From Probabilistic Horn Logic to Chain Logic

Nivea Ferreira Arjen Hommersom Peter Lucas

Institute for Computing and Information Sciences,
Radboud University Nijmegen, Nijmegen, The Netherlands
Email: {nivea,arjenh,peterl}@cs.ru.nl

Abstract

Probabilistic logics have attracted a great deal of attention during the past few years. Where logical languages have, already from the inception of the field of artificial intelligence, taken a central position in research on knowledge representation and automated reasoning, probabilistic graphical models with their associated probabilistic basis have taken up in recent years a similar position when it comes to reasoning with uncertainty. There are now several different proposals in literature to merge logic and probabilistic graphical models. Probabilistic Horn logic combines Horn logic with probability theory, which yields a probabilistic logic that allows reasoning with classes of Bayesian networks. Bayesian logic is similar in expressive power to probabilistic Horn logic; the main difference is that it is primarily meant as a language for generating Bayesian networks. Finally, Markov logic networks have recently been proposed as a language for generating Markov networks using a model-theoretic interpretation of a logical specification. However, whereas Bayesian networks have an attractive semantics, they suffer from the fact that different Bayesian networks may represent exactly the same independence relation. Markov networks, on the other hand, lack in expressiveness when representing independence information. The formalism of chain graphs is increasingly seen as a natural probabilistic graphical formalism as it generalises both Bayesian networks and Markov networks, and has an attractive semantics in the sense that any Bayesian network has a unique graphical representation as a chain graph. In this paper, a new probabilistic logic, called chain logic, is developed along the lines of probabilistic Horn logic. This new probabilistic logic allows representing subtle independence information that cannot be represented by all previously developed probabilistic logics.

1 Introduction

There has been a considerable amount of work in the field of artificial intelligence during the past two decades on integrating logic and probability theory. This research was motivated by perceived limitations of both formalisms. Logic has for long acted as the common ground for almost all research on knowledge representation and reasoning in artificial intelligence; yet, uncertainty cannot be handled easily in logic. Probabilistic graphical models have been proposed as formalisms for reasoning with uncertainty, taking probability theory as their foundation. Although their associated graphical representation allows specifying relationship among objects in the domain of discourse such that it is possible to reason about their statistical dependences and independences, probabilistic graphical models are essentially propositional in nature, and they lack the representational richness of logics.

Several researchers have proposed probabilistic logics that merge the two types of languages in an attempt to redress their individual shortcomings. A variety of such languages is now available, each of them adopting a different view on the integration. Unfortunately, it appears that all of the available frameworks are still restricted in one way or the other. For instance, probabilistic Horn logic, as originally proposed by Poole in [4], offers a framework that was shown to be as powerful as Bayesian networks, yet it has the advantage that it is a first-order language that integrates probabilistic and logical reasoning in a seamless fashion. However, usually the graphical representation associated with a Bayesian network does not offer a unique

This work has been done as part of the B-Screen project funded by the Netherlands Organisation for Scientific Research (NWO) under BRICKS/FOCUS grant number 642.066.605, and as part of the Octopus project under the responsibility of the Embedded Systems Institute. The Octopus project is partially supported by the Dutch Ministry of Economic Affairs under the Bsik program.

way to represent the independence information, which makes the interpretation of Bayesian networks cumbersome. Bayesian logic programs [1] have similar limitations as probabilistic Horn logic; in addition, they are only proposed as formalisms to specify Bayesian networks in a logical way and reasoning is done in the generated Bayesian networks. Finally, the framework of Markov logic networks [5] has been proposed as a powerful language based on first-order logic to specify Markov networks. Yet, Markov networks are seen by researchers in probabilistic graphical models as the weakest type of such models, as much of the subtleties of representing conditional dependence and independence cannot be handled by Markov networks.

In this paper, we propose modifications and extensions to probabilistic Horn logic, yielding a first-order language that is more expressive than the languages mentioned above, in the sense that the probabilistic models that can be specified and reasoned about have Bayesian networks and Markov networks as special cases. This new probabilistic logic is called *chain logic*.

