Practical Session: Knowledge Processing

1. Logic
2. MLN

Logic — Family

Formalize the concepts *Spouse*, *Bigamist*, *Polygamist*, *Widow(er)*, *Bachelor*, *Stepparent* in

1. predicate logic
2. description logic

by using following predicates:

- *isMarried(x,y)*
- *Alive(x)*
- *hasChild(x,y)*

Predicate Logic

Define *Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent* by using *isMarried(x,y)*, *Alive(x)* and *hasChild(x,y)*

$$Spouse(x) \Leftrightarrow \exists y(isMarried(x,y) \wedge Alive(y))$$

Predicate Logic

Define *Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent* by using *isMarried(x,y)*, *Alive(x)* and *hasChild(x,y)*

$$Spouse(x) \Leftrightarrow \exists y(isMarried(x,y) \wedge Alive(y))$$

$$Bigamist(x) \Leftrightarrow \exists y,z(isMarried(x,y) \wedge Alive(y) \wedge isMarried(x,z) \wedge Alive(z) \wedge \neg(y=z) \wedge \neg\exists w(isMarried(x,w) \wedge Alive(w) \wedge \neg(y=w) \wedge \neg(z=w))$$

Predicate Logic

Define *Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent* by using *isMarried(x,y), Alive(x)* and *hasChild(x,y)*

$Spouse(x) \Leftrightarrow \exists y(isMarried(x,y) \wedge Alive(y))$

$Bigamist(x) \Leftrightarrow \exists y,z(isMarried(x,y) \wedge Alive(y) \wedge isMarried(x,z) \wedge Alive(z) \wedge \neg(y = z) \wedge \neg\exists w(isMarried(x,w) \wedge Alive(w) \wedge \neg(y = w) \wedge \neg(z = w))$

$Polygamist(x) \Leftrightarrow \exists y,z(isMarried(x,y) \wedge Alive(y) \wedge isMarried(x,z) \wedge Alive(z) \wedge \neg(y = z))$

Predicate Logic

Define *Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent* by using *isMarried(x,y), Alive(x)* and *hasChild(x,y)*

$$Spouse(x) \Leftrightarrow \exists y(isMarried(x,y) \land Alive(y))$$

$$Bigamist(x) \Leftrightarrow \exists y,z(isMarried(x,y) \land Alive(y) \land isMarried(x,z) \land Alive(z) \land \neg(y = z) \land \neg\exists w(isMarried(x,w) \land Alive(w) \land \neg(y = w) \land \neg(z = w))$$

$$Polygamist(x) \Leftrightarrow \exists y,z(isMarried(x,y) \land Alive(y) \land isMarried(x,z) \land Alive(z) \land \neg(y = z))$$

$$Widow(x) \Leftrightarrow \exists y(isMarried(x,y) \land \neg Alive(y))$$

Predicate Logic

Define *Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent* by using *isMarried(x,y), Alive(x)* and *hasChild(x,y)*

$$Spouse(x) \Leftrightarrow \exists y (isMarried(x, y) \wedge Alive(y))$$

$$Bigamist(x) \Leftrightarrow \exists y, z (isMarried(x, y) \wedge Alive(y) \wedge isMarried(x, z) \wedge Alive(z) \wedge \neg(y = z) \wedge \neg\exists w (isMarried(x, w) \wedge Alive(w) \wedge \neg(y = w) \wedge \neg(z = w))$$

$$Polygamist(x) \Leftrightarrow \exists y, z (isMarried(x, y) \wedge Alive(y) \wedge isMarried(x, z) \wedge Alive(z) \wedge \neg(y = z))$$

$$Widow(x) \Leftrightarrow \exists y (isMarried(x, y) \wedge \neg Alive(y))$$

$$Bachelor(x) \Leftrightarrow \neg\exists y (isMarried(x, y))$$

Predicate Logic

Define *Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent* by using *isMarried(x,y)*, *Alive(x)* and *hasChild(x,y)*

$$Spouse(x) \Leftrightarrow \exists y(isMarried(x,y) \wedge Alive(y))$$

$$Bigamist(x) \Leftrightarrow \exists y,z(isMarried(x,y) \wedge Alive(y) \wedge isMarried(x,z) \wedge Alive(z) \wedge \neg(y=z) \wedge \neg\exists w(isMarried(x,w) \wedge Alive(w) \wedge \neg(y=w) \wedge \neg(z=w))$$

