

Model-Theoretic Expressivity Analysis

Manfred Jaeger

Institut for Datalogi, Aalborg University
Selma-Lagerlöfs Vej 300, 9220 Aalborg Ø, Denmark
jaeger@cs.aau.dk

1 Introduction

In the preceding chapter the problem of comparing languages was considered from a behavioral perspective. In this chapter we develop an alternative, model-theoretic approach.

In this approach we compare the expressiveness of probabilistic-logic (pl-) languages by considering the models that can be characterized in a language. Roughly speaking, one language L' is at least as expressive as another language L , if every model definable in L also is definable in L' . Results obtained in the model-theoretic approach can be somewhat stronger than results obtained in the behavioral approach in that equivalence of models entails equivalent behavior with respect to any possible type of inference tasks. On the other hand, the model-theoretic approach is somewhat less flexible than the behavioral approach, because only languages can be compared that define comparable types of models. A comparison between Bayesian Logic Programs (defining probability distributions on possible worlds) and Stochastic Logic Programs (defining probability distributions over derivations), therefore, is already quite challenging in a model-theoretic approach, as it requires first to define a unifying semantic framework. In this chapter, therefore, we focus on pl-languages that exhibit stronger semantic similarities (Bayesian Logic Programs (BLPs) [6], Probabilistic Relational Models (PRMs) [1], Multi-Entity Bayesian Networks [7], Markov Logic Networks (MLNs) [12], Relational Bayesian Networks (RBNs) [4]), and first establish a unifying semantics for these languages. However, the framework we propose is flexible enough (with a slightly bigger effort) to also accommodate languages like Stochastic Logic Programs [9] or Prism [13].

The focus of this chapter is expressivity analysis. Clearly, expressivity is only one relevant aspect in the comparison of languages. Further highly important issues are compactness of representation, efficiency of inference, and learnability in different languages. A meaningful comparison of these issues, however, requires concepts of equivalence of models and inferences, which is just what our expressivity analysis provides. Thus, this analysis is to be understood as a first step towards more comprehensive comparisons.

2 PL-Models

In this chapter the word *model* is used to refer to unique distributions over some state space. This is consistent with the usage in logic, where “model” refers to

a unique structure. It is different from the usage in statistics, where “model” refers to a parametric class of distributions. Specifically, when we talk about the model represented by some BLP, RBN or MLN, for example, we are referring to a fully quantified BLP, etc., i.e. all numeric parameters set to specific values.

As a first step towards a unifying semantics for different pl-languages, we have to find a common structure of the state spaces on which distributions are defined. A sufficiently general class of state spaces consists of the spaces that are generated by a set of random variables that can be written in the syntactic form of ground atoms, e.g. $blood_pressure(tom)$, $sister(susan,tom)$, $genotype(mother(tom))$, . . . These random variables take values in finite sets of states that are associated with the relation symbol, e.g. $states(genotype)=\{AA, Aa, aa\}$. At this point we do not consider continuous variables. We call any assignment of states to the set of all ground atoms constructible over a given vocabulary S of relation, function and constant symbols (the *Herbrand base of S* , $HB(S)$) a *Multi-valued Herbrand interpretation*.

To reason about identity we allow that $= \in S$. The symbol $=$ is seen as a binary Boolean relation symbol. Interpretations of $=$ are constrained to be consistent with identity relation on domain elements (i.e. they must satisfy the axioms of equality). Some languages (including RBNs and MLNs) use the $=$ relation to define models, but do not provide probabilistic models of $=$ itself. Some approaches have been proposed to model “identity uncertainty”, i.e. to build probabilistic models for $=$ [10, 8, 11].

The set of all multi-valued Herbrand interpretations for S is denoted $MVHI(S)$. We use ω, ω', \dots to denote individual multi-valued Herbrand interpretations. In the case where all relations in S are Boolean, then these ω are also referred to as *possible worlds* (in agreement with standard logic terminology). When $\omega \in MVHI(S)$, and $S' \subseteq S$ then $\omega[S']$ denotes restriction of ω to the symbols in S' . Similarly, when α is an arbitrary vector of ground S -atoms, then $\omega[\alpha]$ denotes the state assignment in ω to the ground atoms in α . Another notational convention we will use is to refer by r/k to a relation symbol of arity k . Specifically, $r/k \in S$ is to be read as “ r is a k -ary relation symbol in S ”. Similarly for function symbols.

We always assume that probability distributions P on $MVHI(S)$ are defined on the σ -algebra $\mathcal{A}(S)$ generated by all elementary events of the form $\alpha = s$, where $\alpha \in HB(S)$, and s is a value in the state space of α .

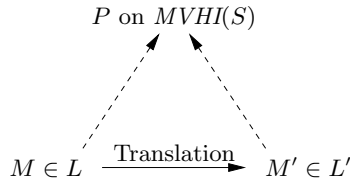


Fig. 1. Preliminary translation schema

Figure 1 gives a preliminary view of the model-theoretic language comparison: a language L' is at least as expressive as a language L , if for every model $M \in L$ defining a distribution P on $MVHI(S)$ there exists a model $M' \in L'$ defining the same distribution. The schema in Figure 1 is not yet fully adequate, however. The first problem with this schema is that we cannot expect the model M' to define a distribution on exactly the same state space $MVHI(S)$ as M . For example, language L' might only permit Boolean relations, whereas L operates with multi-valued relations. The translation from M to M' then will involve a “binarization” of the vocabulary S , leading to a new vocabulary S' , and hence a different probability space $MVHI(S')$. We must therefore allow that M' does not represent exactly the distribution P defined by M , but only that M' defines some P' that encodes all the information contained in P . In the following definition we formalize this scenario. The definition also provides for the case where the model M' does not encode all of M , but only as much as is needed to answer a restricted class of queries.