The organisation of this paper is as follows. In Section 2 we provide an overview of the basic notions of Horn clauses and chain graphs. Section 3 contains an introduction to the chain logic language, with details on its syntax and semantics. Finally, Section 4 contains a comparison to other work and Section 5 presents our conclusions.

2 Preliminaries

2.1 Abduction Logic

Function-free Horn logic is a subset of first-order logic, whose formulae are constructed using *constants* representing individual objects in the domain, *variables* for quantifying over individuals, and *predicates* for representing relations among individuals. Predicates applied to a tuple of terms are called atomic formulae, or *atoms* for short. Formulae in Horn logic, called *Horn clauses*, have the following form:

$$D \leftarrow B_1, \ldots, B_n$$

where $D, B_1, \ldots, B_n$ are atoms, $\leftarrow$ stands for logical implication and the commas ‘,’ stand for conjunction. Sets of Horn clauses are interpreted as conjunctions. All variables appearing in a Horn clause are *universally* quantified. D is called the *head* and $B_1, \ldots, B_n$ constitutes the *body* of the clause. When simultaneously replacing all occurrences of variables in a Horn clause ψ by constants, a so-called *ground* formula results.

Abduction logic is defined as a special variant of function-free Horn logic, where the syntax of Horn clauses is slightly modified, and $\leftarrow$ is given a causal interpretation. *Abduction clauses* have the following form:

$$D \leftarrow B_1, \ldots, B_n : R_1, \ldots, R_m$$

where the predicates of the atoms D and B_i are unary and the atoms R_j, called *templates*, express relationships among variables, where all variables appearing in the atoms D and B_i occur in at least one template R_j. Atoms that do not occur as head of a clause are called *assumables*. From a logical point of view, the ‘:’ operator has the meaning of a conjunction; it is only included in the syntax to allow separating atoms that are templates from non-template atoms. The basic idea is to use unary predicates to represent variables (later referred to as *random* variables), and the templates R_j to represent relations among those variables.

Abduction logic has a standard model-theoretic semantic, defined in terms of the logical consequence operator $\vDash$, and a sound and complete procedural semantics, defined in terms of the deduction relation, indicated by $\vdash$. Let T be a set of abduction clauses, called an *abductive theory* in this paper, then, using model theory or deduction,concluding formula ψ from the theory is denoted by $T \vDash \psi$ and $T \vdash \psi$, respectively.

For abduction logic a special type of logical reasoning has been proposed, called *abduction*, which is defined in terms of model theory or deduction using so-called *explanations*. Let A be the set of all assumables and let A' denote the set of ground instances of A. Given a set of atoms O, interpreted as *observations*, then these observations are explained in terms of the abductive theory and a set of assumables.

Definition 1. *An* explanation *of a set of atoms O based on the pair $\langle T, A\rangle$ is defined as a set of ground assumables $H \subseteq A'$ satisfying the following conditions:*

- *$T \cup H \models O$, and*
- *$T \cup H$ is consistent, i.e., $T \cup H \not\models \bot$.*

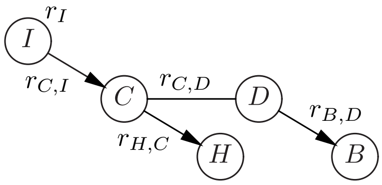

Figure 1: Causal network model of causal and associational knowledge about influenza.

A minimal explanation *H of O is an explanation whose proper subsets are not explanations of O. The set of all minimal explanations is denoted by $\mathcal{E}_T(O)$.*

Consider the following example.

Example 1. *Suppose that we have the following piece of medical knowledge. Influenza (I) causes coughing (C), where coughing is known as a possible cause for hoarseness (H). In addition, coughing is known to be associated with dyspnoea (shortness of breath) (D), although a clear cause-effect relationship is missing. Dyspnoea restricts the oxygen supply to the blood circulation; the resulting low oxygen saturation of the blood will turn the skin to colour blue (B), which is a condition called cyanosis. This qualitative knowledge is represented by the causal network shown in Figure 1. The associated abductive theory T is the following:*

