Graduate Program in Production Engineering and Systems

(PPGEPS)

Important Terms in an Ontology (Part 4/6)

Dr. Yongxin Liao

Course Outlines

- Important Terms in an Ontology (4/6)
  - Data Ranges
    - Data Types
    - Enumeration of Data Values (Literals)
    - Data Range Connectives
  - Complex Class Expressions (2/2)
    - Data Property Restrictions
- Protégé Practices

PUCPR

Important Terms in an Ontology

- Several Important Terms
  - Axioms
  - Concepts (Individuals and Classes)
  - Relationships (Class Assertions, Subclasses Disjoint/Equivalent Classes, Individual Equality/Inequality Properties, Property Assertions, Property Characteristics, and Property Descriptions)
  - Complex Class Expressions (Enumeration of Individuals, Propositional Connectives, Object Property Restrictions, Necessary and Sufficient Conditions, **Data Property Restrictions)**
  - **Data Ranges (Data Types and Data Type Restrictions)**
  - Reasoning Rules
  - Knowledge Base (T-box and A-box)
  - SPARQL Query

Part 1/6

Part 2/6

Part 3/6

Part 4/6

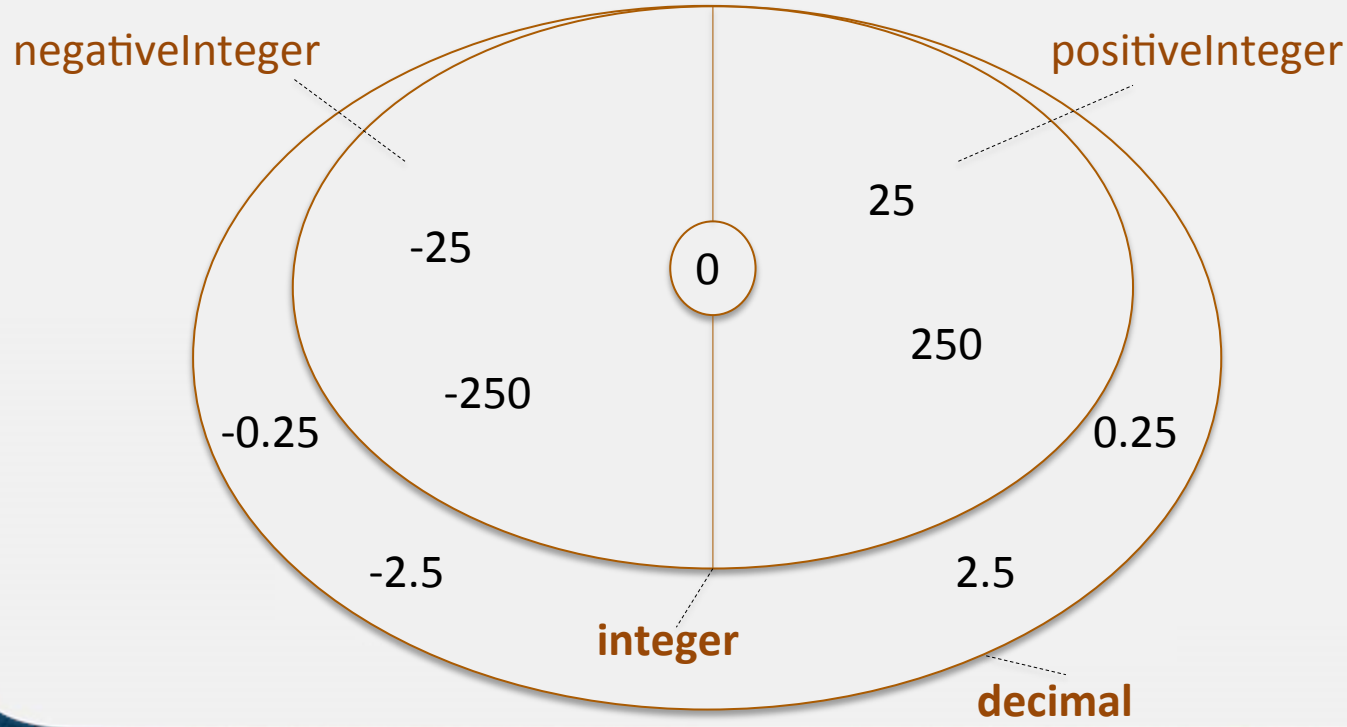
Part 5/6

Part 6/6

PUCPR

Data Ranges

- Data Type
  - Each kind of data values is called a data type.
  - Data Types for Numbers



Data Ranges

- Data Type
  - Data Types for Strings
    - string character strings

    E.g. “A pizza that only has Mozzarella and Tomato Sauce Toppings.”^^string

    “Harry James Potter”^^string
  - Data Types for Time Instants
    - dateTime instances of time “YYYY-MM-DDThh:mm:ss”

    E.g. “2015-10-30T14:00:00”^^dateTime

    “2015-10-30T14:00:00-03:00”^^dateTime

    “2015-10-30T14:00:00+08:00”^^dateTime
  - Data Types for Boolean Values
    - boolean the values of two-valued logic

