

Модели на софтуерни системи

доц. Олга Георгиева

СУ ФМИ

катедра “Софтуерни технологии”

Модели на софтуерни системи

Лекция 4: Релация, композиция, функция, редици. Индукция

Олга Георгиева

СУ, Факултет по Математика и информатика,
Катедра СТ

Релации – дефиниция, описание, класификация

- Често в една формална спецификация е необходимо да се опише **релацията** м-ду обекти т.е. връзката (отношението) между обектите.

Пр.:

$A == \{ (1,1), (1,2), (2,2) \}$

$B == \{ (2, \text{red}), (5, \text{blue}), (3, \text{red}) \}$

$C == \{ (\text{David}, \text{Jun } 1), (\text{Mary}, \text{Aug } 2), (\text{Bill}, \text{Feb } 5) \}$

Дефиниция: Релацията е множество от наредени n -торки, които са подмножество на Декартовото произведение.

- Въпроси:**
 - как се дефинира релация?
 - как да се извлече информация от съществуваща релация?
 - как релациите могат да се обръщат или композират, за да дефинират нови обекти?

Релации – дефиниция, описание, класификация (2)

Дефиниция за Бинарна релация:

Ако X и Y са две множества, то $X \leftrightarrow Y$ е множество на всички бинарни релации между X и Y т.е.

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

Пр.:

A: $N \leftrightarrow N$ B: $N \leftrightarrow \text{Color}$ C: $\text{Person} \leftrightarrow \text{Date}$

- Нотация за двойка (“maps to”):

а) “ \mapsto ”

б) $\{(2,\text{red}), (2,\text{blue}), (3,\text{red})\} = \{2 \not\in \text{red}, 2 \not\in \text{blue}, 3 \not\in \text{red}\}$

Релации – дефиниция, описание, класификация (3)

Видове релации:

- хомогенни и хетерогенни;
- рефлексивни, симетрични, транзитивни
- **бинарни релации**

Пр.

$$id\ X == \{x : X \bullet x \mapsto x\}$$

R е рефлексивна, ако

$$Reflexive\ [X] == \{R : X \leftrightarrow X \mid id\ X \subseteq R\}$$

$$\text{защото } \forall x : X \bullet x \mapsto x \in R$$

R е симетрична, ако

$$Symmetric[X] == \{R : X \leftrightarrow X \mid \forall x, y : X \bullet x \mapsto y \in R \Rightarrow y \mapsto x \in R\}$$

Example 7.2 The relation *drives* is used to record which makes of car are driven by the members of a small group of people. If the group of people is defined by

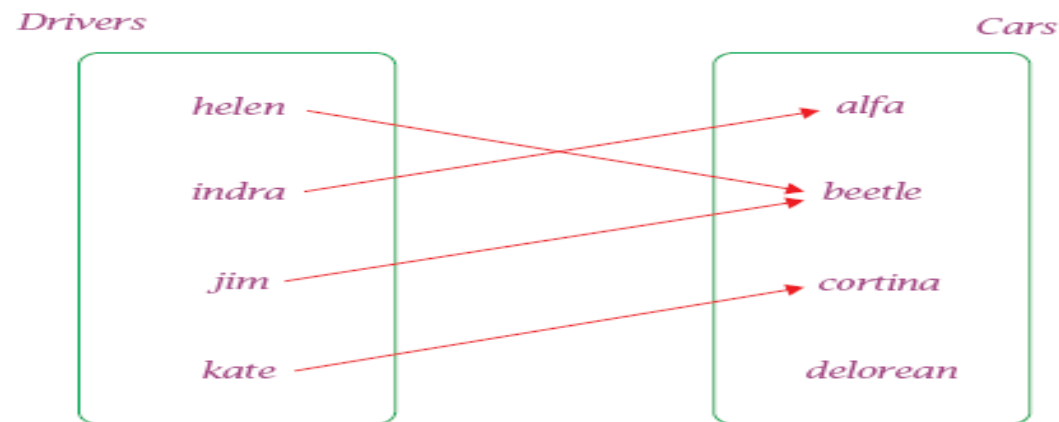
Drivers == { *helen*, *indra*, *jim*, *kate* }

and the choice of cars is defined by

Cars == { *alfa*, *beetle*, *cortina*, *delorean* }

then *drives* is an element of *Drivers* \leftrightarrow *Cars*, and the statement 'Kate drives a cortina' could be formalised as *kate* \mapsto *cortina* \in *drives*. \square

Алтернативно:



Example 7.3 The relation *drives* could be defined by

drives : *Drivers* \leftrightarrow *Cars*

drives = { *helen* \mapsto *beetle*, *indra* \mapsto *alfa*, *jim* \mapsto *beetle*, *kate* \mapsto *cortina* }

That is, Helen and Jim drive Beetles, Indra drives an Alfa, Kate drives a Cortina, and nobody drives a DeLorean. \square

