Релацията е множество от n-торки, които са подмножество на декартовото произведение.

Бинарна релация- Ако X и Y са две множества, то **X ↔ Y** е множество на всички бинарни релации между X и Y т.е.

$X \leftrightarrow Y == P (X \times Y)$

Видове релации: Бинарни, хомогенни и хетерогенни, рефлексивни, симетрични, транзитивни.

Област (domain), Source на дадената релация R е множество от елементи от X, които са свързани с елементи от Y

Обхват (range), **Target** на дадената релация R е множество от елементи Y, с които елемент от X е свързан

Част от областта (domain): Ако $A \subseteq X$, то $A \lhd R$

2) Част от обхвата (range): Ако $B \subseteq Y$, то $R \lhd B$

3) Изваждане от областта (X \ A) или $A ⩤ R$ - всички елементи от областта X, които не принадлежат на множеството A

4) Образ на A в релацията R (| A |) = ran($A ⩤ R$)

5) Обратна релация R~

Композиция (Composition) - Ако **източникът (source) на релация R2 е цел (target) на друга релация R1**, то двете релации могат да формират нов обект, наречен **композиция на две релации (R1°R2)**. Знакът ○ , както и ⨾ , се използват за означаване на композиция.

$x \mapsto z \in R1 \circ R2 \Leftrightarrow \exists y : Y \bullet x \mapsto y \in R1 \land y \mapsto z \in R2$

Композицията на две релации свързва началния елемент на първата релация с крайния елемент на втората релация, ако съществува междинен елемент, който ги свързва последователно.

Функции - е специален вид релация, при която елемент от едно множество е свързан с най-много един елемент от друго множество

Функцията е такава релация, която няма двойки, съдържащи еднакъв първи елемент.

Видове функции: Тотални, частични, инекции, сюрекции, биекции.

f е функция такава, че за **всяко x ∈ A** съществува **едно y ∈ B** f е **тотална функция**, която се записва **A→B**

f е функция такава, че за **всяко x ∈ A** съществува **най-много едно y ∈ B** f е **частична функция**, която се записва A ⇸ B (there may be elements of A that are not related to any element of B)

f е функция, дефинирана за крайно множество от стойности на A f е крайна функция, която се записва A⇻ B

“one-to-one” – инекция - X ↣Y (покрива всички елементи на X)

f е “onto” или сюрекция и X ↠ Y (покрива всички елементи на Y)

f е едновременно one-to-one и onto - биекция - X⤖ Y

Ламбда нотация (λ декларация | ограничение ● резултат), където “резултат” е математически израз:

f = (λ x: T ● израз)

f = (λ x: Z ● x^2)

Тъй като релациите са множества (от наредени двойки), то върху тях можем да приложим операторите за множества.

R1 = {(1,red), (2,blue)}

R2 = {(3,green), (2,blue)}

R1 U R2 = {(1,red), (2,blue), (3,green)}

R1 ∩ R2 = {(2,blue)}

(R1 U R2) = 3 - брой елементи в обединението R1 U R2

Отменяне (overriding) Често се налага да сменим стойността на функцията за една или повече стойности на областта

Това означава, че новата релация f⨁g ще приема:

- стойностите на функцията f, където f не е дефинирана в g,
- и стойностите на функцията g, където g е дефинирана

- Отменянето се отнася **само** до стойностите на областта на функциите.

- Операторът ⨁ е приложим само към функции от **един и същи тип**.

оператор "**между**" - дефиниране на крайни множества**2 .. 5 = {2,3,4,5}**

Редици: Подредени множества, които се различават от множествата с фиксирана дължина и **позволяват дублиране**, като всяка **позиция е важна.** Празна редица: ⟨ ⟩

Множествата не са редици. Множествата се различават от редиците, защото:

- Множествата **нямат определен ред на елементите**

- Множествата **не позволяват дублиране** на елементи, докато в редицата може да има повтарящи се елементи.
  AskedQns == ⟨ Rob, Peter, Mark, Mark, Matt ⟩
  head (AskedQns) = Rob
  tail (AskedQns) = ⟨ Peter, Mark, Mark, Matt ⟩
  front (AskedQns) = ⟨ Rob, Peter, Mark, Mark ⟩
  last (AskedQns) = Matt
  # AskedQns = 5

Свързване (concatenation) ⟨ 1,3,1 ⟩ ^ ⟨ 3,4 ⟩ = ⟨ 1,3,1,3,4 ⟩

Обобщение - distributed concatenation (flattening) - обединяваме вложени структури в една подредена редица, премахвайки вложеността. ∧/⟨⟨a,b,c⟩,⟨d,e⟩,⟨f,g,h⟩⟩

Достъпът до отделните елементи позволява проверка на стойността на елемент на дадена позиция.