    E.g. “true”^^boolean “false”^^boolean

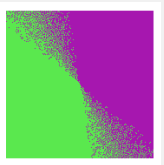
PUCPR

Data Ranges

- Enumeration of Data Values
  - An enumeration of data values contains **exactly** the explicitly specified data values
  - In protégé, it is described by
    - Precisely listing all data values
    - Separating them by ","
    - Inside curly brackets ("{" and "}")

E.g. Bitmaps can require a color-depth of up to 24 bits per pixel (1,4,8,16,24 bits)

Bitmap Examples



(**1** bits/pixel)

(**8** bits/pixel)

(**24** bits/pixel)

The range of the data property hasBits is {1, 4, 8, 16, 24}

{"1"^^integer, "4"^^integer, "8"^^integer, "16"^^integer, "24"^^integer}

{"1"^^xsd:integer, "4"^^xsd:integer, "8"^^xsd:integer, "16"^^xsd:integer, "24"^^xsd:integer}

PUCPR

Data Ranges

- Datatype Restrictions
  - A datatype restriction contains *a data type* and *its value space restrictions*
  - In protégé, it is described by giving a data type and its value space restrictions inside square brackets ("[" and "]").
    - A data type + "[" + value space restrictions + "]"
      - Restrictions on Data types for numbers

        e.g. integer[>1, <5]
      - Restrictions on Data types for strings

        e.g. string[length 5]
      - Restrictions on Data types for time instances

        e.g. dateTime[> 1991-09-10T00:00:00]

PUCPR

Data Ranges

- Datatype Restrictions
  - Restrictions on Data types for numbers

A data type + "[" + value space restrictions + "]"

"decimal",
"integer",
"negativeInteger",
"positiveInteger"

- Each value space restriction is composed of:
  a restriction symbol + a number
- It is possible to have multiple restrictions that are separated by ",".
- The restriction symbol can be "<=", ">=", "<", ">"

The relation among those restrictions is intersection.

E.g. decimal[< -12]

integer[> 1, < 5]

PUCPR

Data Ranges

- Datatype Restrictions
  - Restrictions on Data types for strings

A data type + “[” + value space restrictions + ”]”

“string”

Zero or a positive integer

- Each value space restriction is composed of:
  a restriction symbol + a number
- It is possible to have multiple restrictions that are separated by “,”.
- The restriction symbol can be “length”, “minLength”, or “maxLength”

The relation among those restrictions is intersection.

E.g. string[length “5"^^integer]

string[minLength "4"^^integer, maxLength "6"^^integer]

PUCPR

Data Ranges

- Datatype Restrictions
  - Restrictions on Data types for Time Instances

A data type + “[” + value space restrictions + ”]”

“dateTime”

- Each value space restriction is composed of:
  a restriction symbol + a date time
- It is possible to have multiple restrictions that are separated by “,”.
- The restriction symbol can be “<=“, “>=”, “<”, “>”

The relation among those restrictions is intersection.

E.g. dateTime[> 1991-09-10T00:00:00]

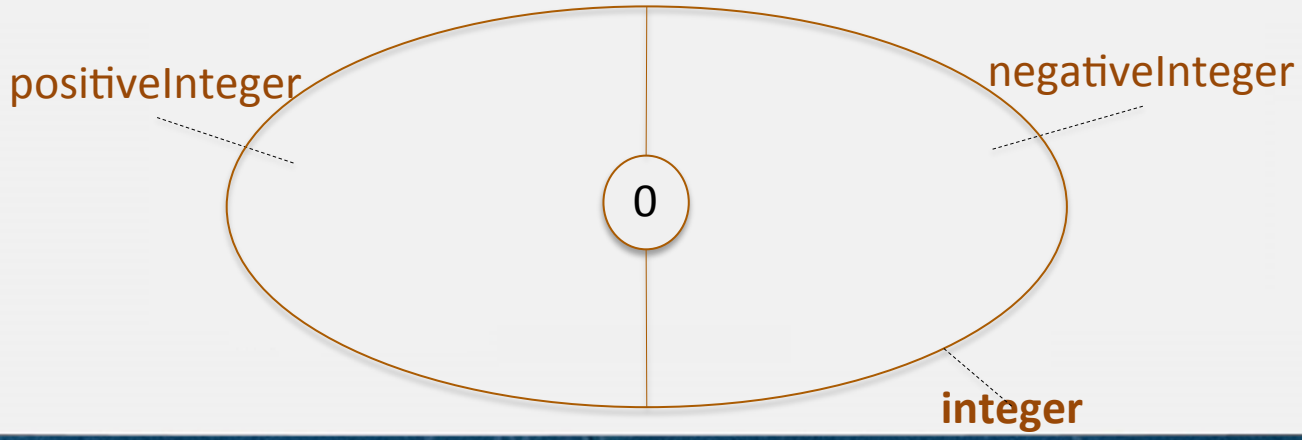
dateTime[> 1980-10-10T00:00:00 , < 1990-10-10T00:00:00]

PUCPR

Data Ranges

- Data Range Connectives
  - Intersection of Data Ranges
    - An intersection data range contains all the data values that are **shared (co-owned)** by all the data ranges in this expression
    - In protégé, it is described by combining two or more data ranges using the "and" operator.