Релации: Област и обхват

Ако R е релация от типа $X \leftrightarrow Y$, то:

1. **Област (domain)** на дадената релация R е множество от елементи от X , които са свързани по някакъв начин с елементи от Y :

$$\text{dom}R = \{x : X; y : Y \mid x \mapsto y \in R \bullet x\}$$

Source

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

$$\text{ran}R = \{x : X; y : Y \mid x \mapsto y \in R \bullet y\}$$

Target

Пр.:

$A == \{ (1,1), (1,2), (2,2) \}$ $\text{dom } A = \{ 1, 2 \}$ and $\text{ran } A = \{ 1, 2 \}$

$B == \{ (2, \text{red}), (5, \text{blue}), (3, \text{red}) \}$ $\text{dom } B = \{ 2,3,5 \}$ and $\text{ran } B = \{ \text{red}, \text{blue} \}$

$C == \{ (\text{David}, \text{Jun } 1), (\text{Mary}, \text{Aug } 2), (\text{Bill}, \text{Feb } 5) \}$

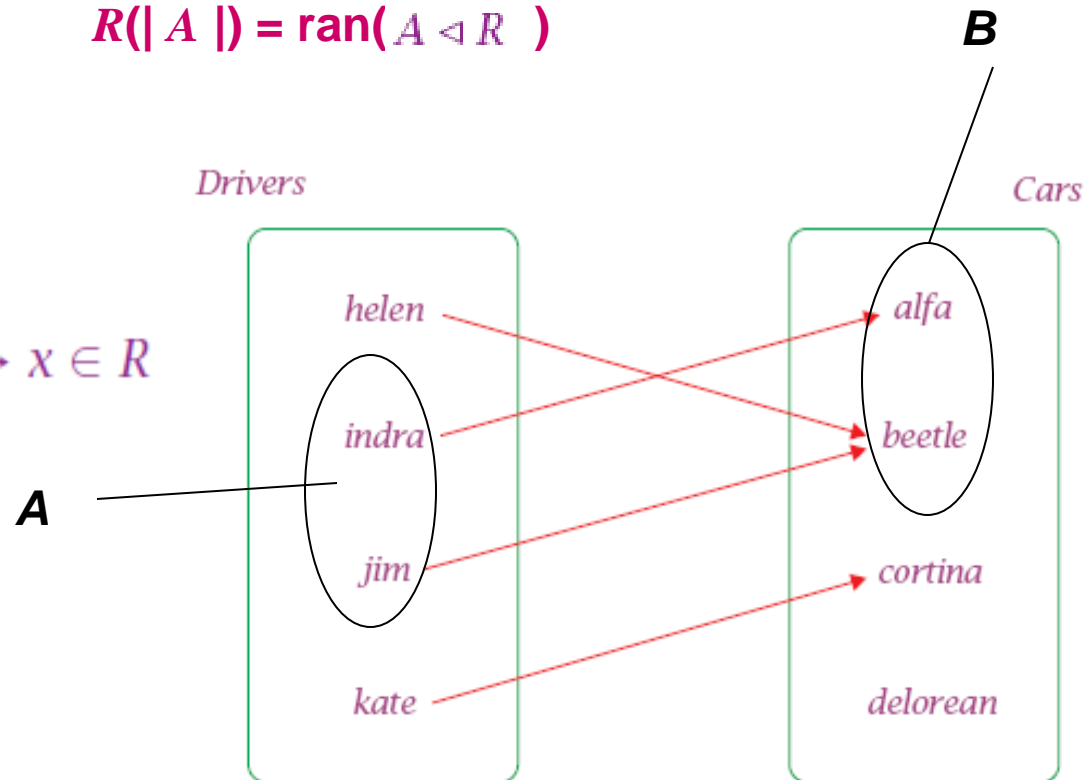
$\text{dom } C = \{ \text{David}, \text{Mary}, \text{Bill} \}$ $\text{ran } C = \{ \text{Jun } 1, \text{Aug } 2, \text{Feb } 5 \}$

Допълнителни определения

- 1) **Част от областта (domain):** Ако $A \subseteq X$, то $A \triangleleft R$
- 2) **Част от обхвата (range):** Ако $B \subseteq Y$, то $R \triangleright B$
- 3) **Изваждане от областта ($X \setminus A$), изваждане от обхвата ($Y \setminus B$) или $A \triangleleft R$**
- 4) **Образ на A в релацията R** $R(| A |) = \text{ran}(A \triangleleft R)$
- 5) **Обратна релация R^\sim**

Описание на оператора:

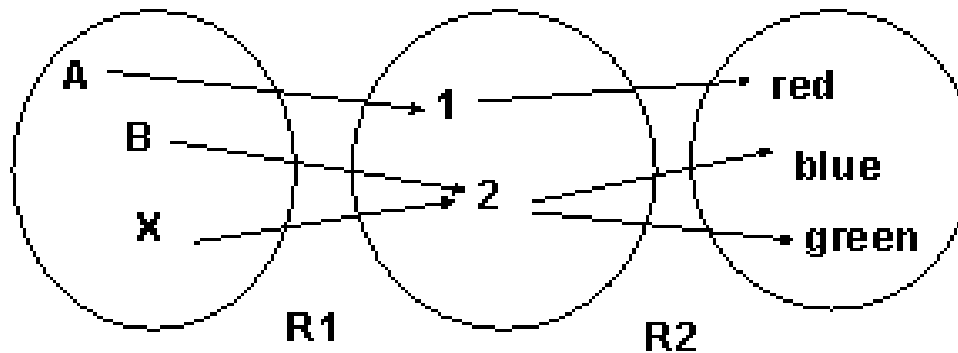
$$\forall x:X; y:Y \bullet x \mapsto y \in R^\sim \Rightarrow y \mapsto x \in R$$



Композиция (Composition)

Дефиниция:

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** \circ **R2**). Знакът \circ , както и \circ , се използват за означаване на композиция..



Пр.: Два елемента **x** и **z** са свързани чрез композицията $R1 \circ R2$, ако съществува междинен елемент **y** такъв, че **x** е свързан с **y** и **y** е свързан със **z**.

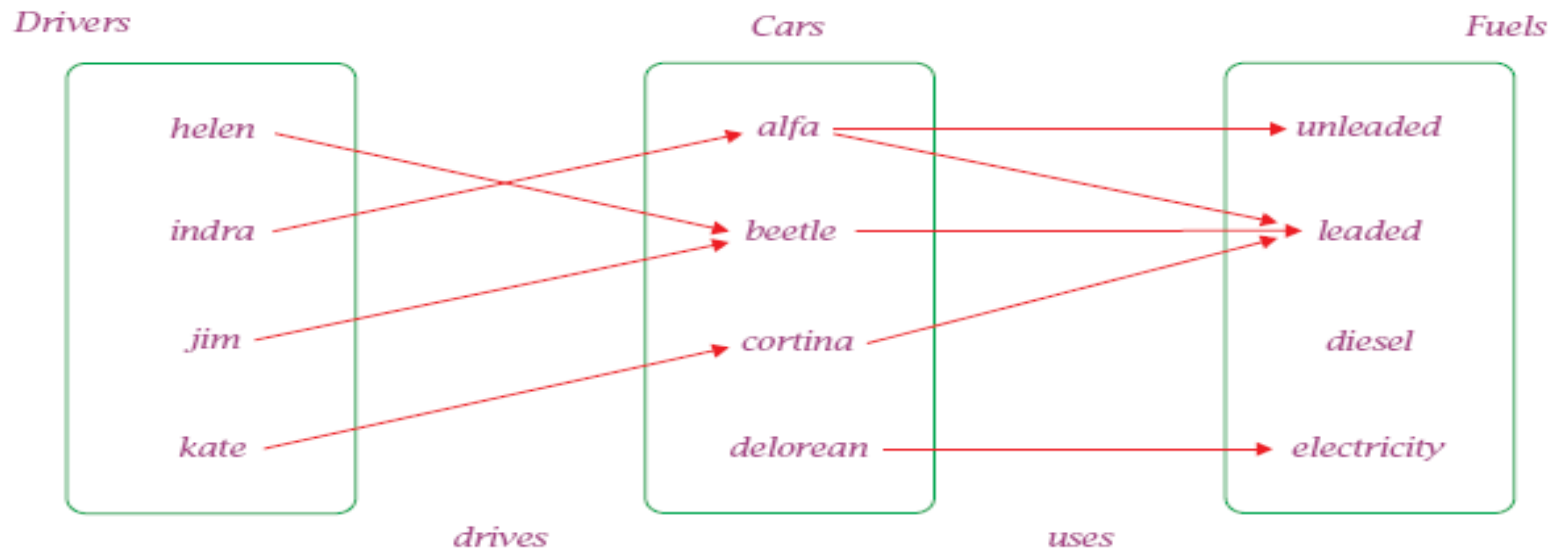
$$x \mapsto z \in R1 \circ R2 \Leftrightarrow \exists y: Y \bullet x \mapsto y \in R1 \wedge y \mapsto z \in R2$$

Пример: The relation *uses* of type *Cars* \leftrightarrow *Fuels* tells us which fuel is used by each of the cars in .

uses : *Cars* \leftrightarrow *Fuels*

uses = { *alfa* \mapsto *unleaded*, *alfa* \mapsto *leaded*, *beetle* \mapsto *leaded*,
cortina \mapsto *leaded*, *delorean* \mapsto *electricity* }

We may compose the relations *drives* and *uses* to find out which fuels a driver may purchase. If *buys* = *drives* \circ *uses* then *buys* is a relation of type *Drivers* \leftrightarrow *Fuels* such that



Which fuel to buy?