Definition 1. Let P, P' be probability distributions over $MVHI(S)$, respectively $MVHI(S')$. Let $\mathcal{Q} \subseteq \mathcal{A}(S)$. A \mathcal{Q} -embedding of P in P' is a mapping

$$h : \mathcal{Q} \rightarrow \mathcal{A}(S') \tag{1}$$

such that for all $Q \in \mathcal{Q}$:

$$P(Q) = P'(h(Q)).$$

We write $P \preceq_{\mathcal{Q}} P'$ if there exists a \mathcal{Q} -embedding of P in P' .

A conditional \mathcal{Q} -embedding of P in P' is a mapping (1) together with a subset $C \in \mathcal{A}(S)$, such that for all $Q \in \mathcal{Q}$:

$$P(Q) = P'(h(Q) \mid C).$$

We write $P \preceq_{\mathcal{Q},c} P'$ if there exists a conditional \mathcal{Q} -embedding of P in P' .

If $\mathcal{Q} = \mathcal{A}(S)$ we just write \preceq, \preceq_c instead of $\preceq_{\mathcal{Q}}, \preceq_{\mathcal{Q},c}$.

An important example for \mathcal{Q} is the set of all events of the form $\alpha = s$. If then $P \preceq_{\mathcal{Q}} P'$, we can retrieve from P' all single variable marginals of P , but not necessarily joint or conditional distributions of P .

We now turn to a second, more subtle and fundamental deficiency of the schema in Figure 1. Consider the situation where $MVHI(S)$ is finite (which happens when S contains only finitely many constant and no function symbols). In this case basically every pl-language will be able to represent any distribution P on $MVHI(S)$ (P could be expressed by a Bayesian network with one node for each $\alpha \in HB(S)$; for essentially all pl-languages it is known that they can encode any standard Bayesian network). Thus, it would immediately follow that for purely relational vocabularies all pl-languages are equally expressive, and that they have the same expressive power as standard Bayesian networks.

To see why this argument misses the point, consider a typical pl-model for genotypes in a pedigree. Such a model would be given by two distinguishable elements: on the one hand, there are general probabilistic rules that specify, for

example, that each of the two alleles of one gene is passed from parent to child with equal probability, or that specify the probability of a random mutation. On the other hand, there are basic facts that describe the structure of the pedigree, e.g. that John and Mary are the parents of Paul. The power and usefulness of pl-languages derives from this modularity that separates generic underlying probabilistic rules from domain-specific information.

The modularity in the model specification is most clearly expressed in PRMs, where the specification of the *skeleton structure* is distinguished from the actual probabilistic model, and in RBNs, where the specification of an *input structure* is distinguished from the specification of the actual RBN model. BLPs make a distinction between the *intensional* and the *extensional* model part, where the extensional part mostly is expressed in terms of special *logical relations*, roughly corresponding to the *predefined relations* of RBNs.

In the following we adopt the extensional/intensional terminology (originating in database theory), and by the following definition demand that a pl-model has a modular structure that separates the generic, high-level (intensional) part of the model from a specific, non-probabilistic (extensional) domain specification.

Definition 2. (*PL-model*) A PL-model M for a vocabulary S is a specification in a formal language L of a probability distribution $P[M]$ on $MVHI(S)$. The model M can be decomposed as $M = (M_{int}, M_{ext})$, such that

- (i) for a given M_{int} there exist infinitely many different $M_{ext}^{(1)}, M_{ext}^{(2)}, \dots$, such that $(M_{int}, M_{ext}^{(i)})$ defines a distribution on some $MVHI(S_i)$, where for $i \neq j$ the vocabularies S_i, S_j contain different constant symbols.
- (ii) if $\alpha \in HB(S)$ with $0 < P[M](\alpha) < 1$, then there exists $M' = (M'_{int}, M_{ext})$ with $P[M'](\alpha) \neq P[M](\alpha)$.

Definition 2 requires that a model M has a modular structure (M_{int}, M_{ext}) . Moreover, conditions (i) and (ii) make certain minimal requirements for the components: condition (i) ensures that the generic, intensional part of the model gives rise to infinitely many concrete model instances obtained by exchanging the extensional part, and that these changes permit a change of the underlying domain as represented by the constants in S . Condition (i) alone would permit the trivial decomposition $M = (\emptyset, M_{ext})$. Condition (ii), therefore, requires that M_{int} actually contains the essential probabilistic information, and that by changes to M_{int} (typically just by change of numerical parameter values) one can change the quantitative aspects of the model.

It must be emphasized that the partitioning of a model into intensional and extensional part may not be unique. For some languages there exists a canonical decomposition that is also reflected in a syntactic distinction between the two parts. For other languages, several meaningful partitions satisfying Definition 2 can exist.

We now arrive at the refined schema in Figure 2: to show that L' is at least as expressive as L , we have to find translations between L and L' models that respect the modular structure of the models, i.e. we need separate translations

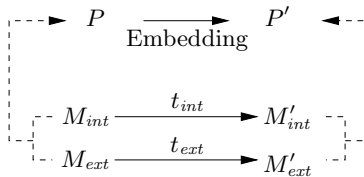


Fig. 2. Translations and Embeddings

for the intensional and extensional parts. For the following precise definition we take it for granted that for L and L' decompositions into intensional and extensional parts have been defined, and emphasize that modifications to how intensional and extensional parts are identified can lead to changes in the partial expressivity order \preceq here defined.

