

Week 13 Suggested Tutorial Solution: Predicates

Semester 1, 2006

Consider the rental accommodation exchange from the week 2 tutorial and the representation in the solution to the week 4 tutorial and following, as represented in RDFS/OWL in the last tutorial. (Make plausible extensions if necessary).

- a. Define subclasses using CL.
- a. Declared subclass – a lessee is a person

```
(forall p
  (implies
    (Lessee p)
    (Person p)
  )
)
```

- Defined subclass

On the suggested solution of week 4 tutorial, class *Property* could have subclasses *Furnished* and *Unfurnished* shown figure 1.

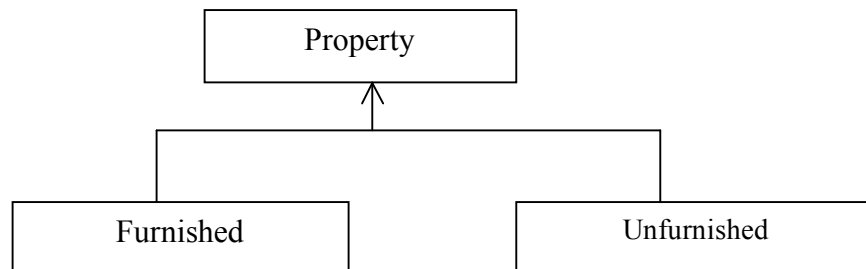


Figure 1

- Definition of class *Furnished*.

```
(forall p
  (implies
    (exists f
      (and
        (Furnishings f)
        (Property p)
        (rental:has p f)
      )
    )
    (Furnished p)
  )
)
```

- Definition of *Unfurnished*

```
(forall p
  (implies
    (not exists f
      (and
        (Furnishings f)
        (Property p)
        (rental:has p f)
      )
    )
  )
)
```

b. Define some integrity constraints using CL.

- Each lease has at least one lessee (minimum cardinality is one).

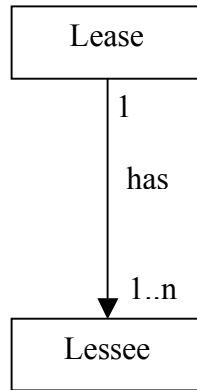


Figure 2

```
(forall lease
  (implies
    (Lease lease)
    (exists lessee (rental:has lease lessee))
  )
)
```

- Each lease has a unique number (hasLeaseNumber is total and functional).

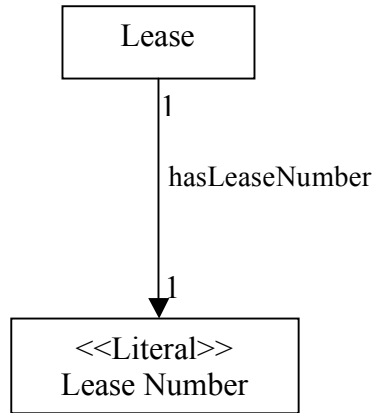


Figure 3

```

(forall lease
  (implies
    (Lease lease)
    (and
      (exists num (hasLeaseNumber lease num))
      (implies
        (forall n (hasLeaseNumber lease n))
        (= n num))
      )
    )
  )
)
  
```

- A lessee pays a rent which amount is specified by the lease confirmed by the lessee. (For this example, it cannot be interpreted in OWL since it is impossible to show all properties for particular instances– *pays*, *confirms* and *specifies* in a chain in OWL.)

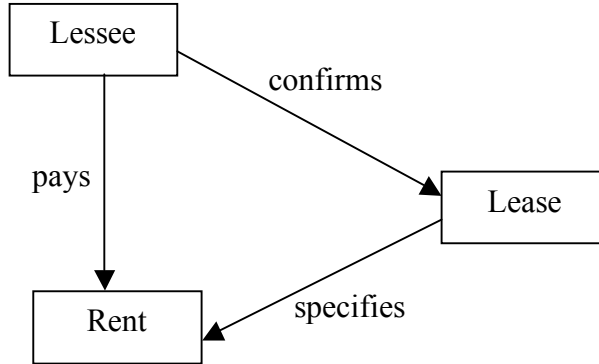


Figure 4

```

(forall (lessee rent)
  (implies
    (and
      (Lessee lessee)
      (Rent rent)
      (rental:pays lessee rent)
    )
    (exists lease
      (and
        (Lease lease)
        (rental:confirms lessee lease)
        (rental:specifies lease rent)
      )
    )
  )
)

```

c. *Formulate some queries as class definitions in CL.*

- Select the instances of *Building*, which are close to only bus stop number 18.

(hasValue in OWL).

```
(forall b
  (iff
    (Building_BusStop#18 b)
    (rental:closest_bus_stop_18 b BusStop#18)
  )
)
```

- Select the instances of *Building*, which are close to only bus stops

(allValuesFrom in OWL).

```
(forall b
  (iff
    (Building_BusStop b)
    (forall bs
      (and (rental:nearness b bs)
        (Bus_Stop bs)
      )
    )
  )
)
```

- Select the instances of *Building*, which are close to bus stops and shops simultaneously (someValuesFrom in OWL).

```
(forall bbs
  (iff
    (Building_BusStop_Shop bbs)
    (and
      (exists bs
        (and
          (rental:nearness bbs bs)
          (Bus_Stop bs)
        )
      )
      (exists s
        (and
          (rental:nearness bbs s)
          (Shop s)
        )
      )
    )
  )
)
```