Функции (Functions)

Дефиниция:

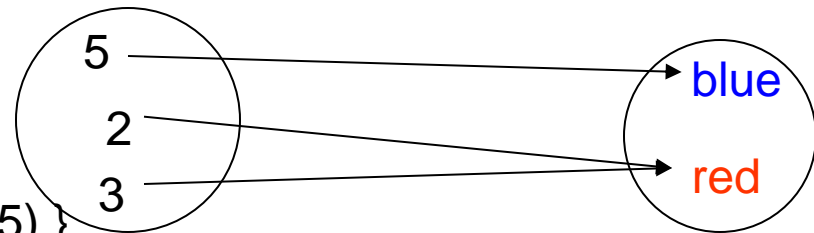
Функцията е *специален* вид релация, при която елемент от едно множество е свързан с **най-много един** елемент от друго множество.

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

Пр.:

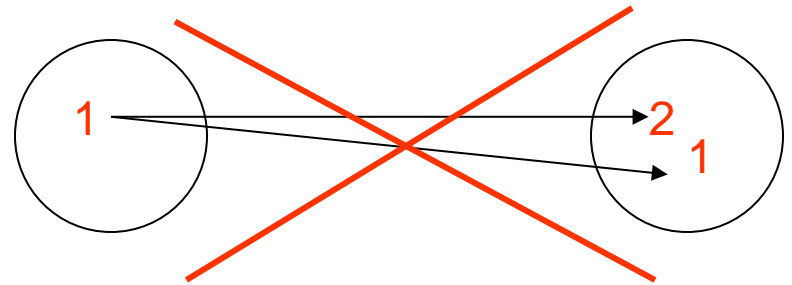
$B == \{ (2, \text{red}), (5, \text{blue}), (3, \text{red}) \}$

$C == \{ (\text{David}, \text{Jun } 1), \{ \text{Mary}, \text{Aug } 2 \}, \{ \text{Bill}, \text{Feb } 5 \} \}$



Но НЕ !:

$A == \{ (1,1), (1,2), (2,2) \}$



• Въпроси:

- класификация на функциите;
- нотацията за функция и приложенията и';
- свойства на функции;

Специални случаи -1

Нека $f: A \rightarrow B$

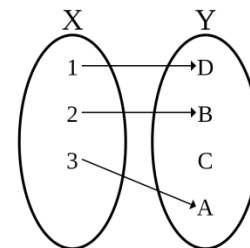
1. f е функция такава, **че за всяко $x \in A$ съществува точно едно $y \in B$**
 f е **тотална** функция, която се записва $A \rightarrow B$
2. f е функция такава, **че за всяко $x \in A$ съществува най-много едно $y \in B$**
 f е **частична** функция, която се записва $A \rightarrowtail B$
3. f е функция, дефинирана за **крайно множество** от стойности на A
 f е **крайна** функция, която се записва $A \dashrightarrow B$

Специални случаи - 2

Нека $f: X \rightarrow Y$

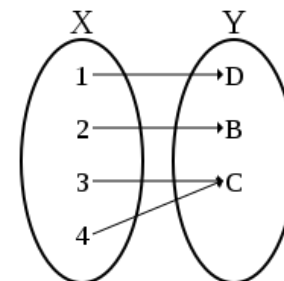
4. f е функция, за която всеки елемент от $\text{dom}(f)$ се асоциира с не повече от един елемент на $\text{ran}(f)$, то

f е “one-to-one” или инекция и $X \rightarrowtail Y$



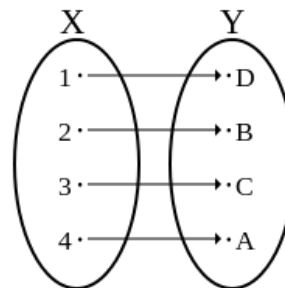
5. f е функция, за която всеки елемент от $\text{ran}(f)$ се асоциира с поне един елемент на $\text{dom}(f)$