$H(x) \leftarrow C(y) : r_{H,C}(x, y)$
$B(x) \leftarrow D(y) : r_{B,D}(x, y)$
$C(x) \leftarrow I(y) : r_{C,I}(x, y), r_{C,D}(x, z)$
$D(x) \leftarrow: r_{C,D}(y, x)$
$I(x) \leftarrow: r_I(x)$

where each of the variables has $\{f, t\}$ as domain. It now holds that:

$$T \cup \{r_I(t), r_{H,C}(t, t), r_{C,I}(t, t), r_{C,D}(t, t)\} \vDash H(t)$$

and $T \cup \{r_I(t), r_{H,C}(t, t), r_{C,I}(t, t), r_{C,D}(t, t)\} \nvDash \bot$.

The intuition behind the syntax of abduction clauses, such as $C(x) \leftarrow I(y) : r_{C,I}(x, y), r_{C,D}(x, z)$, is that $C(x) \leftarrow I(y)$ expresses the potential *existence of a causal relation between the referred atoms, here $I(y)$ and $C(x)$. Templates R_i, e. g. $r_{C,I}(x, y)$, expresses whether the relationship actually does or does not hold. When there are no atoms to the left of the ':' operator, such as in the clause $D(x) \leftarrow: r_{C,D}(z, x)$, the template represents an association rather than a causal relation.*

2.2 Chain Graphs

A chain graph (CG) is a probabilistic graphical model that consists of labelled vertices, that stand for random variables, connected by directed and undirected edges. This graphical representation allows chain graphs to be considered as a framework that generalises both acyclic directed graph (ADG) models, i.e., Bayesian networks, and undirected graph (UG) models, i.e., Markov networks [3]. The definitions with respect to chain graphs given in this paper are in accordance with Lauritzen [2].

Let $G = (V, E)$ be a *hybrid graph*, where V denotes the set of *vertices* and E the set of *edges*, where an edge is either an *arc* (directed edge), or a *line* (undirected edge). Let indexed lower case letters, e.g., v_1 and v_2, indicate vertices of a chain graph. We denote an arc connecting two vertices by → and a line by '$-$'. Consider two vertices v_1 and v_2. If $v_1 \rightarrow v_2$ then v_1 is a *parent* of v_2. If $v_1 - v_2$ then v_1 is a *neighbour* of v_2. The set of parents and neighbours of a vertex v are denoted by $\mathrm{pa}(v)$ and $\mathrm{ne}(v)$, respectively.

A *path* of length n in a hybrid graph $G = (V, E)$ is a sequence of distinct vertices $v_1, \ldots, v_n$, such that either $v_i - v_{i+1} \in E$ or $v_i \rightarrow v_{i+1} \in E$. A *directed path* is a path which includes at least one arc, and where all arcs have the same direction. A *cycle* is a path where the first and last vertex are the same. A *chain graph* is a hybrid graph with the restriction that no directed cycles exist.

If there is an edge between every pair of vertices in a set of vertices, than this set is named *complete*. A *clique* is a maximally complete subset. Now, consider the graph obtained from a chain graph by removing all its arcs. What are left are vertices connected by lines, called *chain components*; the set of all chain components is denoted here by $\mathcal{C}$.

Associated to a chain graph $G = (V, E)$ is a joint probability distribution $P(X_V)$ that is faithful to the chain graph G, i.e., it includes all the independence information represented in the graph. This is formally expressed by the following *chain graph Markov property*:

$$P(X_V) = \prod_{C \in \mathcal{C}} P(X_C \mid X_{\mathrm{pa}(C)}) \tag{1}$$

with $V = \bigcup_{C \in \mathcal{C}} C$, and where each $P(X_C \mid X_{\mathrm{pa}(C)})$ factorises according to

$$P(X_C \mid X_{\mathrm{pa}(C)}) = Z^{-1}(X_{\mathrm{pa}(C)}) \prod_{M \in M(C)} \varphi_M(X_M) \tag{2}$$

given that $M(C)$ is the complete set in the moral graph[1] obtained from the subgraph $G_{C \cup \mathrm{pa}(C)}$ of G. The functions φ are real positive functions, called *potentials*; they generalise joint probability distributions in the sense that they need not be normalised.

