INFS3101/7100 Ontology and the Semantic Web

Module 13 Predicates

Last lecture: Key Terms

- Widely-used structural features include: **countable/ bulk** classes, **concept/representation** classes, **dimension systems**, **mereological structures**, **metaproperties** and **extent-descriptive metaclasses.** All can be modelled using extensions to OWL.

Predicates

- A predicate can be true or false.
  - Facts
    - "My surname is Colomb"
    - "Regina is a student at the University of Queensland"
  - Integrity constraints
    - "Every student has exactly one student identification number"
  - Definitions
    - "All students enrolled in INFS3101 or in INFS7100 are on my classlist"
  - Rules for properties
    - "If X is a distance in metres, then X/1000 is the same distance in kilometers"
  - Queries rely on definitions
    - "The extent of my classlist is …?"

Formal Reasoning

- Done by inference engines.
  - given a fact, determine classes using definitions
  - given a definition, determine facts (compute the result of a query)
  - Calculate properties by their rules
  - check integrity constraints after proposed changes
  - determine whether or not a definition is consistent
  - determine whether one definition subsumes another

Aggregation

- Database systems can calculate property of a class
  - number of instances in a class
    - COUNT
  - total of some numeric property
    - SUM
- and derived properties using GROUP BY
- Depends on classes having finite extents
- Not all classes do
  - Natural numbers
- And the unique names assumption

Common Logic

- Anything that can be expressed in OWL can be expressed in CL
- Basic construct is atom
  - (rdfs:Class air:City)
  - (air:City air:LosAngeles)
  - (air:operates air:QANTAS air:QF1)
  - (surname me 'Colomb')
  - (student Regina TheUniversityOfQueensland)
    - (Person Regina)
    - (University TheUniversityOfQueensland)

Quantified Sentences

- ❖ (exists x (air:City x))
- ❖ (forall x (implies
  - (air:International x) (air:Airline x)))
- ❖ Declared subclass

- ❖ (forall x (iff
  - (or (enrolled x INFS3101) (enrolled x INFS7100))
  - (on-my-classlist x)
- ❖))
- ❖ Union of two classes in OWL
- ❖ Defined subclass

More sentences

- ❖ Domain and range of a property
  - (forall (s u) (implies
  - (student s u)
  - (and (Person s) (University u))
  -))
- ❖ Integrity constraint (cardinality) minimum cardinality 1
  - (forall student (implies
  - (exists ID (studentID student ID))
  - (and
  - (forall i (studentID student i))
  - (= i ID)
  -)
  -))

Definitions

- ❖ AllValuesFrom
  - (forall f (iff
  - (USOriginFlights f)
  - (forall c
  - (implies (air:origin f c) (USCity c)))
  -))
- ❖ IntersectionOf
  - (forall f (iff
  - (USFlight f)
  - (and (USOriginFlights f) (USDestFlights f))
  -))
- ❖ Iff equivalent to owl:equivalentClass

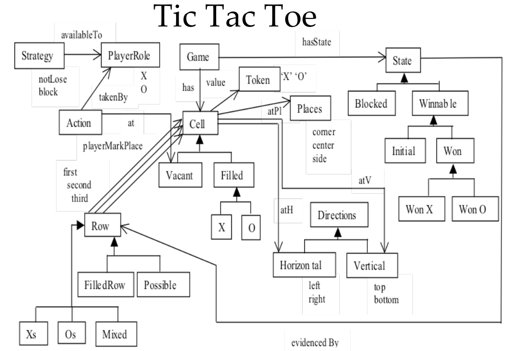
Definitions

- ❖ complementOf
  - (forall f (iff
  - (International f)
  - (and (Flight f) (not (Domestic f)))
  -))
- ❖ someValuesFrom
  - (forall a (iff
  - (InternationalAirline a)
  - (implies
  - (and (Airline a)
  - (exists f (and
  - (operates a f)
  - (International f)
  -))))))

CL not OWL

- ❖ Square is a rectangle with equal sides
  - (forall r (iff
  - (square r)
  - (and (rectangle r) (length r l) (width r w) (= l w))
  -))
- ❖ Can conclude
  - (forall s (implies
    - (square s)
    - (rectangle s)))
- ❖ Defined subclass

Tic Tac Toe



Advantage of Defined Subclasses

- Defined subclasses have more clarity than declared subclasses. Given an object, can tell what classes it is an instance of.
- Also more coherent. Can tell if the definition is consistent, or if one subclass subsumes another.
- If we name semantically meaningful subclass definitions, we gain in extendibility also.
- We will show CL equivalents of some of the OWL subclass definitions for the Tic-Tac-Toe Ontology