f е “onto” или сюрекция и $X \twoheadrightarrow Y$



6. f е едновременно one-to-one и onto

f е биекция и $X \xrightarrow{\sim} Y$



Специални случаи

Релации \leftrightarrow

Частични функции \dashrightarrow

Инекции

\leftrightarrow

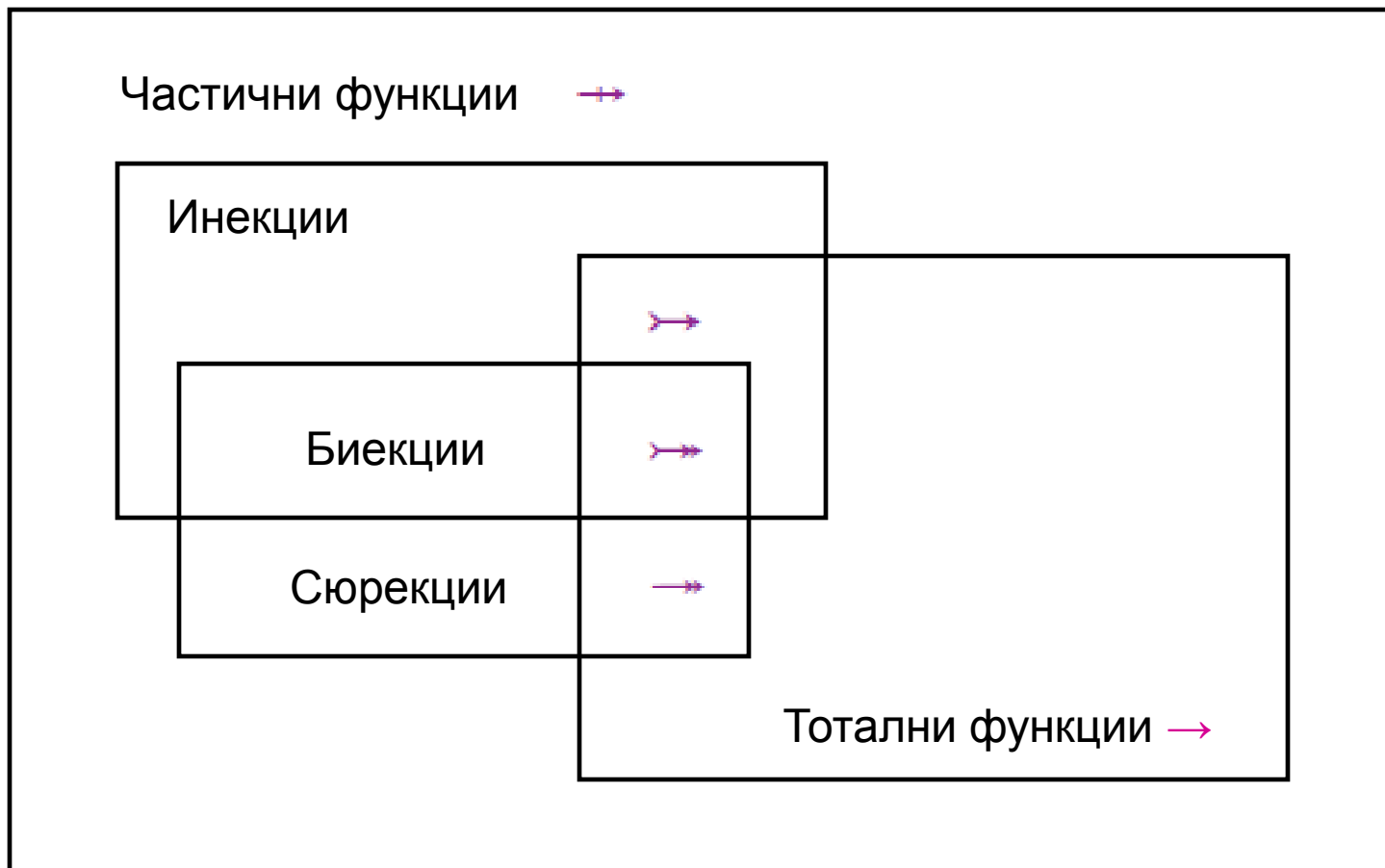
Биекции

\leftrightarrow

Сюрекции

\dashrightarrow

Тотални функции \rightarrow



Дефиниране на функции (1) – Z нотация

- Формализация:

$$\frac{f: A \rightarrow B}{\text{predicate over } f}$$

- Функцията може да бъде дефинирана чрез правило за изчисляване на нейните стойности

Пр.: За да характеризираме “квадратична функция” пишем (неформално):

$$f(x) = x^2$$

или формално:

$$\frac{f: \mathbb{Z} \rightarrow \mathbb{Z}}{\forall x: \mathbb{Z} \bullet f(x) = x^2}$$

Каква е тази функция?

Разсъждения относно функции – Z нотация

Ако a е елемент от dom на функцията f , то записът $f(a)$ означава единственият елемент, който е резултат от приложението на функцията върху a .

- Съществуват две правила за извод, свързани с приложението на функции:

- 1) Ако \exists единствена двойка $a \mapsto b \in f$ с първи елемент a и b – втори елемент, то $b = f(a)$:

$$\frac{\exists_1 p : f \bullet p.1 = a \quad a \mapsto b \in f}{b = f(a)} \text{ [app-intro]}$$

- 2) Ако $b = f(a)$ и \exists единствена двойка с първи елемент a , то $a \mapsto b \in f$:

$$\frac{\exists_1 p : f \bullet p.1 = a \quad b = f(a)}{a \mapsto b \in f} \text{ [app-elim]}$$

Други начини за дефиниране на функции (2) - Ламбда нотация

Ламбда нотация (*Lambda notation*)

(λ декларация | ограничение • резултат)

където “резултат” е математически израз.

