмах.

Лексикографический порядок в коде

Сравнение строк в коде — лексикографическое по умолчанию.

Пример

- `apple < apricot`: первые две буквы совпадают, а `p < r`.
- Сортировка имён, слов, путей — словарный порядок.
- Версии `1.2.0 < 1.2.1 < 1.3.0` — тоже лексикографический.

Когда порядок на компонентах задан, порядок на кортежах строится автоматически.

Приложения отношений порядка

Иерархии типов

Отношение «является подтипом» ($<:$) — частичный порядок на типах.

- Рефлексивно: $T <: T$.
- Транзитивно: если $A <: B$ и $B <: C$, то $A <: C$.
- Антисимметрично: взаимное подтипирование означает один тип.

Если $\text{Dog} <: \text{Animal}$ и $\text{Cat} <: \text{Animal}$, то Dog и Cat несравнимы.

Наименьший общий надтип — супремум в решётке типов.

Векторные часы

В распределённой системе нет глобального «сейчас».

Определение 17 (Произошло-до)

Отношение $\rightarrow$ определяется правилами:

- События в одном процессе упорядочены по времени.
- Отправка сообщения предшествует его получению.
- Замыкаем отношение по транзитивности: если $a \rightarrow b$ и $b \rightarrow c$, то $a \rightarrow c$.

- Это строгий частичный порядок.
- Несравнимые события — **конкурентные**: не связаны причинно.
- Векторные часы реализуют порядок сравнением массивов по компонентам.

Пример: два процесса

Пример

- В процессе 1: e_1 (отправка), затем e_2 (отправка).
- В процессе 2: f_1 (получение e_1), затем f_2 .
- $e_1 \rightarrow f_1$, $f_1 \rightarrow f_2$. e_2 и f_2 конкурентны.

Версионирование

Семантическое версионирование сочетает два порядка на версиях.

- Лексикографический порядок $1.2.0 < 1.2.1 < 1.3.0 < 2.0.0$ — линейный.
- Отношение обратной совместимости — предпорядок (оно рефлексивно и транзитивно, но не обязано быть антисимметричным): $1.2.0$ и $2.0.0$ несравнимы.

Выбор совместимого набора зависимостей — поиск максимальной конфигурации в этом порядке.

Пакетные менеджеры (npm, Cargo, pip) решают именно эту задачу при разрешении зависимостей.

Что мы поняли?

- Частичный порядок допускает несравнимые элементы — это независимость. Линейный всё сравнивает.
- Минимальный не то же, что наименьший. Супремум не то же, что максимум.
- Решётка даёт каждой паре встречу и объединение. Булева алгебра — дистрибутивная решётка с дополнением.
- Порядок — язык структуры: топологическая сортировка, векторные часы, версии пакетов.

Вопросы для самопроверки

- Проверьте, что делимость является частичным порядком на натуральных числах.
- Почему частичный порядок допускает параллелизм, а линейный — нет?
- Чем минимальный элемент отличается от наименьшего? Приведите пример с двумя минимальными и без наименьшего.
- Чем супремум отличается от максимума? Может ли супремум не существовать?
- Является ли множество всех подмножеств $\{1, 2, 3\}$ с включением решёткой? Дистрибутивной?
- Докажите, что в конечном ЧУМ топологическая сортировка существует всегда.
- Сравните строки `apple` и `apricot` в словарном порядке и в `shortlex`.