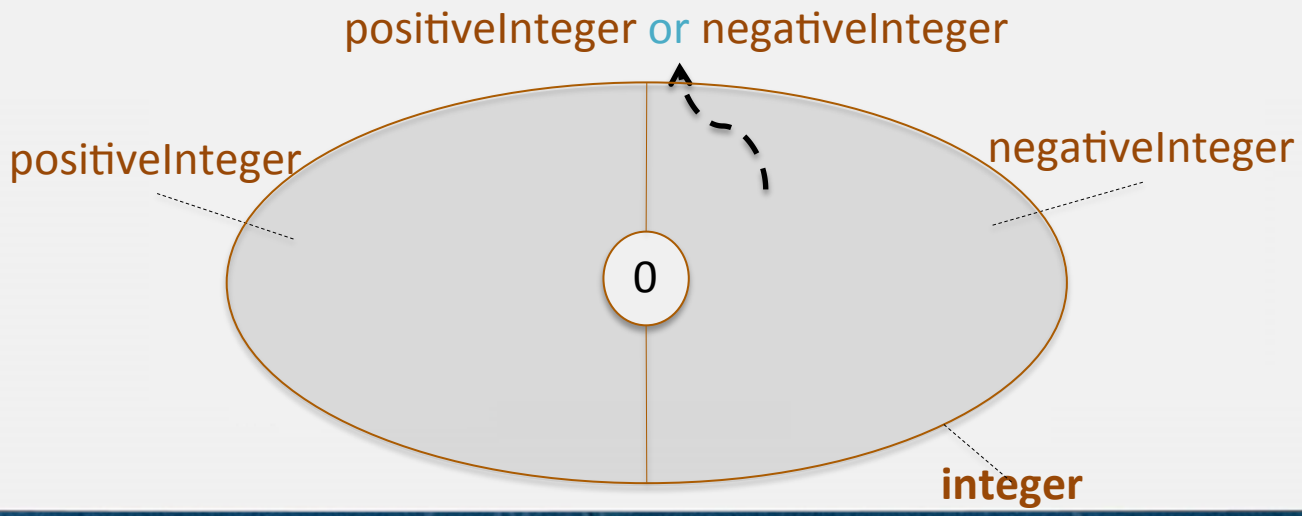
positiveInteger and negativeInteger = ∅



PUCPR

Data Ranges

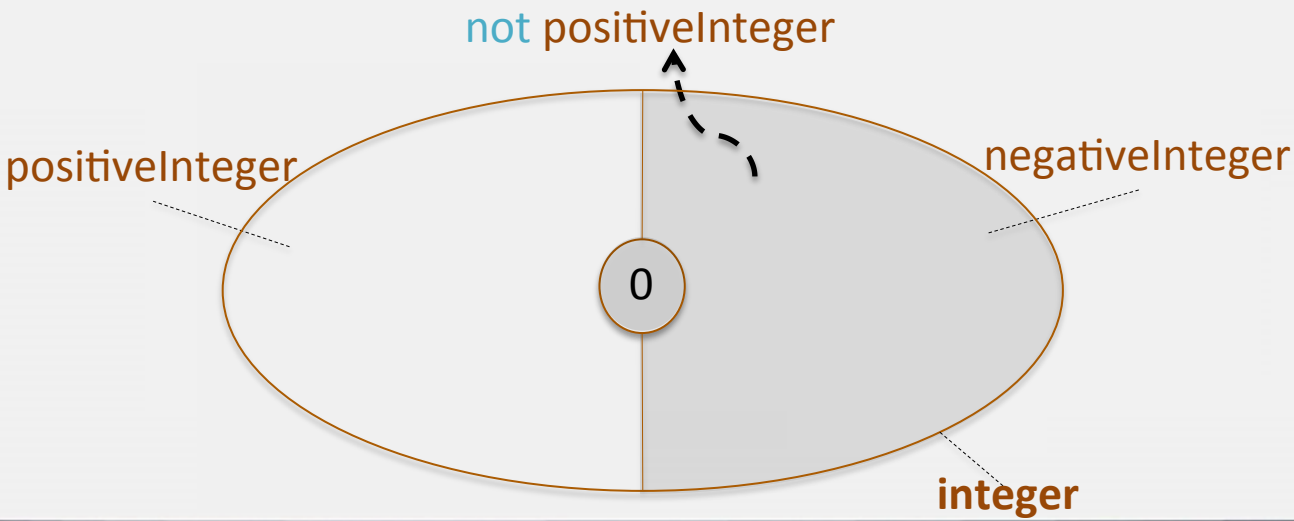
- Data Range Connectives
  - Union of Data Ranges
    - A union data range contains all the data values that are contained in at least one data range in this expression
    - In protégé, it is described by combining two or more data ranges using the "or" operator.