- Обща форма:

$f = (\lambda x: T \bullet \text{израз})$

Пр.: Квадратична функция

$f = (\lambda x: Z \bullet x^2)$

- Може да използваме повече от една променлива, включително и от различни източници, за да дефинираме многопроменливи функции:

Пр.: $lus = (\lambda x,y: Z \bullet x+y)$

Релациите и функциите като множества

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

Пр.:

$$R1 = \{(1, \text{red}), (2, \text{blue})\}$$

$$R2 = \{(3, \text{green}), (2, \text{blue})\}$$

$$R1 \cup R2 = ?$$

$$R1 \cap R2 = ?$$

$$\# (R1 \cup R2) = ?$$

?: Функция ли е обединението и сечението на две функции?

Кога?

Отменяне (overriding)

Често се налага да сменим стойността на функцията за една или повече стойности на областта.

Ако f и g са функции от един и същи тип, то $f \oplus g$ е релационно отменяне на f с g . Това е релация, която приема f навсякъде, където f е извън дефиниционната област на g и приема g , където g е дефинирана.

Пр.:

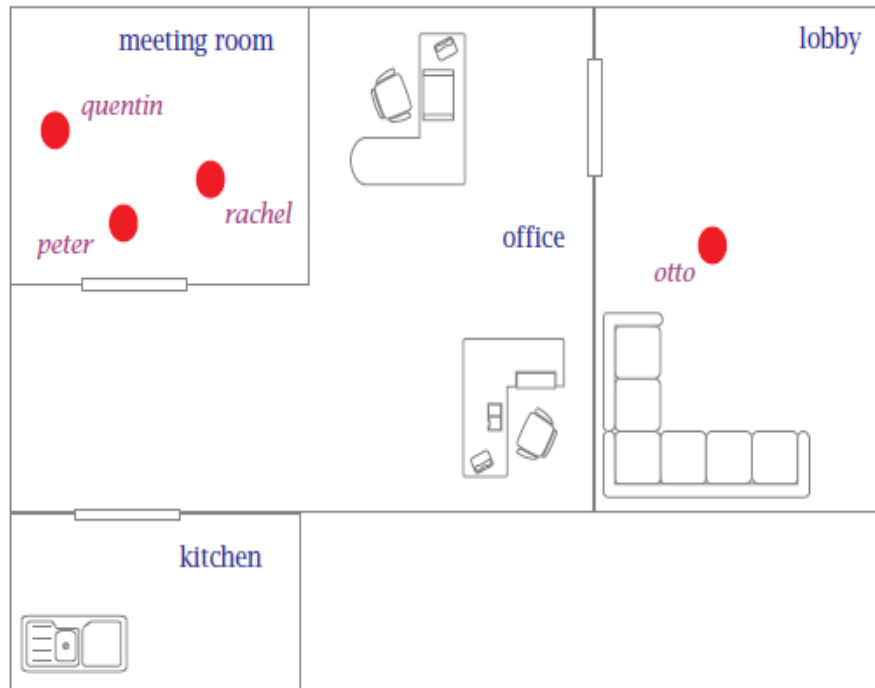
$f == \{(1, \text{red}), (2, \text{blue}), (3, \text{green})\}$

$g == \{(1, \text{pink}), (4, \text{white})\}$

$f \oplus g = \{(1, \text{pink}), (2, \text{blue}), (3, \text{green}), (4, \text{white})\}$

- Ограничения:**
1. Отменянето се отнася само до стойностите на областта.
 2. Операторът е приложим и към функции от един и същи тип!

Пример: Staff location system



An organisation has a system for keeping track of its employees while they are on the premises. Each employee is issued with an active Badge which reports their current position to a central database.

If the set of all people is *Person*, and the set of all locations is *Location*, then the information provided by the system may be described by a relation *where is* of type *Person* \leftrightarrow *Location*. It is impossible for an employee to be in two places at once, so this relation will be a **partial function**:

$$\text{where_is} \in \text{Person} \rightarrow \text{Location}$$

If Rachel is an employee, then we may write *where_is Rachel* as a function to denote her current location.

$$\{\text{otto} \mapsto \text{lobby}, \text{peter} \mapsto \text{meeting}, \text{quentin} \mapsto \text{meeting}, \text{rachel} \mapsto \text{meeting}\}$$

and

$$\text{where_is rachel} = \text{meeting}$$

Additionally

$$\text{where_now} = \text{where_is} \oplus \text{update}$$

Крайни множества

Съществува библиотека от помощни дефиниции. Например:

- оператор “**между**” за дефиниране **на крайни множества**.