Definition 3. Language L' is at least as expressive as L with respect to queries Q , $L \preceq_Q L'$, if $\exists t_{int} \forall M_{int} \exists t_{ext} \forall M_{ext}$

$$P[M_{int}, M_{ext}] \preceq_Q P[t_{int}(M_{int}), t_{ext}(M_{ext})] \quad (2)$$

If (2) is only satisfied by a conditional embedding $\preceq_{Q,c}$, we write $L \preceq_{Q,c} L'$.

The quantifier string $\exists t_{int} \forall M_{int} \exists t_{ext} \forall M_{ext}$ in Definition 3 requires some explanation. According to the definition, the exact translation used for the extensional part may depend on the concrete intensional part. This reflects to some extent the “primacy” of the intensional model part, which is supposed to contain the essential probabilistic specifications, whereas the extensional part contains ancillary domain information. The following example illustrates how the possible dependence of t_{ext} on M_{int} can become relevant in practice.

Example 1. One special application of comparisons of the form $L \preceq L'$ is the case where L' is a fragment of L . In such a case, a relation $L \preceq L'$ is basically a normal form theorem: every model M is equivalent to a model M' in a normal form characterized by the syntactic restrictions of L' . As an example, let L be the language of BLPs, and L' the BLP fragment in which Bayesian clauses are not allowed to contain constant symbols.

Consider the following BLP (here not showing the probability annotation of the intensional clauses):

M_{ext} :
`father(peter, paul)`
`mother(mary, paul)`

M_{int} :
`bloodtype(X) | father(thomas, X)`
`bloodtype(X) | father(Y, X), mother(Z, X), bloodtype(Y), bloodtype(Z)`

Here an intensional probabilistic rule contains the constant ‘thomas’. In order to eliminate the occurrence of this constant, we can introduce a new unary relation symbol *thomas_rel*/1, and translate the original model into

M'_{ext} :
`father(peter,paul)`
`mother(mary,paul)`
`thomas_rel(thomas)`

M'_{int} :
`bloodtype(X) | father(Y,X),thomas_rel(Y)`
`bloodtype(X) | father(Y,X),mother(Z,X),bloodtype(Y),bloodtype(Z)`

In general, t_{int} replaces constants `cons` in M_{int} with new variables, and adds `cons_rel()` atoms to the clauses. This translation is independent of M_{ext} . The translation t_{ext} adds clauses `cons_rel(cons)` to M_{ext} . This depends on M_{int} , because we first have to inspect M_{int} in order to find the constant symbols in need of elimination.

3 Case Study: MLNs and RBNs

In this section we apply the general framework established in the previous section to compare the expressiveness of Markov Logic Networks [12] with Relational Bayesian Networks [4]. We begin by briefly reviewing the essential concepts and definitions for both languages.

3.1 Markov Logic Networks

In the following we give a definition of MLNs following [12]. Notation and presentation are somewhat adapted to our general conceptual setting. In particular, we make explicit an intensional/extensional division in MLN models, which is only implicit in the original definitions. Our definitions are based on the *known functions assumption* stated in [12], which basically stipulates that all function symbols in the language have a fixed and known interpretation, and are not modeled probabilistically. The general MLN paradigm allows to relax or eliminate this assumption. The translation we present in this chapter can be generalized also to MLN versions without the known functions assumption.

Under the known function assumption, MLNs contain a domain specification given by a set of constant symbols S_C^{mln} , and interpretations over this domain of a set of function symbols S_F^{mln} . This domain specification is the extensional part of the model, i.e.

$$M_{ext}^{mln} \in MVHI(S_{ext}^{mln}),$$

where $S_{ext}^{mln} := S_C^{mln} \cup S_F^{mln} \cup \{=\}$.

The intensional part of an MLN is given by a set of pairs

$$M_{int}^{mln} = \{(\phi_i(x_1, \dots, x_{k_i}), w_i) \mid i = 1, \dots, n\}, \quad (3)$$

Here the ϕ_i are first-order logic formulas in $S^{mln} := S_{ext}^{mln} \cup S_R^{mln}$, where S_R^{mln} is a set of Boolean relation symbols. The w_i are numbers in $\mathbb{R} \cup \{\infty\}$. $M^{mln} =$

$(M_{ext}^{mln}, M_{int}^{mln})$ defines a distribution on $MVHI(S^{mln})$ as follows:

$$P[M^{mln}](\omega) = \begin{cases} 0 & \text{if } \omega[S_{ext}^{mln}] \neq M_{ext}^{mln}, \\ & \text{or } \omega \not\models \forall \mathbf{x} \phi_i(\mathbf{x}) \text{ for some } i \text{ with } w_i = \infty \\ \frac{1}{Z} \exp\left(\sum_{\substack{i=1 \\ w_i \neq \infty}}^n n_i(\omega) w_i\right) & \text{otherwise} \end{cases} \quad (4)$$

where $n_i(\omega)$ is the number of true instances of ϕ_i in ω obtained by grounding ϕ_i with constants from S_C^{mln} . Z is a normalizing constant.

