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:

\(-\ is\text{Married}(x,y)\)
\(-\ Alive(x)\)
\(-\ has\text{Child}(x,y)\)
Predicate Logic

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

\[ \text{Spouse}(x) \iff \exists y (\text{isMarried}(x,y) \land \text{Alive}(y)) \]
Predicate Logic

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

Spouse(x) ⇔ ∃y(isMarried(x, y) ∧ Alive(y))

Bigamist(x) ⇔ ∃y, z(isMarried(x, y) ∧ Alive(y) ∧ isMarried(x, z) ∧ Alive(z) ∧ ¬(y = z) ∧ ¬∃w(isMarried(x, w) ∧ Alive(w) ∧ ¬(y = w) ∧ ¬(z = w)))
Predicate Logic

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

\[
\text{Spouse}(x) \iff \exists y (\text{isMarried}(x, y) \land \text{Alive}(y))
\]

\[
\text{Bigamist}(x) \iff \exists y, z (\text{isMarried}(x, y) \land \text{Alive}(y) \land \text{isMarried}(x, z) \land \text{Alive}(z) \land \neg(y = z) \land \neg \exists w (\text{isMarried}(x, w) \land \text{Alive}(w) \land \neg(y = w) \land \neg(z = w)))
\]

\[
\text{Polygamist}(x) \iff \\
\exists y, z (\text{isMarried}(x, y) \land \text{Alive}(y) \land \text{isMarried}(x, z) \land \text{Alive}(z) \land \neg(y = z))
\]
Predicate Logic

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

\[
\text{Spouse}(x) \iff \exists y (\text{isMarried}(x, y) \land \text{Alive}(y))
\]

\[
\text{Bigamist}(x) \iff \exists y, z (\text{isMarried}(x, y) \land \text{Alive}(y) \land \text{isMarried}(x, z) \land \text{Alive}(z) \land \neg(y = z) \land \neg \exists w (\text{isMarried}(x, w) \land \text{Alive}(w) \land \neg(y = w) \land \neg(z = w))
\]

\[
\text{Polygamist}(x) \iff \\
\exists y, z (\text{isMarried}(x, y) \land \text{Alive}(y) \land \text{isMarried}(x, z) \land \text{Alive}(z) \land \neg(y = z))
\]

\[
\text{Widow}(x) \iff \exists y (\text{isMarried}(x, y) \land \neg \text{Alive}(y))
\]
Predicate Logic

Define Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent by using \textit{isMarried}(x,y), \textit{Alive}(x) and \textit{hasChild}(x,y)

\begin{align*}
\text{Spouse}(x) & \iff \exists y (\text{isMarried}(x,y) \land \text{Alive}(y)) \\
\text{Bigamist}(x) & \iff \exists y, z (\text{isMarried}(x,y) \land \text{Alive}(y) \land \text{isMarried}(x,z) \land \text{Alive}(z) \land \neg(y = z) \land \neg\exists w (\text{isMarried}(x,w) \land \text{Alive}(w) \land \neg(y = w) \land \neg(z = w))) \\
\text{Polygamist}(x) & \iff \\
& \exists y, z (\text{isMarried}(x,y) \land \text{Alive}(y) \land \text{isMarried}(x,z) \land \text{Alive}(z) \land \neg(y = z)) \\
\text{Widow}(x) & \iff \exists y (\text{isMarried}(x,y) \land \neg\text{Alive}(y)) \\
\text{Bachelor}(x) & \iff \neg\exists y (\text{isMarried}(x,y))
\end{align*}
Predicate Logic

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

*Spouse*(x) ⇔ ∃y(*isMarried*(x, y) ∧ *Alive*(y))

*Bigamist*(x) ⇔ ∃y, z(*isMarried*(x, y) ∧ *Alive*(y) ∧ *isMarried*(x, z) ∧ *Alive*(z) ∧ ¬(y = z) ∧ ¬∃w(*isMarried*(x, w) ∧ *Alive*(w) ∧ ¬(y = w) ∧ ¬(z = w)))

*Polygamist*(x) ⇔ ∃y, z(*isMarried*(x, y) ∧ *Alive*(y) ∧ *isMarried*(x, z) ∧ *Alive*(z) ∧ ¬(y = z))

*Widow*(x) ⇔ ∃y(*isMarried*(x, y) ∧ ¬*Alive*(y))

*Bachelor*(x) ⇔ ¬∃y(*isMarried*(x, y))

