Métodos Formales para Ingeniería de Software

Conjuntos y Relaciones

Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Argentina

Especificaciones formales

¿ qué es una especificación formal?

Las especificaciones formales son técnicas basadas en matemática o lógica cuyo propósito es ayudar con la implementación de sistemas y software.

Se utilizan para describir un sistema, para analizar su comportamiento y para ayudar en su diseño, verificando las propiedades clave de interés a través de herramientas de razonamiento rigurosas y eficaces.

Estas especificaciones son formales en el sentido de que tienen una sintaxis, su semántica cae dentro de un dominio, y pueden utilizarse para inferir información útil

Conjuntos y Relaciones

Conceptos fundamentales para trabajar con herramientas automáticas de alto nivel (modelos – propiedades simples)

Conjunto (conjunto base):

- Colección de objetos distintos
- Cada conjunto base se construye a partir de un dominio de objetos (en este contexto los conjuntos son homogéneos)

Los conjuntos base están formados por valores del mismo dominio

Conjuntos base

Los conjuntos base se pueden definir por **enumeración**

Útil en casos finitos, pero la precisión es importante y en muchos casos se vuelve inviable

Los conjuntos base se pueden definir por **comprensión**

Se describen a través de una propiedad que todos los elementos deben compartir

Conjuntos: cardinalidad - particiones

La cardinalidad de un conjunto, notada con el símbolo **#**, está definida como la cantidad de sus elementos.

En una partición de conjuntos, los mismos son disjuntos (no comparten elementos)

> Al modelar, algunos conjuntos se dividen en subconjuntos disjuntos, llamados particiones

Ejemplo:

Dominio básico: residencias
Particiones: Departamento, Casa, Hogar, Pieza

Relaciones

Son la manera de hacer referencia a valores estructurados

- Grupo de valores que están relacionados
- Conjunto de tuplas
- Se definen a partir de **conjuntos base**

Ejemplo:

Conjuntos Base: Entero, Natural

Relaciones:

cuadrados: relaciona un número entero con un natural, siendo el natural el cuadrado del número entero

cuadrados = {(-1,1),(2,4),(-3,9),(3,9)}

Relaciones y Conjuntos

Los conjuntos base pueden pensarse como relaciones de aridad **1**.

Tanto para **relaciones** como para **conjuntos base**, pueden utilizarse las operaciones habituales para conjuntos:

- ✓ union (+)
- ✓ intersección (&)
- ✓ diferencia (-)

Deben aplicarse sobre relaciones de misma aridad

También pueden utilizarse las propiedades de clausura

El operador de cardinalidad (#) puede aplicarse también a relaciones.
El resultado es la cantidad de tuplas que la definen.

Relaciones: producto

El producto de dos relaciones A (de aridad n) y B (de aridad m) es la relación AxB (de aridad n+m) tal que los primeros n elementos corresponden a una tupla de la relación A y los restantes m elementos a una tupla de la relación B.

Ejemplo:

A = {a1,a2} B = {b1}

c = A x B = {(a1,b1),(a2,b1)}

D = {d1,d2}

e = D x c = { (d1,a1,b1),(d1,a2,b1), (d2,a1,b1),(d2,a2,b1) }

Relaciones: producto

Una relación binaria definida para los conjuntos base A y B, es un elemento de Partes(AxB), es decir un subconjunto de AxB.

Una relación ternaria es un elemento de Partes(AxBxC)

Esta idea se generaliza para relaciones de cualquier aridad.

Básicamente, toda relación es un producto de conjuntos base o un subconjunto de dicho producto.

Ejemplo:

Persona = { juan,abril,santi,maria }

Persona x Persona = { (juan,juan), (juan,abril), (juan,santi), (juan,maria),
(abril,juan), (abril,abril), (abril,santi), (abril,maria),
(santi,juan), (santi,abril), (santi,santi), (santi,maria),
(maria,juan), (maria,abril), (maria,santi), (maria,maria) }

padre = { (juan,abril), (juan,santi) }

Relaciones: dominio y rango

El **dominio** de una relación, es el conjunto de átomos que aparecen como primer elemento en las tuplas que la definen.
El **rango** de una relación, es el conjunto de átomos que aparecen como último elemento en las tuplas que la definen.