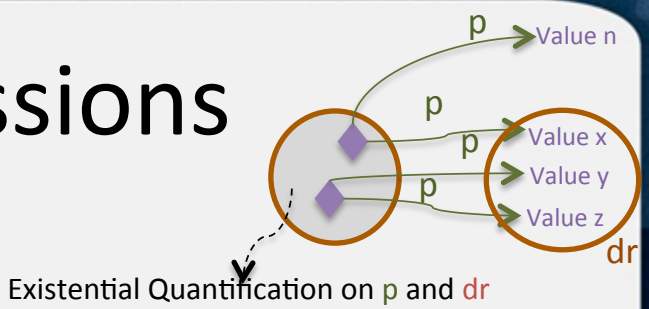
PUCPR

Data Ranges

- Data Range Connectives
  - Complement of Data Ranges
    - The complement of the data range x contains all the data values that are not contained in the data range x.
    - In protégé, it is described by placing a "not" operator **at the beginning of** that data range.

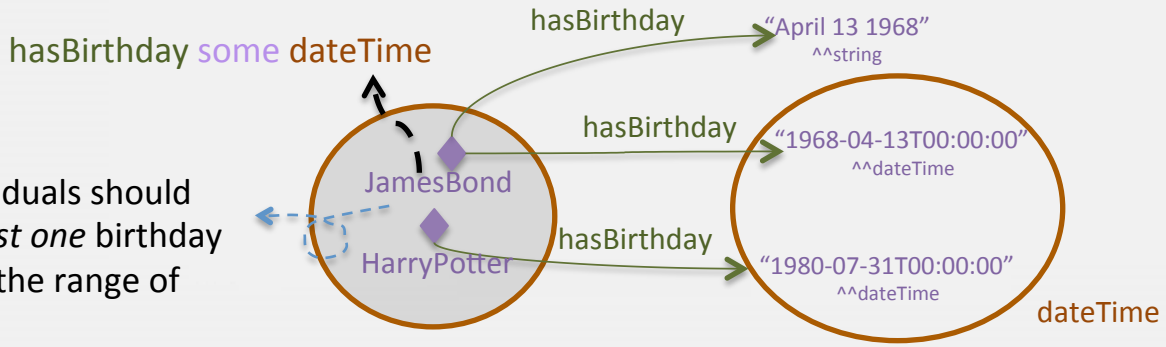


Class Expressions



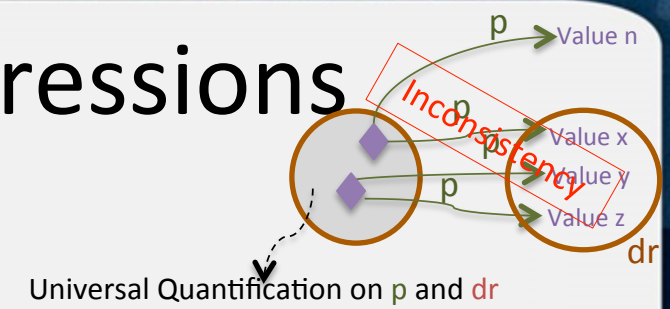
- Quantifier Restrictions
  - Data Property Existential Quantification
    - A data property existential restriction expression consists of a data property p and a data range dr
    - It contains all those individuals that are connected, at least once, by p to the data values that are in the range of dr.
    - In Protégé, it is described as

