贝叶斯网络中的每一个节点都可以用一个命题子句来表 示，因此贝叶斯网络本质上可以表示成命题子句集合，这使得 可以通过提升相应的命题子句到一阶逻辑子句来扩展贝叶斯 网络到一阶逻辑环境下

XXXX

PRMs 是一种非常流行的形式化表示方法，它用实体关

系模型作为表示框架。 PRMs 的基本思想是:有关实体类型

的信息是存在于关系中的。因此，每一个基础原子能存储多

个属性的信息，而且也是在属性水平上建立依赖关系。

PRMs 的优点是:表示方便，学习和测试元环性较为容

易，但是它不能表示函数(函数可以用于表示结构的不确定

性) ，也不能表示概念的否定(可以处理意外情况) ，这都限制

PRMs 的表达能力。 PRMs 现在已经成功地用于解决很多

问题，包括关系聚类、超文本分类等，并且在生物信息学领域

己经取得了较好的成果。

基于 Ngo Haddawy 的工作， Kersting De Raedt入了 BLPs 框架，它可以看作是 PLPs 的简化，而且确定子句逻辑(即纯 Prolog) 和贝叶斯网络都可以看作是 BLPs 的特例。 BLPs 也将原子看作是随机变量，更精确地，最小 Herbrand模型的原子构成了贝叶斯网络的随机变量。

在独立同分布假定下，由 BLPs 形成的 BNs 可以由任何

的贝叶斯网络推理机来计算覆盖关系，而且 BLPs 可以处理

连续的随机变量，并允许使用函数。

BLPs 的最新定义和 LBNs 很接近，主要的不同是 LBNs

用子句集合来指定贝叶斯网络中的随机变量并用另一个子句

集合来指定有向边，而 BLPs 则是使用同一个子句集合来指

定。这点使得 LBNs BLPs 更容易阅读(特别是对于已经

熟悉了 PRMs 的人而言)。

1)

> Kristian Kersting and Luc De Raedt. Bayesian logic programming: Theory and  
>  
> tool. Statistical Relational Learning, page 291, 2007.

Bayesian Logic Programming (2007) is one of several versions of "probabilistic Prolog", distinguished by an unusual degree of mathematical and conceptual elegance.   In essence, BLP allows one to label arbitrary relations in a Prolog-like program with probabilities, and then to propagate these probabilities through inferences.   An isomorphism between the probabilistic notion of a "direct causal influence" and the logical notion of an "immediate derivation" is maintained, so that one has a perfect alignment of the probabilistic and logical aspects, rather than an awkward hybrid.   However, BLP inherits the weaknesses as well as the strengths of Prolog are inherited, included a reliance on specific forms of logic expression normalization and the lack of a scalable inference method.

贝叶斯逻辑编程 (Bayesian Logic Programming，以下称BLP)采用贝叶斯网络来表示知识，其中节点表示命题子句，它是众多“概率Prolog”中的一种。BLP可以用于表示任意带概率的Prolog形式的关系，并能通过逻辑推理来传播这些概率。BLP中维持了概率层次上的“直接因果推理”和逻辑层次上“间接推理”的同构关系，使得该结构中的概率和逻辑推理能达到一致，而无需借助棘手的混合模型。然而，由于BLN依赖于Prolog，也继承了Prolog的不足之处，如依赖于特定的逻辑表达式归一化、缺少可扩展的推理机制等。

2)

Puech and Muggleton (2003) explore the relationship between possible-worlds semantics and directly-experience-driven frequency-counting-based semantics for grounding probabilistic logics. Specifically they compare Bayesian Logic Programs (BLPs) as a possible worlds semantics based formalism, and SLPs, Stochastic Logic Programs, as a formalism based on frequency defined semantics.  What they show is that if one considers appropriately extended versions of the BLP and SLP formalisms, the two approaches have the same expressive power.   In simple terms, what this means is that, if the mathematics is defined in a reasonable and appropriate way, one can interpret one's probabilistic logic expressions either in terms of possible worlds semantics or in terms of frequency based semantics, and it amounts to the same thing.

PUECH和Muggleton探索接地概率逻辑可能世界语义和直接体验驱动型频率计数语义之间的关系。特别是他们比较贝叶斯逻辑程序（BLPs）作为一种可能的基于世界语义学形式主义和学生学习概览，随机逻辑程序，基于对频率来定义语义的形式主义。他们表明的是，如果考虑适当延长BLP和SLP形式化的版本，这两种方法都有相同的表现力。简单来说，这意味着什么是，如果该数学是在合理和适当的方式定义，可以解释或在可能的世界语义学术语或在基于频率的语义方面人的概率逻辑表达式，以及它相当于同一 事情。

 The two different perspectives may have differential value for analyzing various specific phenomena, of course.   A related analysis in the context of PLN has been given by  
(Goertzel and Ikle',  
[http://www.academia.edu/498379/Grounding\_Possible\_Worlds\_Semantics\_in\_Experiential\_Semantics](http://www.academia.edu/498379/Grounding_Possible_Worlds_Semantics_in_Experiential_Semantics" \t "_blank)),  
where it is shown that PLN's experience-based semantics can be  
formalized as a variant of possible-worlds semantics.  
  