Subclasses in Tic-Tac-Toe

- We declare a class of properties.
  - (cellsOfRow first)
  - (cellsOfRow second)
  - (cellsOfRow third)
- The subclasses of *Cell*: *X*, *O*, *Filled* and *Vacant*.
  - (forall c (iff (**X** c) (and (Cell c) (value c 'X'))))
  - (forall c (iff (**O** c) (and (Cell c) (value c 'O'))))
  - (forall c (iff (**Filled** c) (or (X c) (O c))))
  - (forall c (iff (**Vacant** c)
  - (and (Cell c) not (Filled c))))

Subclasses in Tic-Tac-Toe

- Subclasses of *Row*: *Xs*, *Os* and *Mixed*.
- (forall r (iff (**Xs** r)
  - (forall (cr c)
    - (and (cellsOfRow cr)
      - (Rows r) (implies (cr r c) (not O c))))))
- (forall r (iff (**Os** r)
  - (forall (cr c)
    - (and (cellsOfRow cr)
      - (Rows r) (implies (cr r c) (not X c))))))
- (forall r (iff (**Mixed** r)
  - (forall (cr c)
    - (implies (Rows r) (and
      - (exists (cr c) (and (cellsOfRow cr) (cr r c) (X c)))
      - (exists (cr c) (and (cellsOfRow cr) (cr r c) (O c))))))))

Subclasses in Tic-Tac-Toe

- Subclasses of *Row*: *FilledRow* and *Possible*
  - (forall r (iff (**FilledRow** r)
    - (and (Rows r)
      - (not (exists (cr c) (and
        - (cellsOfRow cr)
        - (cr r c)
        - (Vacant c)))))))
  - (forall r (iff (**Possible** r)
    - (and (Rows r)
      - (not (Filled r))
      - (or (Xs r) (Os r)))))

Subclasses in Tic-Tac-Toe

- Subclasses of *State: WonX, WonO, Won*
  - *Note they are all propositions*
  - (iff **WonX**
    - (exists r (and
      - (Rows r) (Filled r) (Xs r))))
  - (iff **WonO**
    - (exists r (and
      - (Rows r) (Filled r) (Os r))))
  - (iff **Won**
    - (or WonX WonO))

Subclasses in Tic-Tac-Toe

- Subclasses *Initial*, *Blocked*, *Winnable*
  - (iff **Initial**
    - (forall c
    - (implies (Cell c) (Vacant c))))
  - (iff **Blocked**
    - (forall r
    - (implies (Row r) (Mixed r))))
  - (iff **Winnable** (not Blocked))

Dimensioned Quantities

- ❖ (dimensioned-quantity, materialClass,
  - •((System, dimension), unit) value)

- ❖ (forall **mc** (implies (materialClass **mc**)
  - •(= (dimensioned-quantity **mc**
    - – ((SI-system distance) metres) (* 1000 x))
    - ▪ (dimensioned-quantity, **mc**,
      - – ((SI-system, distance) kilometers) x)
      - –)))

Dimensioned Quantities

- ❖ (forall (**mc1 mc2** units value1 value2) ((exists **mc3**)
  - •(= (*
    - – (dimensioned-quantity **mc1**
      - o((SI-system distance) units) value1)
    - – (dimensioned-quantity **mc2**
      - o((SI-system distance) units) value2))
  - ▪ (dimensioned-quantity **mc3**
    - – ((SI-system area) (exp units 2)) (* value1 value2))
  - •)
- ❖))

Dimensioned Quantities

- ❖ For example
- ❖ (forall (value1 value2) (=
  - •(* (dimensioned-quantity **distanceValue**
    - – ((SI-system distance) metres) value1)
    - ▪ (dimensioned-quantity **distanceValue**
      - – ((SI-system distance) metres) value2))
  - •(dimensioned-quantity **areaValue**
    - ▪ ((SI-system area) (exp metres 2)) (* value1 value2))
- ❖))

Connecting OWL with CL

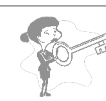
- ❖ (forall r (iff
  - •(map(isSquare) r)
  - •(and
    - ▪ (map(rectangle) r)
    - ▪ (map(length) r l)
    - ▪ (map(width) r w)
    - ▪ (= l w))
- ❖))

- ❖ Need some sort of procedure call mechanism

Russell's Paradox

- ❖ N is the set of all sets which don't contain themselves
- ❖ (forall s (iff (N s) (not (s s))))
- ❖ (implies (N N) (not (N N)))
- ❖ (implies (not (N N)) (N N))
- ❖ Contradiction. Definition inconsistent for N

Summary: Key Terms



- ❖ **Common logic** is a language to express **predicates**. Can map OWL Full to CL. But CL is richer than OWL Full, so can be used to extend OWL. But need to be careful, since CL open to paradox.

Resources

❖Essential
- Notes chapter 13 Predicates

❖Further
- Friendly Guide to Common Logic (on web site)