      a data property + "some" + a data range



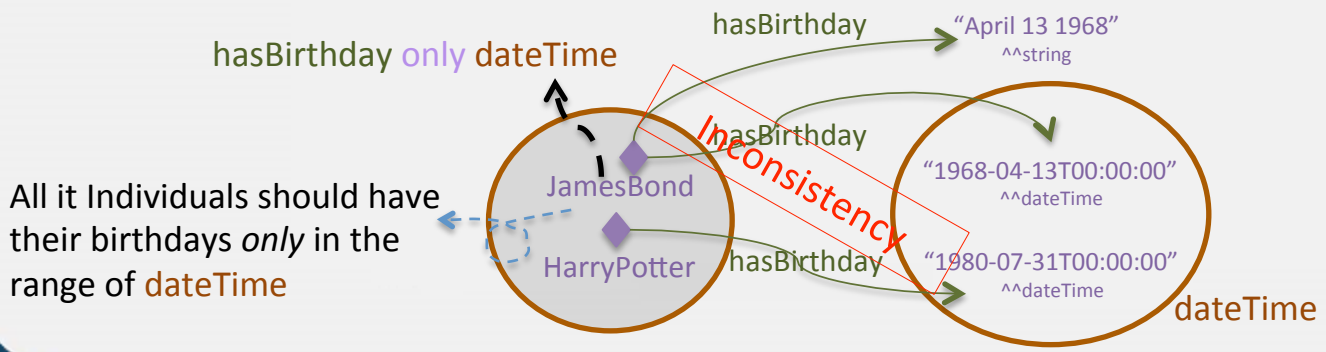
PUCPR

Class Expressions



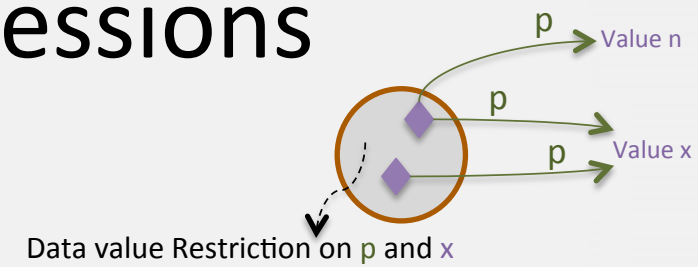
- Quantifier Restrictions
  - Data Property Universal Quantification
    - A data property universal restriction expression consists of a data property p and a data range dr
    - it contains all those individuals that are only connected by p to the data values that are in the range of dr.
    - In Protégé, it is described as

      a data property + "only" + a data range



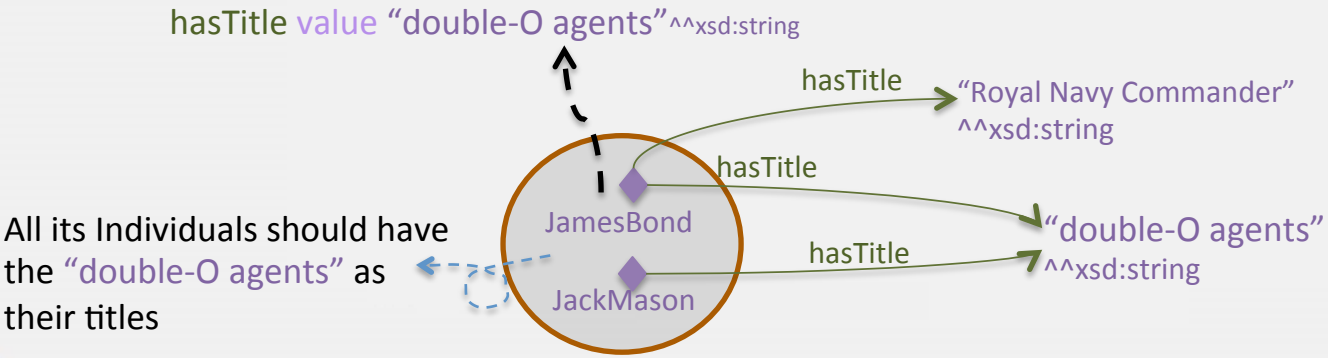
PUCPR

Class Expressions

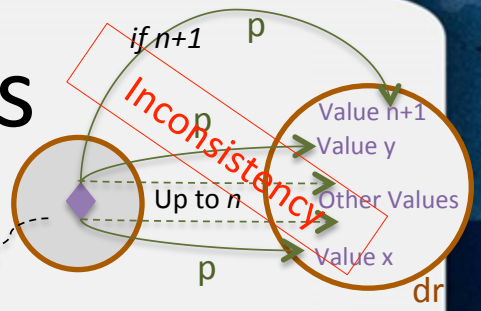


- Value Restrictions
  - Data Value Restriction
    - A data value restriction expression consists of a data property p and a data value x
    - It contains all those individuals that are connected by p to x.
    - In Protégé, it is described as

      a data property + "value" + a data value



Class Expressions

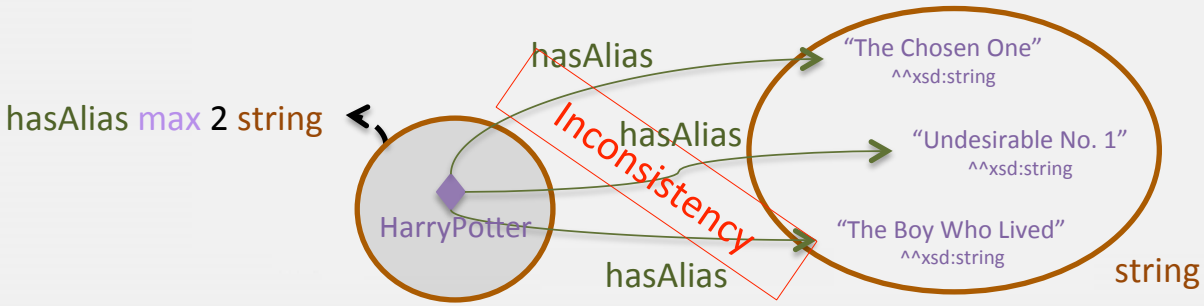


- Cardinality Restrictions

Maximum cardinality restriction on p, *n* and dr

  - Data Property Maximum Cardinality
    - A data property maximum cardinality expression consists of a data property p, a nonnegative integer *n*, and a data range dr
    - It contains all those individuals that are connected by p to **<u>at most *n* different</u>** data values that are in the range of dr.
    - In protégé, it is described as