Table 1. MLN: friends and smokers example

ϕ_i	w_i
$Fr(x, y) \wedge Fr(y, z) \Rightarrow Fr(x, z)$	0.7
$\neg \exists y Fr(x, y) \Rightarrow Sm(x)$	2.3
$Sm(x) \Rightarrow Ca(x)$	1.5
$Fr(x, y) \Rightarrow (Sm(x) \Leftrightarrow Sm(y))$	1.1
$Fr(Anna, Bob)$	∞

Example 2. (adapted from [12]) Table 1 shows a small intensional model using relation symbols $S_R^{mln} = \{Fr(iend), Sm(okes), Ca(ncer)\}$. The model consists of four weighted formulas expressing, respectively, that the friends relation is transitive, friendless people smoke, smoking causes cancer, and friends will either both smoke or both not smoke. Furthermore, there is a hard constraint saying that *Anna* is friends with *Bob* (not necessarily implying the converse). This intensional model is to be combined with domain specifications given by a set of constants, including the constants *Anna, Bob*, e.g. $S_C^{mln} = \{Anna, Bob, Paul\}$. There are no function symbols, so this set of constants (together with the unique names assumption) defines $M_{ext}^{mln} \in MVHI(S_C^{mln} \cup \{=\})$.

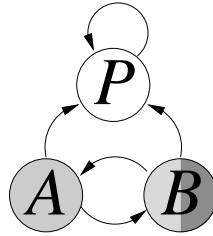


Fig. 3. A small $\omega \in MVHI(S_C \cup S_R \cup \{=\})$

Let $\omega \in MVHI(S_{ext}^{mln})$ as shown in Figure 3, where arrows indicate the interpretation of the *Fr* relation, light shading indicates the objects for which *Sm* is

true, and dark shading indicates the objects for which Ca is true. In ω there are 26 groundings that satisfy $\phi_1(x, y, z)$ (there are 3^3 possible groundings, and only the grounding $x = A, y = B, z = A$ does not satisfy ϕ_1), i.e. $n_1(\omega) = 26$. Similarly, $n_2(\omega) = 3$ (the condition $\neg\exists y Fr(x, y)$ is not satisfied for any x), $n_3(\omega) = 2$, and $n_4(\omega) = 7$ (this example highlights some potential difficulties with calibrating the weights for material implications, which can have a large number of true groundings simply because most groundings do not satisfy the antecedent).

3.2 Relational Bayesian Networks

We here give a condensed summary of all relevant technical definitions for syntax and semantics of RBNs. For more detailed explanations and motivating examples the reader is referred to [5].

In RBN models the vocabulary is partitioned into predefined (extensional) and probabilistic (intensional) symbols: $S^{rbn} = S_{ext}^{rbn} \cup S_{int}^{rbn}$, where S_{ext}^{rbn} consists of relation and constant symbols (including the $=$ relation), and S_{int}^{rbn} of relation symbols only. The extensional part of a RBN model consists of $M_{ext}^{rbn} \in MVHI(S_{ext}^{rbn})$, where the interpretation of $=$ follows the unique names assumption (in the original RBN terminology, M_{ext}^{rbn} is called an *input structure*).

The intensional part (i.e. the RBN proper) consists of a collection of *probability formulas* F_r for the intensional relations:

$$M_{int}^{rbn} = \{F_r(x_1, \dots, x_k) \mid r/k \in S_{int}^{rbn}\}. \quad (5)$$

Probability formulas are formal expressions generated by a syntax which can be seen as a probabilistic counterpart of the syntax of predicate logic: the probability formula constructs of *atoms*, *convex combinations* and *combination functions* (cf. Table 2) correspond to predicate logic constructs of *atomic formulas*, *Boolean connectives*, and *quantification*, respectively. A first-order formula $\phi(x_1, \dots, x_k)$ evaluates for particular domain elements c_1, \dots, c_k from some possible world ω to a truth value $\phi(c_1, \dots, c_k)[\omega] \in \{true, false\}$ (note that $\phi(c_1, \dots, c_k)[\omega] = true$ is synonymous with $\omega \models \phi(c_1, \dots, c_k)$).

A probability formula $F(x_1, \dots, x_k)$ evaluates to a probability value

$$F(c_1, \dots, c_k)[\omega] \in [0, 1].$$

Both the first-order and the probability formula depend for their evaluation usually not on the whole possible world ω , but only on the truth values of a set of ground atoms $\alpha(\phi, \mathbf{c})$, respectively $\alpha(F, \mathbf{c})$. For example, the evaluation of the first-order formula $\phi(x) = \exists y(r(x, y) \wedge t(y))$ depends for $x = c$ on the atoms $r(c, c'), t(c')$ for all c' in the domain of ω .

In the case of probability formulas we will be mostly interested in the dependence on S_{int}^{rbn} -atoms. The set of S_{int}^{rbn} -atoms that the evaluation of $F(\mathbf{c})[\omega]$ depends on is determined by \mathbf{c} and the extensional part of ω , i.e. $\omega[S_{ext}^{rbn}]$. We write $\alpha(F, \mathbf{c}, \omega[S_{ext}^{rbn}])$ for this set of ground S_{int}^{rbn} -atoms.

Table 2 now summarizes syntax and semantics of probability formulas. Shown are the syntactic form of the formulas $F(\mathbf{x})$ constructed via the four different

construction rules, a specification of the sets $\alpha(F, \mathbf{c}, \omega[S_{ext}^{rbn}])$, and the computation rule for the probability value $F(\mathbf{c})[\omega]$. The combination function construct here is only shown for the *noisy-or* combination function, which is the only one we will need for MLN encodings.