Finally, the normalising factor Z is defined as

$$Z(X_{\mathrm{pa}(C)}) = \sum_{X_C} \prod_{M \in M(C)} \varphi_M(X_M) \tag{3}$$

As a Bayesian network is a special case of a chain graph model, Equation (1) simplifies in that case to:

$$P(X_V) = \prod_{v \in V} P(X_v | X_{\mathrm{pa}(v)}) \tag{4}$$

which is the well-known factorisation theorem of Bayesian networks [2]. In this case, the chain components are formed by a family of random variables. Therefore, for each of those random variables the distribution is defined as the conditional probability function of this variable, given the value of its parents. Note that according to Equation (1), chain graphs can also be interpreted as an ADG of chain components.

3 Chain Logic

3.1 Language Syntax

The formalism presented in this section is inspired by probabilistic Horn logic as introduced by Poole in [4]. For the sake of simplicity, we assume here finite domain specifications. Furthermore, the unique names assumption holds for the different constants of the domain.

Chain logic (CL) extends abduction logic as described in Section 2.1 by interpreting templates as representing uncertain events. The actual definition of the uncertainty is done by means of a *weight* declaration. This is of the form

$$weight(a_1 : w_1, \ldots, a_n : w_n) \tag{5}$$

where a_i represents an atom and $w_i \in \mathbb{R}_0^+$. The set of atoms appearing in such declarations are the assumables, denoted by A. Here we assume that the atoms in a weight declaration share the same variables. For a grounded assumable a, the use of function $\omega(a)$ defines the weight w that is associated to this assumable. We require that a ground atom a – which is an instance of one of the assumables – does not appear as an instance of another assumable in another weight declaration.

In short, the weight declaration defines conjunctions of atoms that are mutually exclusive and exhaustive. Therefore, together with the above elements, a CL specification also includes integrity constraint statements. For instance, clauses of the form

$$\bot \leftarrow a_i \wedge a_j \tag{6}$$

for any pair a_i and a_j appearing in the same weight declaration where $i \neq j$. Such clauses are implicit in all of our given examples. We also allow the addition of another set of constraints referring to a pair of assumables appearing in different weight declarations, as seen in the example below.

[1] Moralisation encompasses: (1) adding lines between unconnected parents of a chain component, and (2) conversion of arcs into lines by ignoring their directions.

Example 2. *Consider the description given in Example 1. Uncertainty is defined by replacing the templates by potential functions. For the abductive theory in this example, they are as follows:*

φ_{CI}	i	$\bar{i}$
c	*4*	*2*
$\bar{c}$	*1*	*10*

φ_{CD}	d	$\bar{d}$
c	*18*	*2*
$\bar{c}$	*5*	*2*

φ_{HC}	c	$\bar{c}$
h	*0.6*	*0.1*
$\bar{h}$	*0.4*	*0.9*

φ_{BD}	d	$\bar{d}$
b	*0.3*	*0.001*
$\bar{b}$	*0.7*	*0.999*

φ_{I}	
i	*0.1*
$\bar{i}$	*0.9*

This example can be represented in chain logic using the following abduction clauses:

$$\begin{array}{ll}
I(x) \leftarrow: \varphi_I(x) & C(x) \leftarrow I(y) : \varphi_{CI}(x,y) \wedge \varphi_{CD}(x,z) \\
D(y) \leftarrow: \varphi_{CD}(x,y) & H(x) \leftarrow C(y) : \varphi_{HC}(x,y) \\
B(x) \leftarrow D(y) : \varphi_{BD}(x,y) & \perp \leftarrow \varphi_{CI}(x,y) \wedge \varphi_{CD}(\bar{x},z)
\end{array}$$

Furthermore, we can associate weights to the assumables according to the potential functions. For instance, considering φ_{CD}, we have:

$$weight(\varphi_{CD}(t,t) : 18, \varphi_{CD}(t,f) : 2, \varphi_{CD}(f,t) : 5, \varphi_{CD}(f,f) : 2)$$

In order to be able to probabilistically interpret a CL theory T, a number of assumptions are added to those of abduction logic:

1. the theory is acyclic: if T' is the set of ground instances of elements of T, it is possible to assign a natural number to every ground atom such that for every rule in T' the atoms in the body of the rule are strictly less than the atom in the head;

2. the rules for every ground non-assumable represented in T' are covering, i.e., there is always a rule whose assumable holds – which is used as an explanation of the atom in the head;

3. the bodies of the rules in T' for an atom are mutually exclusive;

4. there is a set of ground assumables, one from every grounded weight declaration, consistent with T.