En general el dominio y rango de una relación **pueden no** coincidir con el dominio y rango de los conjuntos base que la definen

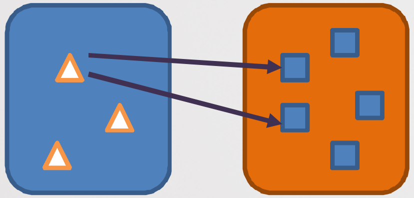
Ejemplo:

padre = { (juan,abril), (juan,santi) }

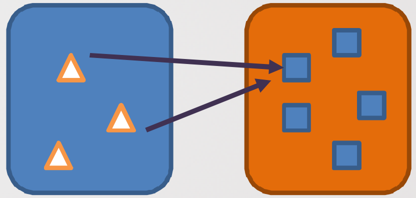
Para esta relación el dominio es {juan} **(no Persona)**
De la misma manera {abril, santi} es el rango **(no Persona)**

Relaciones: estructuras comunes

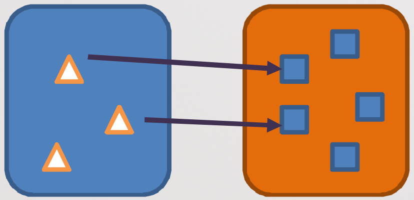
Uno-a-Muchos



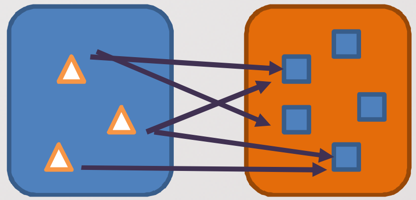
Muchos-a-Uno



Uno-a-Uno



Muchos-a-Muchos



Relaciones funcionales

Una **función** es una relación de aridad n+1, que no puede contener dos tuplas diferentes para los primeros n elementos.

$$\forall\ (a,b) \epsilon\ F, \forall (c,d) \epsilon\ F, (a = c \Rightarrow b = d)$$

Las relaciones funcionales se notarán:

$$F: A_1 \text{ x } A_2 \ldots \text{ x } A_n \mathbin{\textbf{->}} B$$

en lugar de $F: A_1 \text{ x } A_2 \ldots \text{ x } A_n\ \mathbf{x}\ B$

Básicamente una función es una relación X-a-uno (donde X es uno o muchos)

Relaciones funcionales

Una **función** f (con dominio $\subseteq$ S y rango $\subseteq$ T):

- Es una función **total** si está definida para todos los valores de S
- Es una función **parcial** si está definida para algunos valores de S

- Es una función **inyectiva** (define una relación Uno-a-Uno) si no hay elementos del rango asociados a múltiples elementos del dominio

Obs: puede haber relaciones inyectivas no funcionales (uno-a-muchos)

- Es una función **sobreyectiva** (onto) si su rango es T
- Es una función **biyectiva** si resulta ser inyectiva y sobreyectiva