Table 2. RBN syntax and semantics – F_1, F_2, F_3, F' are any probability formulas; $\psi(\mathbf{x}, \mathbf{y})$ is any Boolean combination of S_{ext}^{rbn} -atoms

	$F(\mathbf{x})$	$\alpha(F, \mathbf{c}, \omega[S_{ext}^{rbn}])$
Constant	p ($p \in [0, 1]$)	\emptyset
Atom	$r(\mathbf{x})$ ($r \in S_{int}^{rbn}$)	$r(\mathbf{c})$
Convex Combination	$(F_1 : F_2, F_3)$	$\cup_{i=1}^3 \alpha(F_i, \mathbf{c}, \omega[S_{ext}^{rbn}])$
Combination Function	$noisy-or\{F'(\mathbf{x}, \mathbf{y}) \mid \mathbf{y} : \psi(\mathbf{x}, \mathbf{y})\}$	$\bigcup_{\substack{\mathbf{c}' \\ \omega[S_{ext}^{rbn}] \models \psi(\mathbf{c}, \mathbf{c}')}} \alpha(F, (\mathbf{c}, \mathbf{c}'), \omega[S_{ext}^{rbn}])$

	$F(\mathbf{c})[\omega]$
Constant	p
Atom	$\begin{cases} 1 & \text{if } r(\mathbf{c})[\omega] = true \\ 0 & \text{otherwise} \end{cases}$
Convex Combination	$F_1(\mathbf{c})[\omega]F_2(\mathbf{c})[\omega] + (1 - F_1(\mathbf{c})[\omega])F_3(\mathbf{c})[\omega]$
Combination Function	$1 - \prod_{\substack{\mathbf{c}' \\ \omega \models \psi(\mathbf{c}, \mathbf{c}')}} (1 - F(\mathbf{c}, \mathbf{c}')[\omega])$

A pair $M^{rbn} = (M_{ext}^{rbn}, M_{int}^{rbn})$ induces a dependency relation between S_{int}^{rbn} -atoms:

$$r(\mathbf{c}) \preceq r'(\mathbf{c}') \Leftrightarrow r'(\mathbf{c}') \in \alpha(F_r, \mathbf{c}, M_{ext}^{rbn}).$$

If this dependency relation is acyclic, then we obtain a well-defined probability distribution on $MVHI(S)$ via

$$P[M^{rbn}](\omega) = \begin{cases} 0 & \text{if } \omega[S_{ext}^{rbn}] \neq M_{ext}^{rbn} \\ \prod_{\substack{r \in S_{int}^{rbn} \\ \omega \models r(\mathbf{c})}} \prod_{\substack{\mathbf{c} \\ \omega \models r(\mathbf{c})}} F_r(\mathbf{c})[\omega] \prod_{\substack{\mathbf{c} \\ \omega \not\models r(\mathbf{c})}} (1 - F_r(\mathbf{c})[\omega]) & \text{otherwise} \end{cases}$$

Probability formulas can encode standard first-order formulas in the following sense: for all first-order $\phi(\mathbf{x})$ there exists a probability formula $F_\phi(\mathbf{x})$, such that for all ω, \mathbf{c} : $\phi(\mathbf{c})[\omega] = true$ iff $F_\phi(\mathbf{c})[\omega] = 1$, and $\phi(\mathbf{c})[\omega] = false$ iff $F_\phi(\mathbf{c})[\omega] = 0$ [4]. This encoding will be the cornerstone for our MLN to RBN translation in the next section.

3.3 MLN to RBN translation

Let $M^{mln} = (M_{ext}^{mln}, M_{int}^{mln})$ be a MLN model with M_{int}^{mln} as in (3). We begin by defining the vocabulary for the target RBN model:

$$\begin{aligned} S_{ext}^{rbn} &= S_C^{mln} \cup \{r_f/(k+1) \mid f/k \in S_F^{mln}\} \\ S_{int}^{rbn} &= S_R^{mln} \cup \{r_{\phi_i}/k_i \mid i = 1, \dots, n\}. \end{aligned}$$

Thus, S_{ext}^{rbn} is $S_C^{mln} \cup S_F^{mln}$ with relations instead of functions, and S_{int}^{rbn} adds to S_R^{mln} new relation symbols r_{ϕ_i} corresponding to the formulas in M_{int}^{mln} .

The translation t_{ext} is independent of the intensional model part M_{int}^{mln} , and simply consists of a transformation of $M_{ext}^{mln} \in MVHI(S_{ext}^{mln})$ into $M_{ext}^{rbn} \in MVHI(S_{ext}^{rbn})$ by replacing interpretations of $f/k \in S_F^{mln}$ with corresponding interpretations of $r_f/(k+1) \in S_{ext}^{rbn}$.

To define $M_{int}^{rbn} := t_{int}(M_{int}^{mln})$, we have to define for each relation $r \in S_{int}^{rbn}$ a probability formula. For $r/k \in S_R^{mln}$ we simply define:

$$F_r(x_1, \dots, x_k) = 0.5. \quad (6)$$

The formulas F_r ($r \in S_R^{mln}$) together with the input structure M_{ext}^{rbn} define a uniform distribution over $\{\omega \in MVHI(S_{ext}^{rbn}, S_R^{mln}) \mid \omega[S_{ext}^{rbn}] = M_{ext}^{rbn}\}$.

The core of the translation lies in the definition of probability formulas F_{ϕ_i} for the new relation symbols r_{ϕ_i} . The main component in the construction of the F_{ϕ_i} are sub-formulas H_{ϕ_i} that are essentially encodings of the formulas ϕ_i , as mentioned at the end of the preceding section. More precisely, we construct probability formulas H_{ϕ_i} with the following properties:

- (i) $\alpha(H_{\phi_i}, \mathbf{c}, \omega[S_{ext}^{rbn}])$ only contains atoms in relations from S_R^{mln} .
- (ii) $H_{\phi_i}(\mathbf{c})[\omega] \in \{0, 1\}$ for all ω, \mathbf{c} .
- (iii) for all $\omega \in MVHI(S_{ext}^{mln}), \mathbf{c}$, and $\omega' \in MVHI(S_{ext}^{rbn}, S_R^{mln})$ with $\omega'[S_{ext}^{rbn}] = t_{ext}(\omega[S_{ext}^{mln}])$ and $\omega'[S_R^{mln}] = \omega[S_R^{mln}]$:

$$\phi_i(\mathbf{c})[\omega] = true \Leftrightarrow H_{\phi_i}(\mathbf{c})[\omega'] = 1.$$

The formulas H_{ϕ} are defined inductively in the manner described in [4]. Some additional provisions are necessary for dealing with the transformation of function symbols into a relational representation.