As in Poole's probabilistic Horn logic, these assumptions are not intended to be enforced by the system: it is up to the modeller to comply to these requisites. Under this condition, we can then guarantee the probabilistic properties of the theory, as we show in the next section.

3.2 Probabilistic Reasoning

In this section, we show how we can infer probabilities from a chain logic theory. A conjunction of ground instances, one for each assumable of a grounded weight declaration, is called a *state*. The set of all such states is denoted by S. The set of *consistent states*, with respect to a theory T, will be denoted by CS, i.e., CS $= \{s \in S \mid T \cup \{s\} \nvDash \perp\}$. The last assumption mentioned in the previous section can be expressed formally by CS $\neq \varnothing$.

Definition 2. *Let P_T be a real positive function of S that is defined as follow:*

$$P_T(s) = \begin{cases} \frac{1}{Z} \prod_{a \in s} \omega(a) & \textit{if } s \in \text{CS} \\ 0 & \textit{otherwise} \end{cases}$$

where $Z = \sum_{s \in \text{CS}} \prod_{a \in s} \omega(a)$.

Clearly, the function P_T obeys the axioms of probability theory, as each weight is larger than or equal to 0 and, given that CS $\neq \varnothing$, it follows that $\sum_{s \in S} P_T(s) = 1$, and is thus a joint probability distribution; P_T is sometimes abbreviated to P in the following.

Given T, a minimal explanation e of some formula ψ is equivalent to a disjunction of consistent states, i.e., $T \models e \equiv \bigvee_i s_i$ with $s_i \in$ CS. As all s_i are mutually exclusive, it follows that:

$$P_T(e) = P_T(\bigvee_i s_i) = \sum_i P_T(s_i)$$

which offers the means to assign a probability to minimal explanations. In order to assign a probability to a formula, we have the following result.

Theorem 1. *Under the assumptions mentioned in Section 3.1, if $\mathcal{E}_T(\psi)$ is the set of minimal explanations of the conjunction of atoms ψ from the chain logic theory T, then:*

$$P_T(\psi) = \sum_{e \in \mathcal{E}_T(\psi)} P(e)$$

Proof. This follows exactly the same line of reasoning of [4, page 53, proof of Theorem A.13]. □

This result shows that P is indeed a probability distribution over conjunctions of formulae if we use the definition of P_T above. Other probabilities can be calculated on the basis of these types of formulae, such as conditional probabilities which can be calculated according to the definition of conditional probabilities $P(a|b) = \frac{P(a \wedge b)}{P(b)}$. Below, we will sometimes refer to the resulting probability distribution by P_T in order to stress that we mean the probability calculated using Definition 2.

Example 3. *Reconsider the uncertainty specification concerning influenza as described in Example 2. We will illustrate how probabilities can be calculated from the explanations obtained from an abductive scheme. Consider here that we are interested in calculating the $P(B(t))$ (i.e., the probability of B assumes truth-value t). Recalling the definitions provided in Section 2.1, we obtain the minimal explanations for $B(t)$, i.e., $\mathcal{E}_T(B(t))$ as the set with the following 4 members:*

$$\begin{array}{ll} \{\varphi_{BD}(t,t), \varphi_{CD}(t,t)\} & \{\varphi_{BD}(t,t), \varphi_{CD}(f,t)\} \\ \{\varphi_{BD}(t,f), \varphi_{CD}(t,f)\} & \{\varphi_{BD}(t,f), \varphi_{CD}(f,f)\} \end{array}$$

