$$\pi_x(\forall r.C) = \forall y(r(x,y) \rightarrow \pi_y(C))$$
$$\pi_x(\exists r.C) = \exists y(r(x,y) \wedge \pi_y(C))$$
$$\pi_y(\forall r.C) = \forall x(r(y,x) \rightarrow \pi_x(C))$$
$$\pi_y(\exists r.C) = \exists x(r(y,x) \wedge \pi_x(C))$$

- ○ A ⊓ ∀r.B is subsumed by A ⊓ ∃r.B
- ○ A ⊓ ∃r.(B ⊔ C) is subsumed by A ⊓ ∃r.B
- ◉ A ⊓ ∃r.A ⊓ ∀r.B is subsumed by A ⊓ ∃r.B
- ○ A ⊓ ∃r.B is subsumed by A ⊓ ∀r.B

Consider the following TBox.

$A \sqsubseteq \exists R.B$
$A \sqsubseteq \forall R.C$

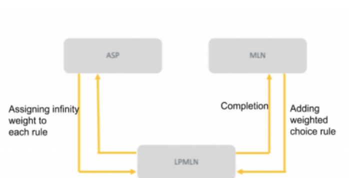
Which option is the First-Order formula that is translated from this TBox?

- ○ $\forall x(A(x) \rightarrow \exists y(R(x,y) \wedge B(y))) \wedge \forall x(A(x) \rightarrow \forall y(R(x,y) \wedge C(y)))$
- ◉ $\forall x(A(x) \rightarrow \exists y(R(x,y) \wedge B(y))) \wedge \forall x(A(x) \rightarrow \forall y(R(x,y) \rightarrow C(y)))$

Weak constraint: LPMLN to ASP

	unsat(2) :- q, not p.
$1 : p$	p :- q, not unsat(2).
$-2 : p \leftarrow q$	
$3 : \perp \leftarrow p$	:~ unsat(2). [-2]

Open world : Absence of information is not interpreted as presence of negative information but simply as lack of knowledge.



Wk 7: Correct: The formalisms for ontology are object-oriented logics.
Categories are organized into taxonomies.
Weak ontology language: UML-diagrams
Properties are like predicates
Statements assert the properties of resources
In the **RDFS**, it allows range restriction and domain restriction.
RDFS provides schema-level alignments.
Prop(r, p, v) -> Type(p, rdfs:Property)
Cyc is a formalized representation of human common sense knowledge
The availability of open sources of knowledge on the web has fueled the development of efforts such as **DBpedi**a, FreeBase.
With **nominals,** {US} ≡ {USA} means US is the same as USA.
Answer set programming is under **Closed World** Assumption.
A **concept** C is satisfiable with respect to a knowledge base K if there exists a model I of K such that CI is not equal to empty
owl:**topObjectProperty** contains all possible role
It is easy to see the class hierarchy in **taxonomies**
SROIQ is a superset of ALC
∃ R.C in description logics can be written as R some C in OWL.
DBpedia automatically evolves as Wikipedia changes.
We can use **reification** to turn a sequence of information into statements in RDF.
Incorrect: An IRI(Internationalized Resource Identifier) can be any object that we want to talk about.
Different **knowledge graphs** use the same ontological primitive in order to avoid ambiguity.
Individual inequality means individual a is not equivalent to individual B written as a
The problem of checking whether the assertion C(a) is satisfied in every model of K

K ⊭ C ≡ ⊥ iff there exists an x such that K ∪{C(x)} is satisfiable.

Wk 6: Probability: P(tautology)=1, P(AVB) = P(A) + P(B) if A and B are mutually exclusive
P(h|e) = P(h^e)/P(e)
Bayes rule: P(h|e) = (P(e|h)* P(h)) / P(e)
Product rule: P(f1 ^ f2) = P(f1) * P(f2|f1)
P(f1 ^ f2 ^ f3...) = P(f1) * P(f2|f1) * P(f3|f1,f2)...
B and A are not marginally independent

$P(B=h) = 0.15$

$P(B=h \mid A=s) = \frac{P(B=h \wedge A=s)}{P(A=s)} = \frac{0.1}{0.4} = 0.25$

$P(B=h \mid A=c) = \frac{P(B=h \wedge A=c)}{P(A=c)} = \frac{0.05}{0.6} = 0.083$

Q: Are C_1 and C_2 marginally independent?