*Stepparent*(x) ⇔ ∃y, z(*isMarried*(x, y) ∧ *hasChild*(y, z) ∧ ¬*hasChild*(x, z))
**Description Logic**

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

\[ \text{Spouse} \equiv \exists \text{isMarried}.\text{Alive} \]
Description Logic

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

Spouse ≜ ∃isMarried⋅Alive

Polygamist ≜ (≥ 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* \( \doteq \exists \text{isMarried}. \text{Alive} \)

*Polygamist* \( \doteq (\geq 2 \text{isMarried}. \text{Alive}) \)

*Bigamist* \( \doteq \text{Polygamist} \sqcap (\leq 2 \text{isMarried}. \text{Alive}) \)
Description Logic

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

*Spouse* $\vdash \exists isMarried . Alive$

*Polygamist* $\vdash (\geq 2 isMarried . Alive)$

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

*Widow* $\vdash \exists isMarried . \neg Alive$
Define Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent by using $\text{isMarried}(x,y)$, $\text{Alive}(x)$ and $\text{hasChild}(x,y)$

- $\text{Spouse} \equiv \exists\text{isMarried}.\text{Alive}$
- $\text{Polygamist} \equiv (\geq 2 \text{isMarried}.\text{Alive})$
- $\text{Bigamist} \equiv \text{Polygamist} \sqcap (\leq 2 \text{isMarried}.\text{Alive})$
- $\text{Widow} \equiv \exists\text{isMarried}.\neg\text{Alive}$
- $\text{Bachelor} \equiv \forall\text{isMarried}.\bot$ or $\text{Bachelor} \equiv \neg\exists\text{isMarried}.\top$
Description Logic

Define Spouse, Bigamist, Polygamist, Widow(er), Bachelor, Stepparent by using $\text{isMarried}(x,y)$, $\text{Alive}(x)$ and $\text{hasChild}(x,y)$

$$\text{Spouse} \doteq \exists \text{isMarried}.\text{Alive}$$

$$\text{Polygamist} \doteq (\geq 2 \text{isMarried}.\text{Alive})$$

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

$$\text{Widow} \doteq \exists \text{isMarried}.\neg\text{Alive}$$

$$\text{Bachelor} \doteq \forall \text{isMarried}.\bot \text{ or } \text{Bachelor} \doteq \neg\exists \text{isMarried}.\top$$

$$\text{Stepparent} \doteq \exists((\text{isMarried} \circ \text{hasChild}) \sqcap \neg\text{hasChild}).\top$$
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) ⇔ ∃y, z(isMarried(x, y) ∧ hasChild(y, z) ∧ ¬hasChild(x, z))

DL: Stepparent = ∃((isMarried ◦ hasChild) △ ¬hasChild).⊤

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

Woman △ ∃(≥ 3 (husband ◦ 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)
Translate the following class diagram into formulas in description logic:

- **Thing**
- **Food**
  - hasTaste: Taste
  - **Banana**
    - hasTaste: SWEET
  - **Meat**
  - **Pizza**
    - hasTopping: Food
    - **Vegetarian Pizza**
      - hasTopping: ¬Meat
Food ⊆ Thing ⊓ ∃ hasTaste. Taste
Food ⊑ Thing ⊓ ∃ hasTaste. Taste
Banana ⊑ Food ⊓ ∃ hasTaste. {SWEET}
Food ⊑ Thing ⊓ ∀∃hasTaste. Taste
Banana ⊑ Food ⊓ ∃hasTaste. {SWEET}
Meat ⊑ Food
Food ⊆ Thing \(∩\) \(∃\) hasTaste. Taste

Banana ⊆ Food \(∩\) \(∃\) hasTaste. \{SWEET\}

Meat ⊆ Food

Pizza ⊆ Food \(∩\) \(∃\) hasTopping. Food
Food ⊑ Thing ∩ ∃hasTaste.Taste
Banana ⊑ Food ∩ ∃hasTaste.\{SWEET\}
Meat ⊑ Food
Pizza ⊑ Food ∩ ∃hasTopping.Food
VegetarianPizza ⊑ Pizza ∩ ∀hasTopping.¬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.
1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.
Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

   \[\text{PizzaMargherita} \sqsubseteq \text{Pizza} \]
   \[\text{PizzaMargherita} \sqsubseteq \exists \text{hasTopping}. \text{TomatoTopping} \sqcap \exists \text{hasTopping}. \text{MozzarellaTopping} \]
   \[\text{PizzaMargherita} \sqsubseteq \forall \text{hasTopping}. (\text{TomatoTopping} \sqcup \text{MozzarellaTopping}) \]
Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