We can then sum over the states that are consistent with these explanations (by extending the explanations with consistent instances of φ_I, φ_{CI}, and φ_{HC}[2]):

$$P(B(t)) = \sum_{e \in \mathcal{E}_T(D(t))} = Z^{-1}(0.3 \cdot 18 \cdot 0.1 \cdot 4 \cdot 1 + 0.3 \cdot 18 \cdot 0.9 \cdot 2 \cdot 1 + 0.3 \cdot 5 \cdot 0.1 \cdot 1 \cdot 1 \ldots) = 25.756/Z$$

Notice that explanations are extended in order to incorporate other consistent influences. For instance, reasoning about the probability of B being true might only include φ_{BD} in the explanations. However, φ_{CI} and φ_I – which are relevant for such computation – are also taken into account.

3.3 Specification of Chain Graphs

In this section, we present the formal relation between chain graphs with discrete random variables and chain logic. For the sake of simplicity, we focus on chain graphs with binary variables, i.e., the set of constants is $\{t, f\}$, although the theory generalises to arbitrary arities. Complementary constants are denoted with a bar, i.e., $\bar{t} = f$ and $\bar{f} = t$.

The translation from a chain graph G to a chain logic theory T is as follows. Consider a vertex v in G. For each component $C \in \mathcal{C}$ of G, there is a set of potential functions defined on the moral graph of the sub-graph $G_{C \cup \text{pa}(C)}$ which contains v. This set of potential functions is denoted by $\Phi(G, C, v)$. For every vertex v, we have the following formula in T:

$$V(x) \leftarrow \bigwedge\{V'(x_{v'}) \mid v' \in \text{pa}(v)\} : \bigwedge\{\varphi_M(x_1, \ldots, x, \ldots, x_n) \mid \varphi_M \in \Phi(G, C, v)\}$$

and we ensure that each of the predicates defined for the same random variable shares that variable in the formula. However, this is not strictly necessary as different values for the same random variable in a component is also disallowed by the integrity constraints.

The integrity constraints are defined as follows. If we have two potential functions, namely an n-ary $\varphi_M(\ldots, v, \ldots)$ and an m-ary $\varphi'_M(\ldots, v, \ldots)$, i.e., which share a variable v *in the same chain component* (i.e., not between chain components), then we add the following formula to T:

$$\bot \leftarrow \varphi_M(x_0, \ldots, x, \ldots, x_n) \wedge \varphi'_M(x'_0 \ldots, \bar{x}, \ldots, x'_m)$$

for each variable that they share. As mentioned earlier, this ensures we do not generate explanations which have inconsistent assignments to the random variables within the same chain component.

Finally, for each potential function φ_M, we define a weight declaration containing $\varphi_M(c_0, \ldots, c_n) : w$ if $\varphi_M(X_M = (c_0, \ldots, c_n)) = w$.

[2] Notice that $\{\varphi_{BD}(t,t) \cdot \varphi_{CD}(t,t) \cdot \varphi_I(t) \cdot \varphi_{CI}(t,t) \cdot \varphi_{HC}(t,t)\} + \{\varphi_{BD}(t,t) \cdot \varphi_{CD}(t,t) \cdot \varphi_I(t) \cdot \varphi_{CI}(t,t) \cdot \varphi_{HC}(f,t)\} = \{\varphi_{BD}(t,t) \cdot \varphi_{CD}(t,t) \cdot \varphi_I(t) \cdot \varphi_{CI}(t,t)\}$, as $\varphi_{HC}(t,t) + \varphi_{HC}(f,t) = 1$.

Example 4. *Consider again our influenza domain example. Consider further that we are interested in the probability of $P(I(t) \mid B(t))$ (i.e., the conditional probability of I being true given that B is true). This probability can be obtained by the having $P(I(t) \wedge B(t))$ divided by $P(B(t))$. The calculation of $P(B(t))$ was shown in Example 3. Therefore, we follow by calculation the minimal explanations for $I(t) \wedge B(t)$, i.e., $\mathcal{E}_T(I(t) \wedge B(t))$ is a set with the following 4 members:*

$$\begin{array}{ll} \{\varphi_{BD}(t,t), \varphi_{CD}(t,t), \varphi_I(t)\} & \{\varphi_{BD}(t,t), \varphi_{CD}(f,t), \varphi_I(t)\} \\ \{\varphi_{BD}(t,f), \varphi_{CD}(t,f), \varphi_I(t)\} & \{\varphi_{BD}(t,f), \varphi_{CD}(f,f), \varphi_I(t)\} \end{array}$$