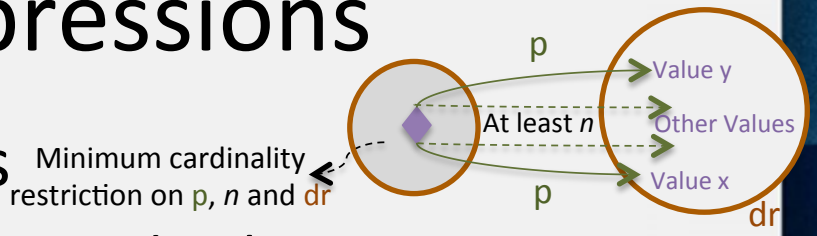
a data property + "max" + a nonnegative integer + a data range



PUCPR

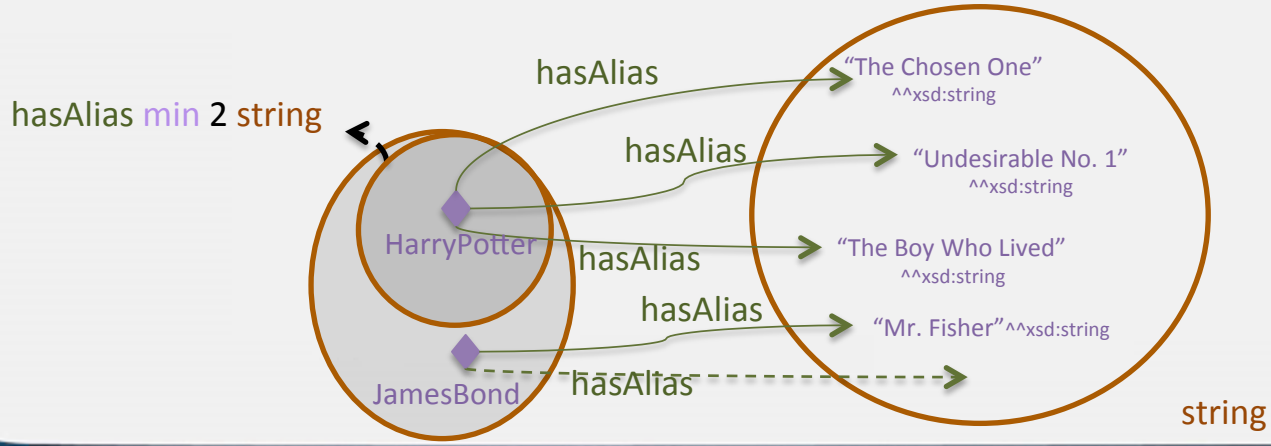
Class Expressions

- Cardinality Restrictions

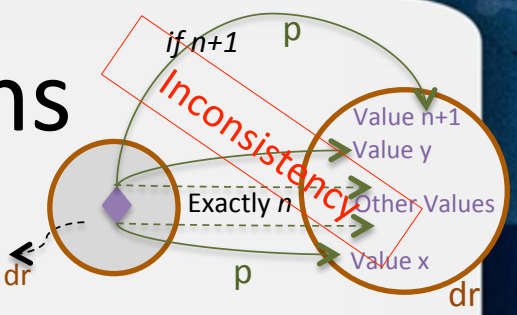


  - Data Property Minimum Cardinality
    - A data property minimum cardinality restriction is consist of a data property p, a nonnegative integer *n*, and a data range dr
    - It contains all those individuals that are connected by p to **<u>at least *n* different</u>** data values that are in the range of dr.
    - In protégé, it is described as