   \[ \text{PizzaMargherita} \sqsubseteq \text{Pizza} \]
   \[ \text{PizzaMargherita} \sqsubseteq \exists \text{hasTopping}. \text{TomatoTopping} \sqcap \exists \text{hasTopping}. \text{MozzarellaTopping} \]
   \[ \text{PizzaMargherita} \sqsubseteq \forall \text{hasTopping}. (\text{TomatoTopping} \sqcup \text{MozzarellaTopping}) \]

2. Each PizzaTopping is a CheeseTopping, MeatTopping or VegetableTopping.
Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

   \[
   \text{PizzaMargherita} \sqsubseteq \text{Pizza} \\
   \text{PizzaMargherita} \sqsubseteq \exists \text{hasTopping}. \text{TomatoTopping} \sqcap \exists \text{hasTopping}. \text{MozzarellaTopping} \\
   \text{PizzaMargherita} \sqsubseteq \forall \text{hasTopping} . (\text{TomatoTopping} \sqcup \text{MozzarellaTopping})
   \]

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

   \[
   \text{PizzaTopping} \doteq \text{CheeseTopping} \sqcup \text{MeatTopping} \sqcup \text{VegetableTopping}
   \]
Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

\[
\text{PizzaMargherita} \sqsubseteq \text{Pizza} \\
\text{PizzaMargherita} \sqsubseteq \exists \text{hasTopping} . \text{TomatoTopping} \sqcap \exists \text{hasTopping} . \text{MozzarellaTopping} \\
\text{PizzaMargherita} \sqsubseteq \forall \text{hasTopping} . (\text{TomatoTopping} \sqcup \text{MozzarellaTopping})
\]

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

\[
\text{PizzaTopping} \doteq \text{CheeseTopping} \sqcup \text{MeatTopping} \sqcup \text{VegetableTopping}
\]

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

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

   \[ \text{PizzaMargherita} \sqsubseteq \text{Pizza} \]
   \[ \text{PizzaMargherita} \sqsubseteq \exists \text{hasTopping} \cdot \text{TomatoTopping} \sqcap \exists \text{hasTopping} \cdot \text{MozzarellaTopping} \]
   \[ \text{PizzaMargherita} \sqsubseteq \forall \text{hasTopping} \cdot (\text{TomatoTopping} \sqcup \text{MozzarellaTopping}) \]

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

   \[ \text{PizzaTopping} \equiv \text{CheeseTopping} \sqcup \text{MeatTopping} \sqcup \text{VegetableTopping} \]

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

   \[ \text{Pizza} \equiv \text{VegetarianPizza} \sqcup \text{MeatyPizza} \]
Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

\[ \text{PizzaMargherita} \subseteq \text{Pizza} \]
\[ \text{PizzaMargherita} \subseteq \exists \text{hasTopping}. \text{TomatoTopping} \sqcap \exists \text{hasTopping}. \text{MozzarellaTopping} \]
\[ \text{PizzaMargherita} \subseteq \forall \text{hasTopping}. (\text{TomatoTopping} \sqcup \text{MozzarellaTopping}) \]

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

\[ \text{PizzaTopping} \models \text{CheeseTopping} \sqcup \text{MeatTopping} \sqcup \text{VegetableTopping} \]

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

\[ \text{Pizza} \models \text{VegetarianPizza} \sqcup \text{MeatyPizza} \]

4. John’s Pizza has a MozzarellaTopping and a TomatoTopping.
Logic — Pizza

1. A PizzaMargherita has only TomatoTopping and MozzarellaTopping.

\[ \text{PizzaMargherita} \sqsubseteq \text{Pizza} \]
\[ \text{PizzaMargherita} \sqsubseteq \exists \text{hasTopping}. \text{TomatoTopping} \sqcap \exists \text{hasTopping}. \text{MozzarellaTopping} \]
\[ \text{PizzaMargherita} \sqsubseteq \forall \text{hasTopping}. (\text{TomatoTopping} \sqcup \text{MozzarellaTopping}) \]

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

\[ \text{PizzaTopping} \models \text{CheeseTopping} \sqcup \text{MeatTopping} \sqcup \text{VegetableTopping} \]

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

\[ \text{Pizza} \models \text{VegetarianPizza} \sqcup \text{MeatyPizza} \]