Following the same reasoning as before, we obtain that $P(I(t) \wedge B(t)) = 2.32/Z$. Finally, we have that $P(I(t) \wedge B(t))/P(B(t) = (2.32/Z)/(25.756/Z) \simeq 0.0901$[3].

In the following theorem, we establish that probabilities calculated from the chain logic theory corresponds to the chain graph semantics.

Theorem 2. *Suppose $v_1, \ldots, v_n$ are vertices in a chain graph, with T as the corresponding chain logic theory by the translation described above, then:*

$$P(X_{v_1} = c_1, \ldots, X_{v_n} = c_n) = P_T(V_1(c_1), \ldots, V_n(c_n))$$

Proof. There is only one minimal explanation of $V_1(c_1) \wedge \cdots \wedge V_n(c_n)$, namely $\varphi_M(c_0^M, \ldots, c_m^M)$ for all potential functions in cliques in the moral graphs of chain components with their parents, such that the constants filled into the potential functions correspond to the values for each of the random variables.

The explanation describes exactly one state. Denote this as s. As the potential functions are related to exactly one component, we have the following equation:

$$\prod_{a \in s} \omega(a) = \prod_{C \in \mathcal{C}} \prod_{\varphi_j^C(c_0^j, \ldots, c_n^j) \in s} \varphi_j^C(X_{v_0^j} = c_0^j, \ldots, X_{v_n^j} = c_n^j) = \prod_{C \in \mathcal{C}} \prod_{M \in M(C)} \varphi_M(X_M) \tag{7}$$

where φ^C are potential functions defined for component C and $M(C)$ are the complete sets in the moral graph from the sub-graph $G_{C \cup \mathrm{pa}(C)}$.

Let $Z = \sum_{s \in \mathrm{CS}} \prod_{a \in s} \omega(a)$. Since there are no integrity constraints between variables in chain components (i.e., combinations of consistent potential functions which are in different chain components are consistent), we have that:

$$Z = \sum_{s \in \mathrm{CS}} \prod_{a \in s} \omega(a) = \prod_{C \in \mathcal{C}} \sum_{s \in \mathrm{CS}(C)} \prod_{\varphi_j^C(c_0^j, \ldots, c_n^j) \in s} \varphi_j^C(X_{v_0^j} = c_0^j, \ldots, X_{v_n^j} = c_n^j) = \prod_{C \in \mathcal{C}} Z(X_{\mathrm{pa}(C)}) \tag{8}$$

where $\mathrm{CS}(C)$ is the set of consistent states (w.r.t. T) restricted to the potential functions in that chain component. Then, the equivalence follows in the following way:

$$\begin{array}{lll} P(X_{v_1} = c_1, \ldots, X_{v_n} = c_n) & = \prod_{C \in \mathcal{C}} P(X_C \mid X_{\mathrm{pa}(C)}) & \text{(factorisation)} \\ & = \prod_{C \in \mathcal{C}} Z^{-1}(X_{\mathrm{pa}(C)}) \prod_{M \in M(C)} \varphi_M(X_M) \cdot & \text{(factorisation)} \\ & = \left(\prod_{C \in \mathcal{C}} Z^{-1}(X_{\mathrm{pa}(C)})\right) \prod_{C \in \mathcal{C}} \prod_{M \in M(C)} \varphi_M(X_M) & \text{(arithmetic)} \\ & = Z^{-1} \prod_{C \in \mathcal{C}} \prod_{M \in M(C)} \varphi_M(X_M)) & \text{(Eq. 8)} \\ & = Z^{-1} \prod_{a \in w} \omega(a) & \text{(Eq. 7)} \\ & = P_T(V_1(c_1), \ldots, V_n(c_n))) & \text{(def. } P_T\text{)} \end{array}$$

□

As we have shown in Section 3.2 that P_T adheres to the axioms of probability theory, chain graphs and the translated chain logic theory agree on all probabilities. This result shows that chain graphs can be translated to chain logic specifications. The converse is also true: all chain logic programs, which adhere to the assumptions of Section 3.1, correspond to a chain graph – at least in a trivial sense – as a fully connected Markov network models and the associated probability distributions, which can be derived from the chain logic semantics. However, the independence information that is implicit in the chain logic specification will not be represented.