Пр.: $2 \dots 5 = \{2,3,4,5\}$

$$\begin{array}{|l} \dots : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{P}\mathbb{N} \\ \hline \forall m, n : \mathbb{N} \bullet m \dots n = \{i : \mathbb{N} \mid m \leq i \leq n\} \end{array}$$

Множество от всички крайни подмножества на множеството X :

$$\mathbb{F}X == \{s : \mathbb{P}X \mid \exists n : \mathbb{N} \bullet \exists f : 1 \dots n \rightarrow s \bullet \text{true}\}$$

If X is a finite set, then $\mathbb{F}X$ and $\mathbb{P}X$ are equal.

Обобщение на операторите, дефиниращи множества

Нека S, R, T са множества:

$S \cap R$ (intersection), \cup (union), \setminus (difference)
 $P S$ (powerset)
 $\{x: S \mid P(x)\}$ (set comprehension)
 $\{x: S \mid P(x) * E\}$ (generalized set comprehension)
 $S \times R$ (cartesian product)
 $S \leftrightarrow R$ (relations)
 $S \rightarrow R$ (functions), etc.
 \circ (composition)
 \oplus (overwriting) \equiv overriding

Вижте пълния списък на символи на Z нотацията на MOODLE страницата на курса.

Редици (sequences) и индукция

В тази лекция **приключихме** разглеждането на множествата, релациите и функциите, както и въпросите:

- Релационната композиция

- Релациите & функциите като множества

- Използването на релациите за моделиране на декларации

- Ламбда нотация

Предстои: специални видове релации:

- Редици* и др.

- Операции* върху редици

Накрая: идеята за доказателство чрез *индукция*

- Върху целите числа

- Върху други структури

Редици

Дефиниция:

Редицата е подредена сбирка (съвкупност) от обекти.

- Множествата не са редици, защото техните елементи не са подредени и не позволяват дублиране.
- n -торките са подредени, но имат фиксирана дължина.

Редицата X , отбелязвана с **$seq[X]$** , е функция (изображение) на положителните цели числа $1 \dots N$ към елементите на X .

Пр.:

Queue = $\langle \text{Rob, Peter, Mark, Mark, Matt} \rangle$

$seq[\text{Queue}] = \{(1, \text{Rob}), (2, \text{Peter}) \dots\} =$

$= \{1 \mapsto \text{Rob}, 2 \mapsto \text{Peter}, \dots\}$

– **Празна редица:** $\langle \rangle$

Операции с редици

- Нека

AskedQns == $\langle \text{Rob, Peter, Mark, Mark, Matt} \rangle$

тогава

head (AskedQns) = Rob (елемент)

tail (AskedQns) = $\langle \text{Peter, Mark, Mark, Matt} \rangle$ (редица)

front (AskedQns) = $\langle \text{Rob, Peter, Mark, Mark} \rangle$

last (AskedQns) = Matt

AskedQns = ?

- **Свързване (*concatenation*)** – нова редица !

$\langle 1, 3, 1 \rangle \wedge \langle 3, 4 \rangle = \langle 1, 3, 1, 3, 4 \rangle$

Пр.: AskedQns = head(AskedQns) \wedge tail(AskedQns)

- **Обобщение - *distributed concatenation (flattening)*:**

$\wedge / \langle \langle a, b, c \rangle, \langle d, e \rangle, \langle f, g, h \rangle \rangle$

Пример: Операции с редици

The ticket office in a railway station has a choice of two counters at which tickets may be purchased. There are two queues of people, one at each counter; these may be modelled as sequences:

$queue_a = \langle sally, tim, ulla \rangle$
 $queue_b = \langle vicky, wilson, xavier \rangle$

Sally and *Vicky* are at the head of their respective queues, but - just as *Vicky* is about to be served the ticket machine at Counter **b** breaks down, and the people waiting there join the end of other queue. Order is maintained, so the result is given by

$queue_a \frown queue_b,$

Or the sequence:

$\langle sally, tim, ulla, vicky, wilson, xavier \rangle$

Операции с редици

- Достъп до отделните елементи:

AskedQns(1) = Rob или AskedQns 1 = Rob
AskedQns(3) = Mark или AskedQns 3 = Mark

↗ **Защо ?**

- Филтър – запазва реда и мултипликативността на елементите

$$\langle a, b, c, d, e, d, c, b, a \rangle \upharpoonright \{a, d\} = \langle a, d, d, a \rangle$$

- Пр.

trains == $\langle (10.15, \text{london}), (10.38, \text{edinburgh}), (10.40, \text{london}),$
 $(11.15, \text{birmingham}), (11.20, \text{reading}), (11.40, \text{london}) \rangle$

We are interested only in those trains that are going to London

$$\textit{trains} \upharpoonright \{ t : \textit{Time} \bullet (t, \textit{london}) \}$$

$\langle (10.15, \textit{london}), (10.40, \textit{london}), (11.40, \textit{london}) \rangle$

... the first train to London: ...