$$Polygamist(x) \Leftrightarrow \exists y,z(isMarried(x,y) \wedge Alive(y) \wedge isMarried(x,z) \wedge Alive(z) \wedge \neg(y=z))$$

$$Widow(x) \Leftrightarrow \exists y(isMarried(x,y) \wedge \neg Alive(y))$$

$$Bachelor(x) \Leftrightarrow \neg\exists y(isMarried(x,y))$$

$$Stepparent(x) \Leftrightarrow \exists y,z(isMarried(x,y) \wedge hasChild(y,z) \wedge \neg hasChild(x,z))$$

Description Logic

Define *Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent* by using *isMarried(x,y), Alive(x)* and *hasChild(x,y)*

$Spouse \doteq \exists isMarried.Alive$

Description Logic

Define *Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent* by using *isMarried(x,y), Alive(x)* and *hasChild(x,y)*

$$Spouse \doteq \exists isMarried.Alive$$

$$Polygamist \doteq (\geq 2\ isMarried.Alive)$$

Description Logic

Define *Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent* by using *isMarried(x,y), Alive(x)* and *hasChild(x,y)*

$Spouse \doteq \exists isMarried.Alive$

$Polygamist \doteq (\geq 2\ isMarried.Alive)$

$Bigamist \doteq Polygamist \sqcap (\leq 2\ isMarried.Alive)$

Description Logic

Define *Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent* by using *isMarried(x,y), Alive(x)* and *hasChild(x,y)*

$Spouse \doteq \exists isMarried.Alive$

$Polygamist \doteq (\geq 2\ isMarried.Alive)$

$Bigamist \doteq Polygamist \sqcap (\leq 2\ isMarried.Alive)$

$Widow \doteq \exists isMarried.\neg Alive$

Description Logic

Define *Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent* by using *isMarried(x,y), Alive(x)* and *hasChild(x,y)*

$Spouse \doteq \exists isMarried.Alive$

$Polygamist \doteq (\geq\ 2\ isMarried.Alive)$

$Bigamist \doteq Polygamist \sqcap (\leq\ 2\ isMarried.Alive)$

$Widow \doteq \exists isMarried.\neg Alive$

$Bachelor \doteq \forall isMarried.\bot$ **or** $Bachelor \doteq \neg\exists isMarried.\top$

Description Logic

Define *Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent* by using *isMarried(x,y), Alive(x)* and *hasChild(x,y)*

$Spouse \doteq \exists isMarried.Alive$

$Polygamist \doteq (\geq 2\ isMarried.Alive)$

$Bigamist \doteq Polygamist \sqcap (\leq 2\ isMarried.Alive)$

$Widow \doteq \exists isMarried.\neg Alive$

$Bachelor \doteq \forall isMarried.\bot$ **or** $Bachelor \doteq \neg\exists isMarried.\top$

$Stepparent \doteq \exists((isMarried \circ hasChild) \sqcap \neg hasChild).\top$

UPDATE Role composition (∘) and Other DL Resources

Note: we corrected the notations

Derivation of the concept *Stepparent*. Compare predicate logic (FOL) and description logic (DL):

FOL: $Stepparent(x) \Leftrightarrow \exists y, z(isMarried(x, y) \wedge hasChild(y, z) \wedge \neg hasChild(x, z))$

DL: $Stepparent \doteq \exists((isMarried \circ hasChild) \sqcap \neg hasChild).\top$

Another example: *Woman that has at least three brothers who are lawyers*:

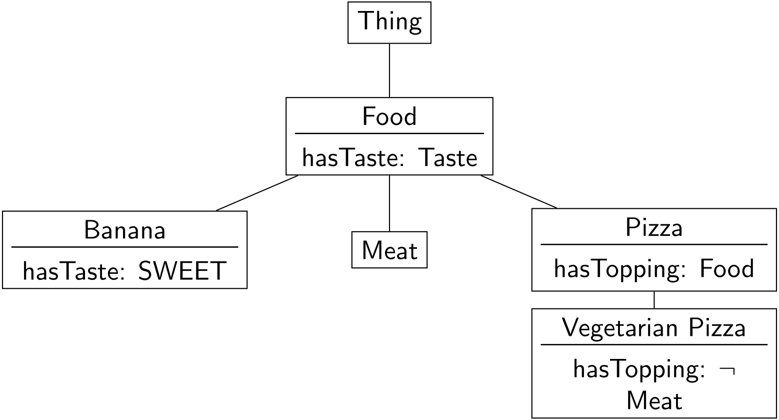
$Woman \sqcap \exists(\geq 3\ (husband \circ brother).Lawyer)$