Permite la definición concisa de propiedades

Relaciones

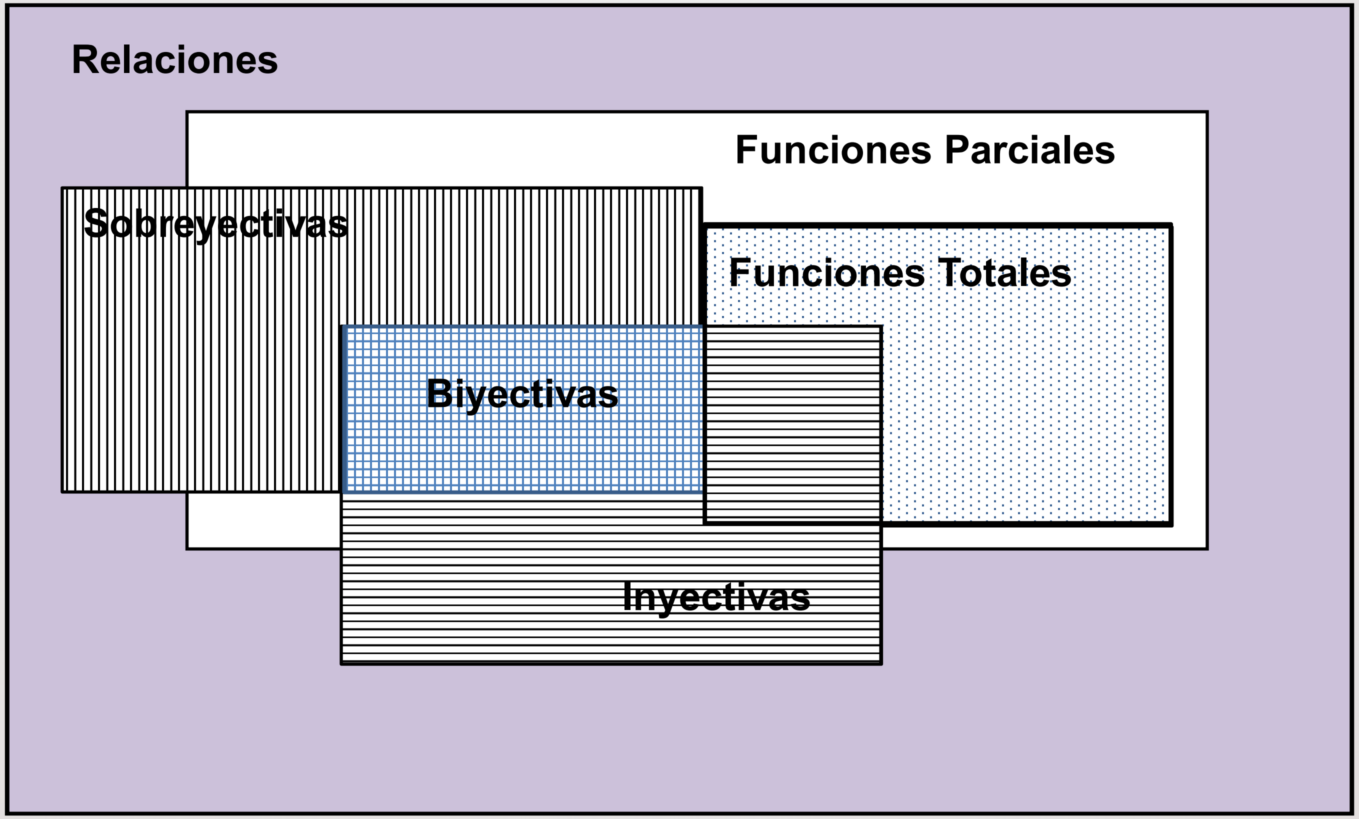
Una **relación** r (con dominio $\subseteq$ S y rango $\subseteq$ T):

- Es una relación **inyectiva** si define una relación Uno-a-Muchos o Uno-a-Uno (en cuyo caso es una relación funcional)

- Es una relación **sobreyectiva** (onto) si su rango es T

- Si la relación es **biyectiva** si resulta ser inyectiva y sobreyectiva lo que la convierte un una relación funcional

Resumen


Relaciones
Funciones Parciales
Sobreyectivas
Funciones Totales
Biyectivas
Inyectivas

Relaciones: composición

La composición permite generar una nueva relación a partir de dos relaciones

- La relación nueva mapea el dominio de la primer relación al rango de la segunda
- Dada s: A x B y r: B x C entonces s;r: A x C

$$s;r = \{ (a,c) \mid (a,b) \in s \text{ and } (b,c) \in r \}$$

Relaciones: clausura

Intuitivamente, la clausura de la una relación, es la composición continua de la misma hasta que no se consigan nuevas tuplas

- Sea r: S x S

$$r^{+} = r \cup (r;r) \cup (r;r;r) \cup \ldots$$

Relaciones: traspuesta

Intuitivamente, la relación traspuesta está formada por los pares reversos de los que componen la relación original

- Sea r: S x T

$$\boxed{\sim r = \{(b,a) \mid (a,b) \in r\}}$$

Agradecimientos

Material de lectura

- capítulo 3 de “Software Abstractions: Logic, Language, and Analysis” de Daniel Jackson, revised edition, MIT Press, 2012

Algunos contenidos están adaptados de:

- las slides de David Garlan’s para su curso “Software Models”
- las slides de Cesare Tinelli de la Universidad de Iowa.