Case 1a: ϕ is a relational atom. This is the most difficult case, as it involves the elimination of function. We demonstrate the construction of H_{ϕ} by a generic example: let $\phi = r(f(d), x)$, with $r \in S_R^{mln}$, $f \in S_F^{mln}$, $d \in S_C^{mln}$. Define

$$H_{\phi}(x) := noisy-or\{r(y, x) \mid y : r_f(d, y)\}.$$

According to the semantics of probability formulas, the evaluation $H_{\phi}(c)[\omega']$ performs all substitutions $r(y, c)[y/c']$ for $c' \in S_C^{mln}$ that satisfy $r_f(d, c')$, by evaluating the resulting ground probability formulas, and by combining all values so obtained by noisy-or. We first observe that this evaluation does not require truth values of any atoms in the relations r_{ϕ_i} , so that (i) is satisfied. By the

definition of M_{ext}^{rbn} , the condition $r_f(d, y)$ is satisfied exactly for $y = c'$ with $c' = f(d)$ in M_{ext}^{mln} . Thus, only this substitution is performed, and the evaluation of $H_\phi(c)[\omega']$ reduces to the evaluation of $noisy-or\{r(c', c)\}$. The evaluation of $r(c', c)$ returns 1, respectively 0, according to whether $r(c', c)$ is true, respectively false, in ω' , or, equivalently $r(f(d), c)$ is true, respectively false, in ω . Since, finally $noisy-or\{0\} = 0$ and $noisy-or\{1\} = 1$, we obtain (ii) and (iii).

Case 1a: ϕ is an equational atom. This case is similar. The formula for the equational atom $f(c) = x$ is given by $noisy-or\{1 \mid y : r_f(c, y) \wedge y = x\}$. This construction utilizes the convention that $noisy-or\emptyset := 0$.

Case 2a (Negation): $\phi(x) = \neg\psi(x)$. Define $H_\phi(x) := (H_\psi(x) : 0, 1)$, using the convex combination construct for probability formulas.

Case 2a (Conjunction): $\phi(x) = \psi(x) \wedge \chi(x)$: Define $H_\phi(x) := (H_\psi(x) : H_\chi(x), 0)$, again using convex combinations.

Case 3 (Existential quantifiers): $\phi(x) = \exists y\psi(x, y)$. Define $H_\phi(x) := noisy-or\{H_\psi(x, y) \mid y : \tau\}$, where τ stands for a tautological constraint (e.g. $y = y$). Thus, the sub-formula $H_\psi(x, c)$ will be evaluated for all $c \in C$, and $H_\phi(x)$ returns 1 iff $H_\psi(x, c)$ evaluates to 1 for at least one $c \in C$.

In all cases condition (i) is immediate from the syntactic form of the constructed probability formulas (they do not contain any ϕ_i -atoms), and (ii),(iii) follow from the evaluation rules for probability formulas.

Given the formulas H_{ϕ_i} , we define the final probability formulas F_{ϕ_i} as follows:

$$F_{\phi_i}(x) := \begin{cases} (H_{\phi_i} : 1, 0) & \text{if } w_i = \infty \\ (H_{\phi_i} : 1, 1/e^{w_i}) & \text{if } \infty > w_i \geq 0 \\ (H_{\phi_i} : e^{w_i}, 1) & \text{if } w_i < 0 \end{cases}$$

Example 3. Table 3 shows the formulas H_{ϕ_i} and F_{ϕ_i} for ϕ_1, \dots, ϕ_5 from Table 1. Here we have translated implications $\phi \Rightarrow \psi$ directly into probability formulas $(H_\phi : H_\psi, 1)$, rather than applying the translation given above for \neg and \wedge to $\neg(\phi \wedge \neg\psi)$. Note, too, that we need to encode $Fr(Anna, Bob)$ in the roundabout way shown in the table, because the RBN syntax does not allow constants from S_{ext}^{rbn} as arguments in atomic relation formulas (cf. Table 2).

Table 3. Translation of M_{int}^{mln} of Table 1

H_{ϕ_i}	F_{ϕ_i}
$((Fr(x, y) : Fr(y, z), 0) : Fr(x, z), 0)$	$(H_{\phi_1} : 1, 0.496)$
$(noisy-or\{(Fr(x, y) : Sm(x), 1) \mid y : y = y\} : 0, 1)$	$(H_{\phi_2} : 1, 0.1)$
$(Sm(x) : Ca(x), 1)$	$(H_{\phi_3} : 1, 0.223)$
$(Fr(x, y) : (Sm(x) : Sm(y), (Sm(y) : 0, 1)))$	$(H_{\phi_4} : 1, 0.332)$
$noisy-or\{Fr(x, y) \mid x, y : x = Anna \wedge y = Bob\}$	$(H_{\phi_5} : 1, 0)$

Having defined the translations t_{ext}, t_{int} , we have to show that

$$P[M^{mln}] \preceq_c P[M^{rbn}]. \quad (7)$$

where $M^{mln} = (M_{int}^{mln}, M_{ext}^{mln})$ and $M^{rbn} = (t_{int}(M_{int}^{mln}), t_{ext}(M_{ext}^{mln}))$. For this we have to find a suitable embedding, and a conditioning set C .