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

\[ \text{JOHNSPIZZA} : \text{Pizza} \]
\[ \text{MOZARELLA} : \text{MozzarellaTopping} \]
\[ \text{TOMATO} : \text{TomatoTopping} \]
\[ (\text{JOHNSPIZZA}, \text{MOZARELLA}) : \text{hasTopping} \]
\[ (\text{JOHNSPIZZA}, \text{TOMATO}) : \text{hasTopping} \]
5. MushroomTopping, TomatoTopping and OliveTopping are VegetableToppings.
5. MushroomTopping, TomatoTopping and OliveTopping are VegetableToppings.

\[
\begin{align*}
\text{Mushroom Topping} & \subseteq \text{Vegetable Topping} \\
\text{Tomato Topping} & \subseteq \text{Vegetable Topping} \\
\text{Olive Topping} & \subseteq \text{Vegetable Topping}
\end{align*}
\]
5. MushroomTopping, TomatoTopping and OliveTopping are VegetableToppings.

\[
\begin{align*}
\text{MushroomTopping} & \subseteq \text{VegetableTopping} \\
\text{TomatoTopping} & \subseteq \text{VegetableTopping} \\
\text{OliveTopping} & \subseteq \text{VegetableTopping}
\end{align*}
\]

6. AmericanPizzaBase is a kind of PizzaBase.
Logic — Pizza

5. MushroomTopping, TomatoTopping and OliveTopping are VegetableToppings.

\[
\begin{align*}
\text{MushroomTopping} &\sqsubseteq \text{VegetableTopping} \\
\text{TomatoTopping} &\sqsubseteq \text{VegetableTopping} \\
\text{OliveTopping} &\sqsubseteq \text{VegetableTopping}
\end{align*}
\]

6. AmericanPizzaBase is a kind of PizzaBase.

\[
\begin{align*}
\text{AmericanPizzaBase} &\sqsubseteq \text{PizzaBase}
\end{align*}
\]
5. MushroomTopping, TomatoTopping and OliveTopping are VegetableToppings.

\[
\begin{align*}
&MushroomTopping \sqsubseteq VegetableTopping \\
&TomatoTopping \sqsubseteq VegetableTopping \\
&OliveTopping \sqsubseteq VegetableTopping
\end{align*}
\]

6. AmericanPizzaBase is a kind of PizzaBase.

\[
AmericanPizzaBase \sqsubseteq PizzaBase
\]

7. Each Pizza has a PizzaTopping and a PizzaBase.
5. MushroomTopping, TomatoTopping and OliveTopping are VegetableToppings.

\[ \text{MushroomTopping} \sqsubseteq \text{VegetableTopping} \]
\[ \text{TomatoTopping} \sqsubseteq \text{VegetableTopping} \]
\[ \text{OliveTopping} \sqsubseteq \text{VegetableTopping} \]

6. AmericanPizzaBase is a kind of PizzaBase.

\[ \text{AmericanPizzaBase} \sqsubseteq \text{PizzaBase} \]

7. Each Pizza has a PizzaTopping and a PizzaBase.

\[ \text{Pizza} \sqsubseteq \exists \text{hasTopping}. \text{PizzaTopping} \]
\[ \text{Pizza} \sqsubseteq \exists \text{hasBase}. \text{PizzaBase} \]
8. MozzarellaTopping is a CheeseTopping.
8. MozzarellaTopping is a CheeseTopping.

\[ \text{MozzarellaTopping} \sqsubseteq \text{CheeseTopping} \]
8. MozzarellaTopping is a CheeseTopping.

\[ \text{MozzarellaTopping} \sqsubseteq \text{CheeseTopping} \]

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

\[ \text{JANESPIZZA} : \text{Pizza} \]
\[ \text{MUSHROOM} : \text{MushroomTopping} \]
\[ \text{BASE} : \text{AmericanPizzaBase} \]
\[ \text{(JANESPIZZA, MUSHROOM)} : \text{hasTopping} \]
\[ \text{(JANESPIZZA, BASE)} : \text{hasBase} \]
8. MozzarellaTopping is a CheeseTopping.

\[
\text{MozzarellaTopping} \sqsubseteq \text{CheeseTopping}
\]