Филтърът премахва нежеланите елементи от структурата, като същевременно запазва реда и мултипликативността на др.елементи.

инективни редици – редици, в които няма повтарящи се елементи

композиция на редица с функция:

-s е редица върху тип X, т.е. **s: seq[X]**, което означава, че s е последователност от елементи от типа X.

-f е функция върху елементите на X, т.е. **f: X → Y**, което означава, че f приема елементи от тип X и връща елементи от тип Y.

Композицията s ○ f означава, че функцията f се прилага към всеки елемент от редицата s. Резултатът от композицията е нова редица, в която на всяка позиция е поставен резултатът от прилагането на f към съответния елемент от s.

Дистрибутивност - Ако имаме редица **s** и функция **f**, свойството на дистрибутивност казва, че ако приложим **f** към редицата, можем да постигнем същия резултат, ако приложим **f** към всеки елемент от редицата поотделно и след това комбинираме тези резултати.

$s \circ (f \circ g) = (s \circ f) \circ (s \circ g)$

Трасета (Traces) са поредици от събития (или операции), които се случват в система; последователност от събития

Event- съвкупност от събития

Ограничения на трасетата: trace ↾ event - извличане на специфични събития като записи или четения

Bags (Чанти) - В теорията на множества bag (или multiset) е съвкупност от елементи, в която същият елемент може да се появява повече от веднъж. Позволяват многократно появяване на същите елементи

Free Types (Свободни типове) - набор от възможни стойности, които могат да бъдат използвани за създаване на типове

colors::=red|orange|yellow|green|blue|indigo|violet - дефинира colors като тип, който може да има една от следните стойности: червен...

Индукция - Използва се за доказателства върху числа, структури като бинарни дървета и редици.

Индукцията е метод за доказателство, който се използва за показване на истинността на дадено твърдение за цяла категория обекти, базирайки се на два основни принципа:

1. **База на индукцията** -Основен случай (**Base Case**):Доказва се, че твърдението е вярно за най-малкия обект от категорията.

Пр: За естествени числа, осн. случай обикновено е за 0 или 1.

2. **Индуктивна стъпка (Inductive Step)**: - ако твърдението е вярно за някакъв обект n, то е вярно и за следващия обект $n + 1$

Видове индукция:

Натурална индукция (Mathematical Induction) - за естествени числа.

Структурна индукция (Structural Induction) - Прилага се към обекти с рекурсивна структура, като дървета, списъци или редици.

Техниките за доказателство:

1. Natural Deduction (Естествено извод) -Това е метод за доказване, който използва **правила за извод** (или логически правила), за да извежда нови твърдения от вече доказани (конюнкция, импликация, дизюнкция и др.). Базира се на **формална логика** и позволява да извеждаме нови твърдения, започвайки от осн. предпоставки.

2. Equational Reasoning (Еквивалентно разсъждение) -Тази техника се основава на принципа на **еквивалентност**. В нея се доказват твърдения чрез преработка на изрази, като се използват **логически еквивалентни формули**. В този процес може да се използват **алгебрични правила** за манипулиране на уравнения или изрази, за да се покаже, че два израза са идентични или че едно твърдение води до друго.

3. Induction (Индукция) -Индукцията е метод за доказване на твърдения, които са верни за всички елементи от дадено множество.

4. Special Forms (Специални форми на доказателства):

- **Case analysis (Анализ на случаи)**: Това е метод, при който разделяме доказателството на различни случаи и доказваме твърдението поотделно за всеки от тези случаи. След това обединяваме резултатите, за да направим обобщение.
- **One-point rule (Правило за една точка)** - доказва се твърдението чрез анализ на един конкретен случай, който е достатъчен за доказване на общото твърдение.
- **Proof by contradiction (Доказателство чрез противоречие)**: Тази техника включва приемането на отрицанието на твърдението и доказването, че това води до противоречие с известни факти или аксиоми. Ако достигнем противоречие, то твърдението трябва да е вярно, тъй като неговото отрицание води до нелогичност.