[3] Even though not used here, remember that Z is calculated according to the set of consistent states, i. e., all the possible instantiations of $\{t, f\}$ in the potential functions which satisfy the integrity constraints and follows domain descriptions.

4 Related Work

As mentioned in Section 3.1, the language presented here is inspired by Poole's probabilistic Horn logic [4]. Besides some changes in the terminology (such as using *weight* declarations in place of *disjoint* ones), the main differences in terms of syntax is the set of integrity constraints allowed and the probabilistic information captured in each formalism. Also, in probabilistic Horn logic the disjoint declarations should sum up to 1, whereas weights can sum up to any value. This enabled the formalisation of potential functions instead of a (normalised) probability distribution. In terms of independence, in Poole's definition instantiations of hypotheses that appear in different disjoint declarations are independent. In our case, by allowing the use of extra integrity constraints, we are able to establish dependences among such hypotheses (cf. Example 2).

In fact, those differences extend Poole's approach and allow us to obtain a more generic probabilistic model, being crucial for the representation of chain graph models. By using potential functions we can represent the quantitative influence between variables in a clique. The additional integrity constraints guarantee that instantiations of those potentials functions appear consistently in each explanation.

Despite such differences, we still share with Poole's approaches some assumptions and similar results, for instance, with respect to the probability densities defined over the theory. One additional assumption in chain logic, namely assumption (4) in Section 3.1, is not present in probabilistic Horn logic since this property is true for any probabilistic Horn logic theory.

As a first-order language-based formalism for probabilistic graphical models, we can also relate our work to, for instance, Bayesian logic programs [1] and Markov logic [5]. We present here a simple language that can be used for the specification of both Bayesian and Markov network models, in such a way that the logical specification is more that a generative language for the model at hand, maintaining a close relation between logical and probabilistic reasoning – without loss of expressiveness.

5 Final Considerations

In this paper we presented a simple and yet powerful language for specifying and reasoning about chain graphs. Besides being able to incorporate both Bayesian and Markov network models as special cases, we maintain a strong relation between logical and probabilistic reasoning.

Our language still presents some restrictions. First, we use finite set of constants, which prohibits the use of continuous variables. For Markov logic networks, it has been shown that special cases of such networks can be extended to infinite domains by defining a Gibbs measure over sets of interpretations for logical formulae [6]. A similar approach could be taken here by defining a measure over the set of consistent states. Another limitation is the acyclicity assumption, which restricts the explicit representation of undirected graphs components. Even though we require certain assumptions for a sound probabilistic interpretation, weakening acyclicity seems feasible.

While we have shown in this paper that chain logic is powerful enough to define and reason about chain graphs, we have no strong reason to suspect that chain logic is restricted to this class of probabilistic graphical models. While chain graphs is a fairly general class of graphs, it might be the case that the language is applicable to a broader set of graphs. Furthermore, modelling the independence implied in chain logic theories into a graphical model is an open question that will be investigated further.

References

[1] K. Kersting and L. de Raedt. Bayesian logic programs. Technical Report 151, Institute for Computer Science - University of Freiburg, 2001. CoRR cs.AI/0111058.

[2] S. L. Lauritzen. *Graphical Models*. Oxford:Clarendon, 1996.

[3] J. Pearl. *Probabilistic Reasoning in Inteligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[4] D. Poole. Probabilistic Horn abduction and Bayesian networks. *AI Journal*, 64(1):81–129, 1993.

[5] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.

[6] P. Singla and P. Domingos. Markov logic in infinite domains. In *Proc. of UAI'07*, pages 368–375. AUAU Press, 2007.