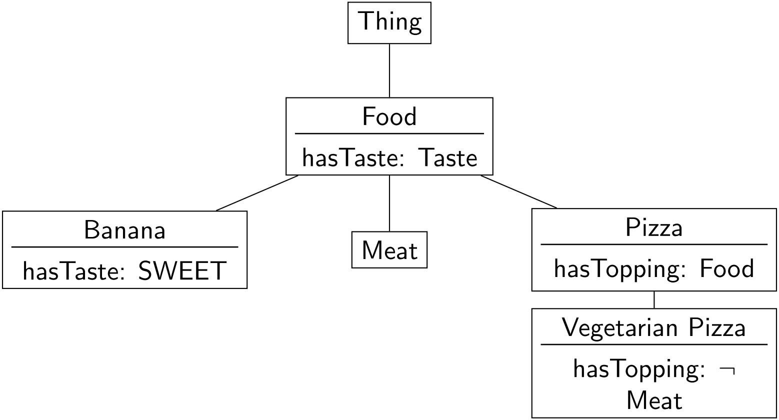
Other helpful resources on Description Logics:

- Short intro to DL (`http://www.edshare.soton.ac.uk/8844/`)
- Course on DL (`http://www.inf.unibz.it/~franconi/dl/course/`)
- Course on Knowledge Representation for the Semantic Web (see DL slides, `http://www.semantic-web-book.org/page/KR4SW-12`)

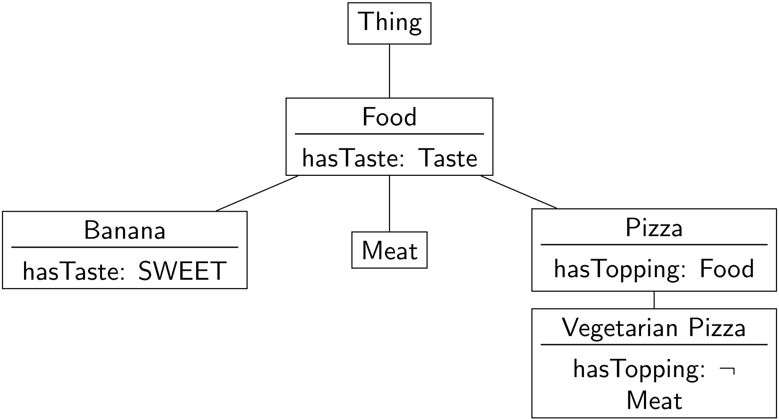
Logic – Food

Translate the following class diagram into formulas in description logic:

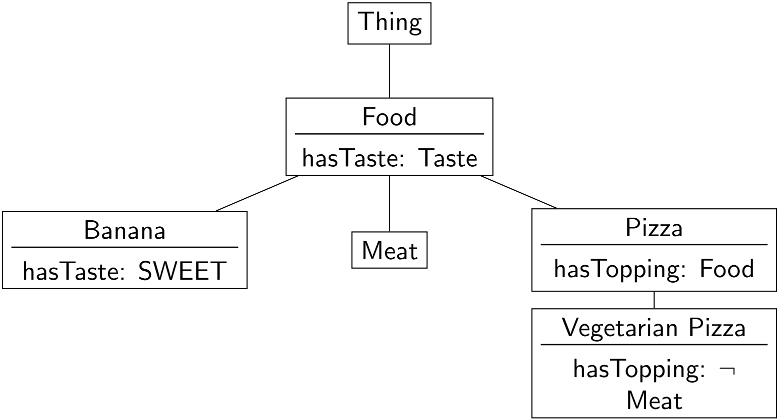




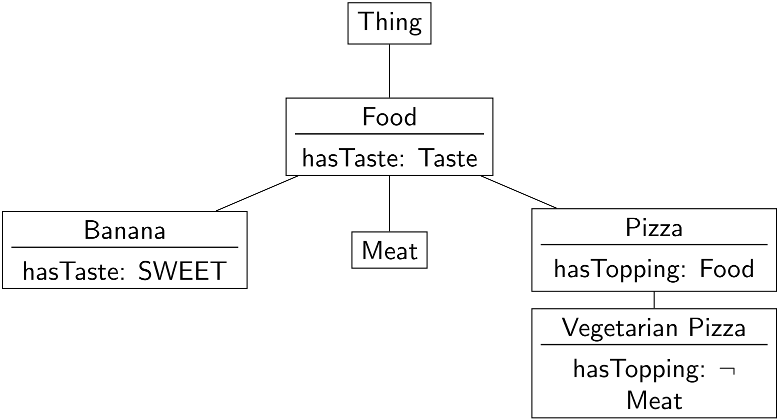
$$Food \sqsubseteq Thing \sqcap \exists hasTaste.Taste$$



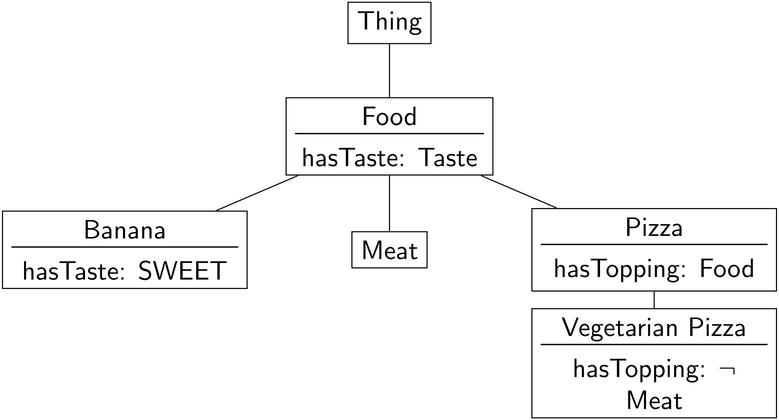
$$Food \sqsubseteq Thing \sqcap \exists hasTaste.Taste$$
$$Banana \sqsubseteq Food \sqcap \exists hasTaste.\{SWEET\}$$



$$Food \sqsubseteq Thing \sqcap \exists hasTaste.Taste$$
$$Banana \sqsubseteq Food \sqcap \exists hasTaste.\{SWEET\}$$
$$Meat \sqsubseteq Food$$



$Food \sqsubseteq Thing \sqcap \exists hasTaste.Taste$
$Banana \sqsubseteq Food \sqcap \exists hasTaste.\{SWEET\}$
$Meat \sqsubseteq Food$
$Pizza \sqsubseteq Food \sqcap \exists hasTopping.Food$



$Food \sqsubseteq Thing \sqcap \exists hasTaste.Taste$
$Banana \sqsubseteq Food \sqcap \exists hasTaste.\{SWEET\}$
$Meat \sqsubseteq Food$
$Pizza \sqsubseteq Food \sqcap \exists hasTopping.Food$
$VegetarianPizza \sqsubseteq Pizza \sqcap \forall hasTopping.\neg Meat$

Logic — Pizza

The following information is given:

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.
2. Each PizzaTopping is a CheeseTopping, MeatTopping or VegetableTopping.
3. Each Pizza is either a VegetarianPizza or a MeatyPizza.
4. John's Pizza has a MozzarellaTopping and a TomatoTopping.
5. MushroomTopping, TomatoTopping and OliveTopping are VegetableToppings.
6. AmericanPizzaBase is a kind of PizzaBase.
7. Each Pizza has a PizzaTopping and a PizzaBase.
8. MozzarellaTopping is a CheeseTopping.
9. Jane's Pizza has a MushroomTopping and an AmericanPizzaBase.

Determine which information is to be represented in the TBOX or the ABOX respectively and formulate the statements in description logic.

Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

$$PizzaMargherita \sqsubseteq Pizza$$
$$PizzaMargherita \sqsubseteq$$
$$\exists hasTopping.TomatoTopping \sqcap \exists hasTopping.MozzarellaTopping$$
$$PizzaMargherita \sqsubseteq \forall hasTopping.(TomatoTopping \sqcup MozzarellaTopping)$$

Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

   $PizzaMargherita \sqsubseteq Pizza$
   $PizzaMargherita \sqsubseteq$
   $\exists hasTopping.TomatoTopping \sqcap \exists hasTopping.MozzarellaTopping$
   $PizzaMargherita \sqsubseteq \forall hasTopping.(TomatoTopping \sqcup MozzarellaTopping)$

2. Each PizzaTopping is a CheeseTopping, MeatTopping or VegetableTopping.

Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

   $$PizzaMargherita \sqsubseteq Pizza$$
   $$PizzaMargherita \sqsubseteq$$
   $$\exists hasTopping.TomatoTopping \sqcap \exists hasTopping.MozzarellaTopping$$
   $$PizzaMargherita \sqsubseteq \forall hasTopping.(TomatoTopping \sqcup MozzarellaTopping)$$

2. Each PizzaTopping is a CheeseTopping, MeatTopping or VegetableTopping.

   $$PizzaTopping \doteq CheeseTopping \sqcup MeatTopping \sqcup VegetableTopping$$

Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

   $PizzaMargherita \sqsubseteq Pizza$
   $PizzaMargherita \sqsubseteq$
   $\exists hasTopping.TomatoTopping \sqcap \exists hasTopping.MozzarellaTopping$
   $PizzaMargherita \sqsubseteq \forall hasTopping.(TomatoTopping \sqcup MozzarellaTopping)$

2. Each PizzaTopping is a CheeseTopping, MeatTopping or VegetableTopping.

   $PizzaTopping \doteq CheeseTopping \sqcup MeatTopping \sqcup VegetableTopping$

3. Each Pizza is either a VegetarianPizza or a MeatyPizza.

Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

   $PizzaMargherita \sqsubseteq Pizza$
   $PizzaMargherita \sqsubseteq$
   $\exists hasTopping.TomatoTopping \sqcap \exists hasTopping.MozzarellaTopping$
   $PizzaMargherita \sqsubseteq \forall hasTopping.(TomatoTopping \sqcup MozzarellaTopping)$

2. Each PizzaTopping is a CheeseTopping, MeatTopping or VegetableTopping.

   $PizzaTopping \doteq CheeseTopping \sqcup MeatTopping \sqcup VegetableTopping$

3. Each Pizza is either a VegetarianPizza or a MeatyPizza.

   $Pizza \doteq VegetarianPizza \sqcup MeatyPizza$

Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

   $PizzaMargherita \sqsubseteq Pizza$
   $PizzaMargherita \sqsubseteq$
   $\exists hasTopping.TomatoTopping \sqcap \exists hasTopping.MozzarellaTopping$
   $PizzaMargherita \sqsubseteq \forall hasTopping.(TomatoTopping \sqcup MozzarellaTopping)$

2. Each PizzaTopping is a CheeseTopping, MeatTopping or VegetableTopping.

   $PizzaTopping \doteq CheeseTopping \sqcup MeatTopping \sqcup VegetableTopping$

3. Each Pizza is either a VegetarianPizza or a MeatyPizza.

   $Pizza \doteq VegetarianPizza \sqcup MeatyPizza$

4. John's Pizza has a MozzarellaTopping and a TomatoTopping.

Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

   $$PizzaMargherita \sqsubseteq Pizza$$
   $$PizzaMargherita \sqsubseteq$$
   $$\exists hasTopping.TomatoTopping \sqcap \exists hasTopping.MozzarellaTopping$$
   $$PizzaMargherita \sqsubseteq \forall hasTopping.(TomatoTopping \sqcup MozzarellaTopping)$$

2. Each PizzaTopping is a CheeseTopping, MeatTopping or VegetableTopping.

   $$PizzaTopping \doteq CheeseTopping \sqcup MeatTopping \sqcup VegetableTopping$$

3. Each Pizza is either a VegetarianPizza or a MeatyPizza.

   $$Pizza \doteq VegetarianPizza \sqcup MeatyPizza$$

4. John's Pizza has a MozzarellaTopping and a TomatoTopping.

   $$JOHNSPIZZA : Pizza$$
   $$MOZARELLA : MozzarellaTopping$$
   $$TOMATO : TomatoTopping$$
   $$(JOHNSPIZZA, MOZZARELLA) : hasTopping$$
   $$(JOHNSPIZZA, TOMATO) : hasTopping$$