Since both $MVHI(S^{mln})$ and $MVHI(S^{rbn})$ are finite, we need to define the embedding $h(Q)$ only for singleton $Q = \{\omega\}$ ($\omega \in MVHI(S^{mln})$). First define $\tilde{h}(\omega) \in MVHI(S_{ext}^{rbn}, S_R^{mln})$ as the unique $\tilde{\omega}$ with $\tilde{\omega}[S_{ext}^{rbn}] = t_{ext}(\omega[S_{ext}^{mln}])$, and $\tilde{\omega}[S_R^{mln}] = \omega[S_R^{mln}]$. Now let

$$h(\omega) := \{\omega' \in MVHI(S^{rbn}) \mid \omega'[S_{ext}^{rbn}, S_R^{mln}] = \tilde{h}(\omega)\}. \quad (8)$$

Thus, $h(\omega)$ contains all possible extensions of $\tilde{h}(\omega)$ with interpretations of the relations $r_{\phi_i}/k_i \in S_{int}^{rbn} \setminus S_R^{mln}$.

Now let

$$C := \{\omega' \in MVHI(S^{rbn}) \mid \forall i = 1, \dots, n \forall \mathbf{c} : r_{\phi_i}(\mathbf{c})[\omega'] = true\}. \quad (9)$$

To show (7) it is sufficient to show that for all $\omega \in MVHI(S^{mln})$:

$$P[M^{mln}](\omega) = 0 \Leftrightarrow P[M^{rbn}](h(\omega) \mid C) = 0, \quad (10)$$

and for all $\omega_1, \omega_2 \in MVHI(S^{mln})$ with $P[M^{mln}](\omega_i) > 0$:

$$\log \frac{P[M^{mln}](\omega_1)}{P[M^{mln}](\omega_2)} = \log \frac{P[M^{rbn}](h(\omega_1) \mid C)}{P[M^{rbn}](h(\omega_2) \mid C)}. \quad (11)$$

It is quite straightforward to verify (10) from the definitions of $P[M^{mln}]$ and $P[M^{rbn}]$. We therefore only show (11), for which we then can make the simplifying assumption that $w_i < \infty$ for all i . By the semantics of MLNs, we obtain for the left-hand side of (11):

$$\log \frac{P[M^{mln}](\omega_1)}{P[M^{mln}](\omega_2)} = \sum_{i=1}^n w_i (n_i(\omega_1) - n_i(\omega_2)), \quad (12)$$

For the right-hand side, we first obtain

$$\begin{aligned} \log \frac{P[M^{rbn}](h(\omega_1) \mid C)}{P[M^{rbn}](h(\omega_2) \mid C)} &= \log P[M^{rbn}](h(\omega_1) \cap C) - \log P[M^{rbn}](h(\omega_2) \cap C) \\ &= \log P[M^{rbn}](h(\omega_1)) P[M^{rbn}](C \mid h(\omega_1)) - \log P[M^{rbn}](h(\omega_2)) P[M^{rbn}](C \mid h(\omega_2)) \\ &= \log P[M^{rbn}](C \mid h(\omega_1)) - \log P[M^{rbn}](C \mid h(\omega_2)). \end{aligned} \quad (13)$$

The last equality follows from $P[M^{rbn}](h(\omega_1)) = P[M^{rbn}](h(\omega_2))$, which holds because $P[M^{rbn}](h(\omega_i))$ is equal to the marginal probability $P[M^{rbn}](\tilde{h}(\omega_i))$ defined by M_{ext}^{rbn} and the probability formulas for $r \in S_R^{mln}$ alone. According to (6), these probabilities are uniform over the ω that have nonzero probability.

We now determine

$$P[M^{rbn}](C \mid h(\omega_i)) = \bigcap_{i=1}^n \bigcap_{\mathbf{c}} P[M^{rbn}](r_{\phi_i}(\mathbf{c}) = true \mid h(\omega_i)).$$

Since $F_\phi(\mathbf{c})$ only depends on relations in S_R^{mln} , we have that the random variables $r_{\phi_i}(\mathbf{c})$ are conditionally independent given an interpretation of all relations in S_R^{mln} . Furthermore, since all $\omega' \in h(\omega)$ have the same interpretation of S_R^{mln} , we obtain

$$P[M^{rbn}](r_{\phi_i}(\mathbf{c}) = true \mid h(\omega_i)) = F_{\phi_i}(\mathbf{c})[\tilde{h}(\omega)].$$

This gives us

$$\begin{aligned} \log P[M^{rbn}](C \mid h(\omega)) &= \sum_{i=1}^n \sum_{\mathbf{c}} \log F_{\phi_i}(\mathbf{c})[\tilde{h}(\omega)] \\ &= \sum_{i=1}^n \left(\sum_{\substack{\mathbf{c} \\ w_i \geq 0 \\ \tilde{h}(\omega) \models \phi_i(\mathbf{c})}} \log(1) + \sum_{\substack{\mathbf{c} \\ \tilde{h}(\omega) \not\models \phi_i(\mathbf{c})}} \log(1/e^{w_i}) \right) \\ &\quad + \sum_{i=1}^n \left(\sum_{\substack{\mathbf{c} \\ w_i < 0 \\ \tilde{h}(\omega) \models \phi_i(\mathbf{c})}} \log(e^{w_i}) + \sum_{\substack{\mathbf{c} \\ \tilde{h}(\omega) \not\models \phi_i(\mathbf{c})}} \log(1) \right) \\ &= \sum_{i:w_i \geq 0} -w_i(N_i - n_i(\omega)) + \sum_{i:w_i < 0} w_i n_i(\omega), \quad (14) \end{aligned}$$

where N_i is the total number of possible groundings \mathbf{c} of $\phi_i(\mathbf{x})$, and, thus, $N_i - n_i(\omega)$ is the number of groundings with $\tilde{h}(\omega) \not\models \phi_i(\mathbf{c})$. The terms $-w_i N_i$ cancel when taking the difference in (13), so that we finally obtain for the right-hand side of (11) the same expression as in (12) for the left-hand side.