In general, it appears that when one cashes them out in terms of the  
actual observations and hypotheses made by a practical AI system  
regarding the everyday world, the difference between different  
philosophical interpretations of probability and logic often become  
mere differences of taste in formalization.   The really hard problems  
involved in using probabilistic logic for AI are not so much in the  
area of semantic interpretation of the uncertain logic expressions  
involved, but rather in the efficient estimation of the numerous  
probabilistic logic expressions needed to derive useful commonsense  
inferences, and the judicious choice of which inferences to make in  
order to achieve an AI system's practical goals.

\cite{Puech2003}

3)

> Luc De Raedt and Kristian Kersting. Probabilistic logic learning. ACM  
>  
> SIGKDD Explorations Newsletter, 5(1):31–48, 2003.  
  
There has been considerable work at the intersection of machine learning and probabilistic logic.   De Raedt and Kersting (2003) give  
a clear and concise review of the core conceptual issues here.   In Bayes net based formalisms such as BLPs and most other probabilistic  
logic programs, the natural application of machine learning is to learn from {\it interpretations} of probabilistic logic programs.   In  
grammar based formalisms such as SLPs (whose semantics is based on frequency counting corresponding to expressions in a logical grammar),  
the most natural application of machine learning is to learn logical entailments.   On the other hand, there are some formalisms such as RMMS and LOHMMs in which learning from traces is the most natural  
approach.    [NOTE: you may want to stick into your thesis references  
to RMM and LOHMM, given in the "probabilistic logic learning" paper]  
  
More recent work by Potapov (2015) points in a similar direction,  
showing that learning from traces is the most natural approach in the  
context of probabilistic programming.   In this latter case the  
learning is carried out by genetic programming performing optimization  
queries over spaces of probabilistic programs.  
  
[http://agi-conf.org/2015/wp-content/uploads/2015/07/agi15\_batischcheva.pdf](http://agi-conf.org/2015/wp-content/uploads/2015/07/agi15_batischcheva.pdf" \t "_blank)  
  (paper was in Proceedings of AGI-15)

4)

Probabilistic Similarity Logic (XX) focuses on the coherent  
integration of probabilistic logical reasoning regarding similarity,  
with domain specific similarity measures that may not be defined in  
logical terms (e.g. statistical similarity  measures between documents  
in an information retrieval context; or sequence alignment based  
similarity measures between gene or protein sequences in biology).  
They show how one can incorporate domain specific similarity  measures  
consistently into a probabilistic logic framework.  This qualifies as  
a simple kind of "hybrid" uncertain logical inference.   Our own use  
of probabilistic logic in our dialogue system has a similar aspect, in  
that we must integrate similarities (and other logical relationships  
such as inheritances) derived via probabilistic reasoning, with  
similarities obtained from other sources such as analysis of  
nonlinguistic data (e.g. machine vision based similarity in the case  
of a dialogue system controlling a robot or any other camera-connected  
agent).

5)

Markov Logic Networks (XX) is a novel formalism for inferring the  
probability distribution associated with a set of entities related by  
propositional logic relationships.   Some of the relationships must be  
given empirically derived probabilities, and then the network derives  
probabilities for the other relationships, via (in one common  
implementation) a Gibbs sampling based algorithm.   The effectiveness  
of the algorithm depends on the connectivity pattern of the graph and  
the complexity of the distribution involved.  For many problems in NLP  
it has proven effective, e.g. for entity extraction and semantic  
relationship extraction.   The formalism underlying the algorithm  
assumes a single probability distribution underlying the entire  
network, which is different philosophically and pragmatically from  
approaches like NARS and PLN that avoid comparable assumptions of  
global consistency.

6)  
  
Poon and Domingos (2009) use a recursive algorithm centered on Markov  
Logic Networks to statistically induce a "semantic grammar" from a  
corpus of documents.   Their approach is shown to outperform existing  
alternatives on a corpus of biomedical abstracts.   A key strength of  
their algorithm is its ability to identify semantically similar  
expressions even if they have significantly different syntactic form,  
a feat achieved via characterizing expressions "intensionally" in  
terms of their patterns of relationship with other expressions.   The  
recursion underlying their algorithm is relatively subtle and involves  
characterizing words and semantic relationships in terms of n'th order  
relationships, where an n'th order relationship is defined as a  
relationship among (n-1)'th order relationships, and a 0'th order  
relationship is a simple co-occurrence relationship.  Their MLN  
formalism allows them to keep track of the relationships on these  
various levels in a probabilistically consistent way.