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

\[
\begin{align*}
\text{JANESPIZZA} : & \text{Pizza} \\
\text{MUSHROOM} : & \text{MushroomTopping} \\
\text{BASE} : & \text{AmericanPizzaBase} \\
(\text{JANESPIZZA}, \text{MUSHROOM}) : & \text{hasTopping} \\
(\text{JANESPIZZA}, \text{BASE}) : & \text{hasBase}
\end{align*}
\]
**TBox**

\[
\text{Pizza} \triangleq \text{VegetarianPizza} \sqcup \text{MeatyPizza} \\
\text{PizzaTopping} \triangleq \text{CheeseTopping} \sqcup \text{MeatTopping} \sqcup \text{VegetableTopping} \\
\text{Pizza} \sqsubseteq \exists\text{hasTopping}.\text{PizzaTopping} \\
\text{Pizza} \sqsubseteq \exists\text{hasBase.}\text{PizzaBase} \\
\text{MushroomTopping} \sqsubseteq \text{VegetableTopping} \\
\text{TomatoTopping} \sqsubseteq \text{VegetableTopping} \\
\text{OliveTopping} \sqsubseteq \text{VegetableTopping} \\
\text{MozzarellaTopping} \sqsubseteq \text{CheeseTopping} \\
\text{AmericanPizzaBase} \sqsubseteq \text{PizzaBase} \\
\text{PizzaMargherita} \sqsubseteq \text{Pizza} \\
\text{PizzaMargherita} \sqsubseteq \exists\text{hasTopping.}\text{TomatoTopping} \sqcap \exists\text{hasTopping.}\text{MozzarellaTopping} \\
\text{PizzaMargherita} \sqsubseteq \forall\text{hasTopping.}(\text{TomatoTopping} \sqcup \text{MozzarellaTopping})
\]
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
An MLN defines a probability distribution over all possible worlds given by:

\[
P(X = x) = \frac{1}{Z} \cdot \exp \left( \sum_i w_i \cdot n_i(x) \right)
\]

\[
= \frac{\exp \left( \sum_i w_i \cdot n_i(x) \right)}{\sum_{x' \in X} \exp \left( \sum_i w_i \cdot n_i(x') \right)}
\]

(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 $\frac{2}{3} = 66.6\%$ of the cases
- Chewing smokeless tobacco leads to cancer in $\frac{2}{3}$ 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 $P(\text{cancerFromSmoking} \lor \text{cancerFromSmoking}) = \frac{8}{9} = 0.88$ (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.
What about this solution?

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

```
\log(2.0/3) \ cancer(x) \land \neg chewing(x)
\log(1.0/3) \neg cancer(x) \land \neg chewing(x)
\log(0) \ cancer(x) \land \neg !chewing(x)
\log(1) \neg cancer(x) \land \neg !chewing(x)
```

```
\log(2.0/3) \ cancer(x) \land \neg smoking(x)
\log(1.0/3) \neg cancer(x) \land \neg smoking(x)
\log(0) \ cancer(x) \land \neg !smoking(x)
\log(1) \neg cancer(x) \land \neg !smoking(x)
```
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:

\[
\text{smoking(person)} \\
\text{chewing(person)} \\
\text{cancerFromSmoking(person)} \\
\text{cancerFromChewing(person)} \\
\text{cancer(person)}
\]

\[
\log(x) \frac{2.0}{3} \text{ cancerFromChewing}(x) \land \neg \text{ chewing}(x) \\
\log(x) \frac{1.0}{3} \neg \text{ cancerFromChewing}(x) \land \neg \text{ chewing}(x) \\
\log(x) 0 \text{ cancerFromChewing}(x) \land \neg \text{ chewing}(x) \\
\log(x) 1 \neg \text{ cancerFromChewing}(x) \land \neg \text{ chewing}(x)
\]

\[
\log(x) \frac{2.0}{3} \text{ cancerFromSmoking}(x) \land \neg \text{ smoking}(x) \\
\log(x) \frac{1.0}{3} \neg \text{ cancerFromSmoking}(x) \land \neg \text{ smoking}(x) \\
\log(x) 0 \text{ cancerFromSmoking}(x) \land \neg \text{ smoking}(x) \\
\log(x) 1 \neg \text{ cancerFromSmoking}(x) \land \neg \text{ smoking}(x)
\]

\[
\text{cancer}(x) \iff \text{ cancerFromSmoking}(x) \lor \text{ cancerFromChewing}(x).
\]
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?
The statement doesn't hold. Example:

\[
\begin{align*}
\text{foo}(\text{object}) \\
\log(2) \text{foo}(x)
\end{align*}
\]

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