We have now shown that RBNs are at least as expressive as MLNs. It is an open question whether the converse also holds.

Beyond the pure expressivity result, our MLN to RBN translation provides some additional insights: first, it is clear that the size of the RBN encoding of a MLN model is linear in the size of the MLN, so that compactness of representation is preserved. Second, one can see that MLN models and their RBN encodings will exhibit very similar behavior in terms of inference complexity: inference for MLNs is conducted on a *ground Markov network* [12] whose nodes are ground atoms in the relations from S_R^{mln} with constants from S_C^{mln} . Inference for RBNs (usually) is conducted on a *ground Bayesian network*, whose nodes are ground atoms in the relations from S_{int}^{rbn} with constants from S_C^{mln} . For inference, this Bayesian network will first be transformed into its *moral graph*. This moral graph turns out to have essentially the same structure as the ground Markov network from the MLN, only that to the cliques of S_R^{mln} -nodes are attached nodes with ground r_ϕ -atoms. Since for inference these nodes are all instantiated to true, they can easily be eliminated, and one ends up with a graphical support structure for inference in the RBN model that is identical to the ground Markov network. Thus, the commonly used inference techniques (exact or approximate) that operate on ground graphical models will show very similar behavior for MLN and RBN encodings. This does not preclude the possibility, however, that

for one language one might find a more sophisticated inference technique, which does not readily translate into a corresponding inference technique for the other language.

4 Conclusion

In this chapter we have developed a model-theoretic framework for comparisons of probabilistic logic languages. The framework is based on the key hypothesis that the essential feature of pl-languages is their modularity: they allow to represent general, high-level probabilistic specifications (the intensional model part), that is combined with the specification of concrete domains (the extensional model part).

Within this framework we have shown that the RBN language can encode MLN models. This result is based on basic versions of RBNs and MLNs. Both languages can be extended in various ways, e.g. to provide probabilistic models of functions in addition to probabilistic relations, or to provide probabilistic models for infinite domains [2, 14]. For some of the simpler extensions the basic translation method described in this chapter will also be applicable. For more complex extensions (notably infinite domains), however, some substantial additional effort may be required to determine whether MLN models can be translated into RBN models, or vice-versa.

Turning our attention to other languages, we conjecture that BLPs and RBNs are equally expressive when both languages are restricted to the noisy-or combination function. Since only noisy-or is required for the RBN encodings of MLNs, this would also mean that $MLN \preceq_c BLP$.

Acknowledgments

The author wants to thank Kristian Kersting and Luc De Raedt for many fruitful discussions on the topic of this paper. A preliminary account of some of the material in this chapter was given in [3]. This work was supported in part by the EU IST program: FP6-508861, Application of Probabilistic ILP II (April-II).

References

1. N. Friedman, Lise Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, 1999.
2. M. Jaeger. Reasoning about infinite random structures with relational bayesian networks. In Anthony G. Cohn, Lenhart Schubert, and Stuart C. Shapiro, editors, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR-98)*, pages 570–581, Trento, Italy, 1998. Morgan Kaufmann.
3. M. Jaeger, K. Kersting, and L. De Raedt. Expressivity analysis for pl-languages (position paper). In *Online Proceedings of the Workshop on Statistical Relational Learning (SRL-06)*, 2006.

4. M. Jaeger. Relational bayesian networks. In Dan Geiger and Prakash Pundalik Shenoy, editors, *Proceedings of the 13th Conference of Uncertainty in Artificial Intelligence (UAI-13)*, pages 266–273, Providence, USA, 1997. Morgan Kaufmann.
5. M. Jaeger. Complex probabilistic modeling with recursive relational Bayesian networks. *Annals of Mathematics and Artificial Intelligence*, 32:179–220, 2001.
6. K. Kersting and L. De Raedt. Towards combining inductive logic programming with bayesian networks. In *Proceedings of the 11th International Conference on Inductive Logic Programming (ILP-01)*, volume 2157 of *LNAI*, pages 118–131, 2001.
7. K. B. Laskey and P. C. G. da Costa. Of starships and klingons: Bayesian logic for the 23rd century. In *Proceedings of UAI-05*, 2005.
8. B. Milch, B. Marthi, S. Russell, D. Sontag, D.L. Ong, and A. Kolobov. Blog: Probabilistic logic with unknown objects. In *Proc. 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1352–1359, 2005.
9. S. Muggleton. Stochastic logic programs. In L. de Raedt, editor, *Advances in Inductive Logic Programming*, pages 254–264. IOS Press, 1996.
10. H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. In *Proceedings of NIPS-03*, 2003.
11. D. Poole. Logical generative models for probabilistic reasoning about existence, roles and identity. In *Proceedings of AAAI-07*, 2007.
12. M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107 – 136, 2006.
13. T. Sato. A statistical learning method for logic programs with distribution semantics. In *Proceedings of the 12th International Conference on Logic Programming (ICLP'95)*, pages 715–729, 1995.
14. P. Singla and P. Domingos. Markov logic in infinite domains. In *Proceedings of UAI-07*, 2007.