time	from	to
10 15	OXFORD	LONDON PADDINGTON
10 38	LONDON PADDINGTON	EDINBURGH
10 40	GREAT MALVERN	LONDON PADDINGTON
11 15	MANCHESTER	POOLE
11 20	OXFORD	READING

Формална основа за дефиниране на редиците

- Редицата е функция:

Пр.1 Крайна редица, дефинирана като крайна функция

If X is a set, then the set of all finite sequences of objects from X is defined by the following abbreviation:

$$\text{seq } X == \{s : \mathbb{N} \multimap X \mid \exists n : \mathbb{N} \bullet \text{dom } s = 1 \dots n\}$$

This definition makes explicit an assumption about sequences: that every element of a given sequence must share the same type.

Пр 2. *trains 2* = ?

- Ограничаваме се до редици с крайна дължина.
- Специални редици: *инективни редици* – редици, в които няма повтарящи се елементи

$$\text{iseq } X == \{s : \text{seq } X \mid s \in \mathbb{N} \rightarrow X\}$$

Пр. ?

Функции върху редици (2)

Дефиниране на функции върху редици:

Въвежда се операция f чрез описание на ефекта ѝ върху празна редица и след това и върху редица, започвайки от свободно избран елемент от редицата (рекурсивен принцип):

$$f\langle \rangle = k$$

$$f(\langle x \rangle \frown s) = g(\langle x \rangle, f(s))$$

•Пр.:

The function '*reverse*' returns a sequence in which the elements appear in reverse order. The two equations

$$\text{reverse}\langle \rangle = \langle \rangle \quad (\text{reverse.1})$$

$$\text{reverse}(\langle x \rangle \frown s) = (\text{reverse } s) \frown \langle x \rangle \quad (\text{reverse.2})$$

are enough to describe the effect of '*reverse*' upon any finite sequence.

• Общо свойство на функциите върху редици: *Дистрибутивност*

$$f(s \frown t) = (f s) \frown (f t)$$

Трасета и ограничения

- **Traces**

$\text{ReadEvents} == \{n: N \bullet \text{read}.n\}$

$= \{\text{read}.1, \text{read}.2, \text{read}.3 \dots\}$

$\text{WriteEvents} == \{n: N \bullet \text{write}.n\}$

$\text{Event} == \text{ReadEvents} \dots \text{WriteEvents}$

$\text{trace}: \text{seq}[\text{Event}]$

$\text{trace} = \langle \text{write}.7, \text{write}.3, \text{read}.3, \text{write}.20, \text{read}.20, \text{read}.20 \rangle$

Ограничения:

$\text{trace} \circ \text{WriteEvents} = \langle \text{write}.7, \text{write}.3, \text{write}.20 \rangle$

$\text{trace} \circ \text{ReadEvents} = \langle \text{read}.3, \text{read}.20, \text{read}.20 \rangle$

Пр.: справки на моб. телефон, ел. поща

- **Bags**

$[a, a, b, b, b, c, c]$

- **Free types**

- $\text{colours} ::= \text{red} \mid \text{orange} \mid \text{yellow} \mid \text{green} \mid \text{blue} \mid \text{indigo} \mid \text{violet}$

Индукция

Техники за *доказване* на твърдения от вида:

$$\forall x : S \bullet P(x)$$

1. Показва се, че твърдението е валидно за всички стойности чрез изброяване на всички случаи (not promising for infinite sets)
2. Приема, че не е вярно и се стига до противоречие.
3. Показва, че $P(i)$ е валидно за произволно " i ", след което се прилага S – въведение (introduction);

- Алтернатива: **Принцип на индукцията** за *естествените числа*:

Основен случай: Показваме истинността на твърдението за $n = 0$

Индуктивна стъпка: Показваме, че ако твърдението е вярно за $n = k$, то е вярно и за $n = k + 1$ т.е. е вярно за всички естествени числа

Структурна индукция

- Структурна индукция (Induction beyond natural numbers):
 - > Покажи за основен случай(и)
 - > Покажи, че може да се докаже истинност за кой да е обект. Ако се знае това, то истиността е в сила и за подобектите на разгледания.

Структурна индукция: примери

- *Example 1: binary trees*

Base case: show holds for tree consisting of a leaf only

Induction step: show that it holds for two sub-tree then it holds for tree built out of them

- *Example 2: sequences*

Base case: show predicate holds for $\langle \rangle$ i.e. $P\langle \rangle$ is true

Induction step: Show that if it holds for sequence s i.e. $P s$, then it holds for $\langle x \rangle \wedge s$ i.e. $P(\langle x \rangle \wedge s)$ is also true.

Обобщение на техниките за доказателство

- Natural Deduction
reasoning with deduction rules
- Equational reasoning
reasoning with equivalences
Induction
natural and structural
- Special forms
case analysis
one point rule
proof by contradiction
etc.