a data property + "min" + a nonnegative integer+ a data range



Class Expressions

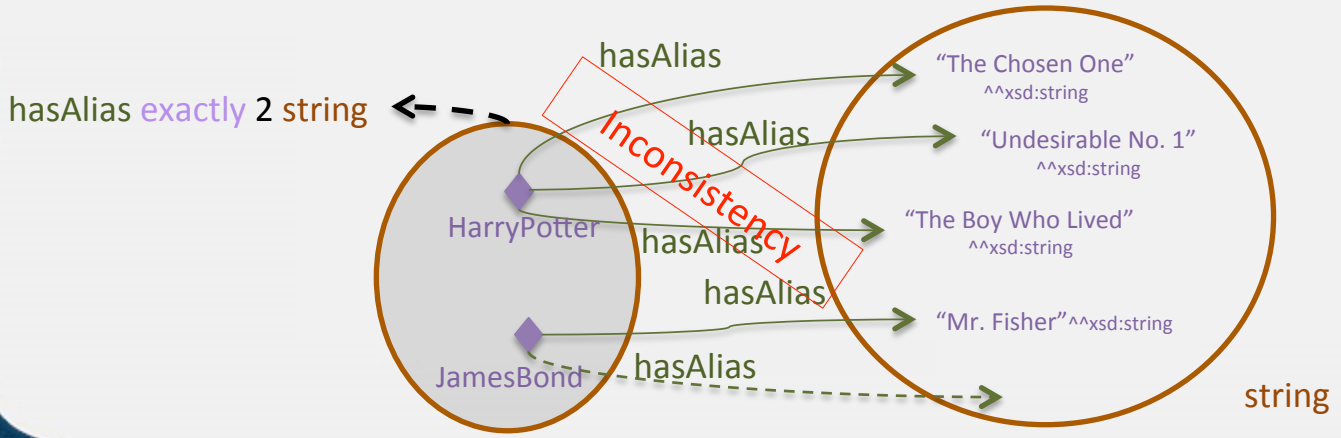


- Cardinality Restrictions

Exact cardinality restriction on p, *n* and dr

  - Data Property Exact Cardinality
    - A data property exact cardinality restriction is consist of a data property p, a nonnegative integer *n*, and a data range dr
    - It contains all those individuals that are connected by p to **exactly *n* different** data values that are in the range of dr.
    - In protégé, it is described as

a data property + "exactly" + a nonnegative integer + a data range



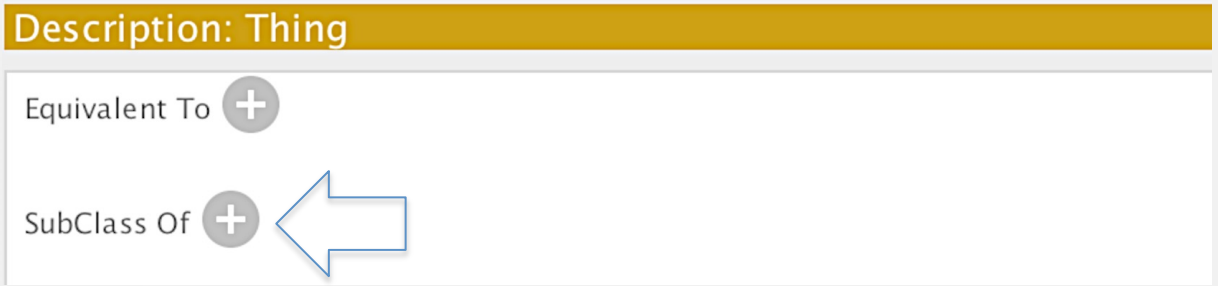
Protégé Practices

- The “Classes” Tab
  - Complex Class Expressions (via Data Properties)
    - Equivalent Classes Axioms
    - Subclass Axioms
- The “Data Properties” Tab
  - Data Property Ranges
    - Data Range Expressions

PUCPR

Protégé Practices

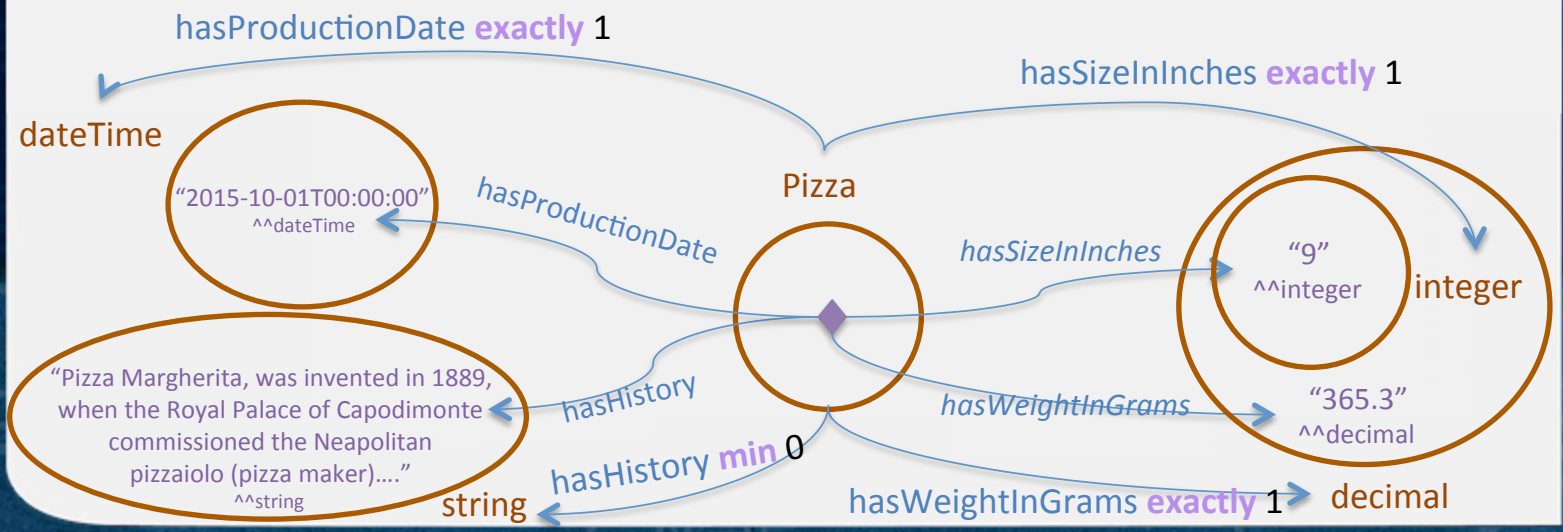
- The “SubClass Of”
  - “SubClass Of” => necessary conditions
    - Let class C be a subclass of some necessary conditions
    - All the individuals and subclasses of C **are necessary to fulfill** those conditions.