Logic — Pizza

5. MushroomTopping, TomatoTopping and OliveTopping are VegetableToppings.

Logic — Pizza

5. MushroomTopping, TomatoTopping and OliveTopping are VegetableToppings.

   $MushroomTopping \sqsubseteq VegetableTopping$
   $TomatoTopping \sqsubseteq VegetableTopping$
   $OliveTopping \sqsubseteq VegetableTopping$

Logic — Pizza

5. MushroomTopping, TomatoTopping and OliveTopping are VegetableToppings.

   $MushroomTopping \sqsubseteq VegetableTopping$
   $TomatoTopping \sqsubseteq VegetableTopping$
   $OliveTopping \sqsubseteq VegetableTopping$

6. AmericanPizzaBase is a kind of PizzaBase.

Logic — Pizza

5. MushroomTopping, TomatoTopping and OliveTopping are VegetableToppings.

   $MushroomTopping \sqsubseteq VegetableTopping$
   $TomatoTopping \sqsubseteq VegetableTopping$
   $OliveTopping \sqsubseteq VegetableTopping$

6. AmericanPizzaBase is a kind of PizzaBase.

   $AmericanPizzaBase \sqsubseteq PizzaBase$

Logic — Pizza

5. MushroomTopping, TomatoTopping and OliveTopping are VegetableToppings.

   $MushroomTopping \sqsubseteq VegetableTopping$
   $TomatoTopping \sqsubseteq VegetableTopping$
   $OliveTopping \sqsubseteq VegetableTopping$

6. AmericanPizzaBase is a kind of PizzaBase.

   $AmericanPizzaBase \sqsubseteq PizzaBase$

7. Each Pizza has a PizzaTopping and a PizzaBase.

Logic — Pizza

5. MushroomTopping, TomatoTopping and OliveTopping are VegetableToppings.

   $MushroomTopping \sqsubseteq VegetableTopping$
   $TomatoTopping \sqsubseteq VegetableTopping$
   $OliveTopping \sqsubseteq VegetableTopping$

6. AmericanPizzaBase is a kind of PizzaBase.

   $AmericanPizzaBase \sqsubseteq PizzaBase$

7. Each Pizza has a PizzaTopping and a PizzaBase.

   $Pizza \sqsubseteq \exists hasTopping.PizzaTopping$
   $Pizza \sqsubseteq \exists hasBase.PizzaBase$

Logic — Pizza

8. MozzarellaTopping is a CheeseTopping.

Logic — Pizza

8. MozzarellaTopping is a CheeseTopping.

   $MozzarellaTopping \sqsubseteq CheeseTopping$

Logic — Pizza

8. MozzarellaTopping is a CheeseTopping.

   $MozzarellaTopping \sqsubseteq CheeseTopping$

9. Jane's Pizza has a MushroomTopping and an AmericanPizzaBase.

Logic — Pizza

8. MozzarellaTopping is a CheeseTopping.

   $MozzarellaTopping \sqsubseteq CheeseTopping$

9. Jane's Pizza has a MushroomTopping and an AmericanPizzaBase.

   $JANESPIZZA : Pizza$
   $MUSHROOM : MushroomTopping$
   $BASE : AmericanPizzaBase$
   $(JANESPIZZA, MUSHROOM) : hasTopping$
   $(JANESPIZZA, BASE) : hasBase$

TBox

$$Pizza \doteq VegetarianPizza \sqcup MeatyPizza$$
$$PizzaTopping \doteq CheeseTopping \sqcup MeatTopping \sqcup VegetableTopping$$
$$Pizza \sqsubseteq \exists hasTopping.PizzaTopping$$
$$Pizza \sqsubseteq \exists hasBase.PizzaBase$$
$$MushroomTopping \sqsubseteq VegetableTopping$$
$$TomatoTopping \sqsubseteq VegetableTopping$$
$$OliveTopping \sqsubseteq VegetableTopping$$
$$MozzarellaTopping \sqsubseteq CheeseTopping$$
$$AmericanPizzaBase \sqsubseteq PizzaBase$$
$$PizzaMargherita \sqsubseteq Pizza$$
$$PizzaMargherita \sqsubseteq \exists hasTopping.TomatoTopping \sqcap \exists hasTopping.MozzarellaTopping$$
$$PizzaMargherita \sqsubseteq \forall hasTopping.(TomatoTopping \sqcup MozzarellaTopping)$$