$P(C_1=h) = 0.5$

$P(C_1=h \mid C_2=h) = \frac{P(C_1=h \wedge C_2=h)}{P(C_2=h)} = \frac{0.25}{0.5} = 0.5$

Bayesian network An Augmented, directed acyclic graph, where each node corresponds to a random variable Xi and each edge indicates a direct influence among the random variables.
Each variable is **independent** of its non descendants given its parents
P(T,F,A,S,L,R) = P(T)* P(F|T)*P(A|F,T)*P(S|A,F,T)*P(L|S,A,F,T)*P(R|L,S,A,F,T)...for bayesian network ignore all non descendants i.e. non parents

w: $\forall x\ Smokes(x) \Rightarrow Cancer(x)$ 	ω: Smokes(a) → Cancer(a)

Consider the Herbrand universe is {a}

- I1 = {} 	P(I1) = $\frac{e^{\omega}}{3 \cdot e^{\omega}+1}$ 	$\omega(I_1) = e^{\omega}$
- I2= {Cancer(a)} 	P(I2) = $P(I_1)$ 	$\omega(I_2) = e^{\omega}$
- I3= {Smokes(a)} 	P(I3) = $\frac{e^{0}}{3 \cdot e^{\omega}+1}$ 	$\omega(I_3) = e^{0}$
- I4= {Smokes(a), Cancer(a)} 	P(I4) = $P(I_1)$ 	$\omega(I_4) = e^{\omega}$

Z = $3 \cdot e^{\omega} + 1$

P(Cancer(a)| Smokes(a)) = $\frac{e^{\omega}}{e^{\omega}+1}$

P(Smokes(a)∧ ¬Cancer(a)) = $\frac{1}{3 \cdot e^{\omega}+1}$

π_x(<= 2 R) is equivalent to ∀ y1 y2 y3(R(x, y1) ∧ R(x, y2) ∧ (x, y3) → (y1 = y2 ∨ y2=y3 ∨ y3=y1)).

Wk 6: For any **MLN program** \Pi Π, every interpretation I of \Pi Π is a model of \Pi Π. And the weight of I is e to the power of S, where S is the sum of the weights of the grounded rules that are satisfied by I.
For any **LPMLN program** \Pi Π and any interpretation I of \Pi Π, it's possible that I is not a stable model of \Pi Π, in which case, the weight of I is 0. This is different from MLN since the weight of any interpretation I under MLN cannot be 0.
Below are the **steps to compute the weight of an interpretation I** of \Pi Π under LPMLN. It is very similar to that under MLN except for a stable model checking process in step 3.
First, we need to ground the given LPMLN program into the following grounded counter-part.
Second, we need to figure it out which of the above grounded rules are satisfied by I = \{smoke(alice), smoke(bob) \}
I={smoke(alice),smoke(bob)}. The satisfied rules are as follows.
Third, we check if I is a stable model of the above set of satisfied grounded rules (without the weights). If I is not a stable model, then the weight of I is 0. If I is a stable model, then we simply accumulate the weights of the above rules, and if the accumulated value is S, the weight of I is exp(S).
The most **probable stable model** is the stable model with the highest probability, which is also the stable model with the highest weight.
Thus, normally, we need to compute the weight of all 4 interpretations of this program: \varnothing, \{p\}, \{q\}, \{p,q\}. ∅,{p},{q},{p,q}.
One trick here is that, from intuition, we know q q can never be in any stable model since there is no support for q q (i.e., there is no rule whose head is q q). Thus we only need to compute the weights of

LPMLN probability = probability of I interpretation /Z(all probabilities
penalty based weight definition, we want to find rules that are not satisfied by interpretation {R(Jo), B(Jo)}. In program \PiΠ, only r5' is not satisfied by {R(Jo), B(Jo)}; e raise to -1 .
Lpmln: Soft stable model = probabilistic stable model, empty phi is always included

LPMLN, Optimal stable model: Identify rules/constraints that are violated then calculate wts as followis. We first check smallest weights at level 1, we found stable models {p}, {q}, {p, q} give 0 weight at level 1. Then we further check these three at level 0, we found only {q} gives the smallest weight, which is -5. As the result, {q} is the optimal stable model of this program.
$ lpmln-infer bird.lpmln -q bird -all
where bird.lpmln contains both hard rules and soft rules.
Calculate and display the probability of all the stable models.
Calculate and display the probability of all atoms with predicate "bird" to be true.
RDF: Every resource has an IRI (Internationalized Resource Identifier)
One advantage of using IRIs is to reduce the homonym problem of distributed data representation.
An object can be a resource, for example, authors, books, publishers, places.
Description logic is under open world assumption, ASP under closed world.
A concept C is **satisfiable** with respect to a knowledge base K if there exists a model I of K such that C^{I} is not equal to phi
C is subsumed by D with respect to a knowledge base K if every model I of K such that C^I is subsumed ⊑ D^I. e = 2.72

\varnothing ∅ and \{p\} {p}.	