PUCPR

Protégé Practices

- Pizzas
  - All pizza should fulfill the following necessary conditions:
    a) Each pizza contains exactly one weight in grams (decimal);
    b) Each pizza contains exactly one size in inches (integer);
    c) Each pizza contains exactly one production date (dateTime);
    d) Each Pizza might have zero or more history backgrounds (string).



PUCPR

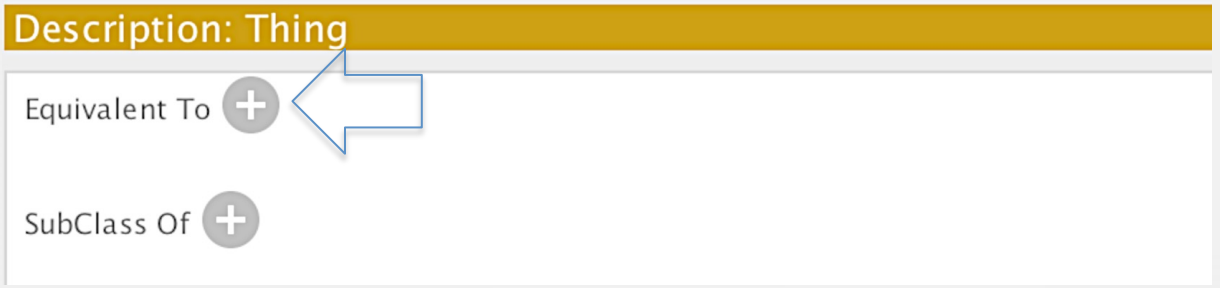
Protégé Practices

- Four Kinds of Pizzas
  - **Margherita Pizza**
    - Its size is 9 inches
    - Its weight is from 360 to 380 grams
    - It has one history background "Pizza Margherita, was invented in 1889, when the Royal Palace of Capodimonte commissioned the Neapolitan pizzaiolo (pizza maker) Raffaele Esposito to create a pizza in honor of the visiting Queen Margherita."
  - **American Pizza**
    - Its size is 9 inches
    - Its weight is from 400 to 420 grams
  - **American Hot Pizza**
    - Its size is either 6 or 9 inches
    - Its weight is from 150 to 450 grams
  - **Soho Pizza**
    - Its size is 12 inches.
    - Its weight is from 550 to 580 grams

PUCPR

Protégé Practices

- The “Equivalent To”
  - “Equivalent To” => necessary & sufficient conditions
    - Let class C be a equivalent class of some necessary &sufficient conditions
    - All the individuals and subclasses of C **are necessary to fulfill** those conditions.
    - Meanwhile, an individual (a class) that satisfy those conditions **is sufficient to be** a member (a subclass) of C.



PUCPR

Protégé Practices

- Data Type: string
- Historical Pizzas (Data Type: string)
  - All Historical Pizzas (all the individuals and subclasses of class: HistoricalPizza) should contains at least one historical background (a historical background in the data type of string).
  - A pizza (a collection of pizzas) that contains at least one historical background (a historical background in the data type of string) is sufficient to be a member (a subset) of Historical Pizzas

PUCPR

Protégé Practices

- Data Type: integer
- Six or Nine Inches Pizzas (Data Type: integer)
  - All Six or Nine Inches Pizzas (all the individuals of class: NineInchesPizza) should be in the size of six or nine inches.
  - A pizza (a collection of pizzas) that in the size of six or nine inches is sufficient to be a member (a subset) of Six or Nine Inches Pizzas.

PUCPR

Protégé Practices

- Data Type: decimal
- Big Pizzas (Data Type: decimal)
  - All Big Pizzas (all the individuals and subclasses of class: BigPizza) should have a weight more than half a kilogram.
  - A pizza (a collection of pizzas) that have a weight more than half a kilogram is sufficient to be a member (a subset) of Big Pizzas

PUCPR

Protégé Practices

- Data Type: dateTime
- Pizzas Produced In July 2015 (Data Type: dateTime)
  - All Pizzas in the class PizzaProducedInJuly2015 should be produced during the July of 2015.
  - A pizza (a collection of pizzas) that is produced during the July of 2015 is sufficient to be a member (a subset) of PizzaProducedInJuly2015

PUCPR

Protégé Practices

- Data Type: boolean
- The Expired Pizzas (Data Type: boolean)
  - Create an new data property named isExpired.
  - Specify the ExamplePizzaX is in the expired state.
  - A Class named ExpiredPizza to classify all pizzas that are expired.

PUCPR

Protégé Practices

- Complete the Additional Protégé Practices
- Answer the corresponding questions.

PUCPR

Thank you
for your attention!

Any Questions?

PUCPR