ABox

$$JOHNSPIZZA : Pizza$$
$$MOZARELLA : MozzarellaTopping$$
$$TOMATO : TomatoTopping$$
$$(JOHNSPIZZA, MOZZARELLA) : hasTopping$$
$$(JOHNSPIZZA, TOMATO) : hasTopping$$
$$JANESPIZZA : Pizza$$
$$MUSHROOM : MushroomTopping$$
$$BASE : AmericanPizzaBase$$
$$(JANESPIZZA, MUSHROOM) : hasTopping$$
$$(JANESPIZZA, BASE) : hasBase$$

MLN

An MLN defines a probability distribution over all possible worlds given by:

$$P(X = x) = \frac{1}{Z} \cdot \exp\left(\sum_i w_i \cdot n_i(x)\right) = \frac{\exp\left(\sum_i w_i \cdot n_i(x)\right)}{\sum_{x' \in \mathcal{X}} \exp\left(\sum_i w_i \cdot n_i(x')\right)} \quad (1)$$

where $n_i(x)$ is the number of true instantiations of formula F_i in world x.

MLN

We look at a variant of the Smoker-MLN. Design a MLN that represents a distribution characterized by the following statements:

- Smoking leads to cancer in $2/3 = 66.\overline{6}\%$ of the cases
- Chewing smokeless tobacco leads to cancer in in $2/3$ of the cases
- Smoking cigarettes and chewing smokeless tobacco cause cancer *independent* of each other, i.e., somebody doing both increases the risk of getting cancer to $P(\text{cancerFromSmoking} \vee \text{cancerFromSmoking}) = 8/9 = 0.\overline{8}$ (Noisy-Or).
- People, who do not consume tobacco, will never get cancer.

All other aspects of the distribution are subject to the principle of maximal entropy, that is, if the statements above do not apply a uniform distribution is assumed.

MLN

What about this solution?

```
smoking(person)
chewing(person)
cancer(person)

log(2.0/3)   cancer(x) ^  chewing(x)
log(1.0/3)  !cancer(x) ^  chewing(x)
log(0)       cancer(x) ^ !chewing(x)
log(1)      !cancer(x) ^ !chewing(x)

log(2.0/3)   cancer(x) ^  smoking(x)
log(1.0/3)  !cancer(x) ^  smoking(x)
log(0)       cancer(x) ^ !smoking(x)
log(1)      !cancer(x) ^ !smoking(x)
```

MLN

If we look at a situation of somebody who smokes *and* chews, we recognize that we don't get the probability of cancer of 8/9 but
$(2/3)^2/((2/3)^2 + (1/3)^2) = 0.80$.
The right probability is generated by applying a stochastic "or" (*Noisy-or*)

MLN

Here is a MLN that fulfills that the requirements of the exercise:

```
smoking(person)
chewing(person)
cancerFromSmoking(person)
cancerFromChewing(person)
cancer(person)

logx(2.0/3)  cancerFromChewing(x) ^  chewing(x)
logx(1.0/3) !cancerFromChewing(x) ^  chewing(x)
logx(0)      cancerFromChewing(x) ^ !chewing(x)
logx(1)     !cancerFromChewing(x) ^ !chewing(x)

logx(2.0/3)  cancerFromSmoking(x) ^  smoking(x)
logx(1.0/3) !cancerFromSmoking(x) ^  smoking(x)
logx(0)      cancerFromSmoking(x) ^ !smoking(x)
logx(1)     !cancerFromSmoking(x) ^ !smoking(x)
cancer(x) <=> cancerFromSmoking(x) v cancerFromChewing(x).
```

MLN 2

Prove or disprove the following statement:

Adding an arbitrary constant $c \in \mathbb{R}$ to all weights in a MLN do not influence its distribution.

Does the statement hold? Why/Why not? And are there special cases?

MLN 2

The statement doesn't hold. Example:

```
foo(object)
log(2) foo(x)
```

If we add, e.g., 1 to the formula in this MLN the probability after marginalization for `foo(X)` and for an arbitrary `X` will change from $2/3$ to $2e/(2e+1)$.

Questions?