Appendix A Proofs

In the appendix we collate the proofs of the claims made in the paper.

Proof of Proposition 6. We have seen that the independent distribution induced on the space of $\mathcal{R}$-structures with domain D is projective. Additionally, closed quantifier-free formulas hold in a substructure if and only if they hold in the original structure. So let ω be an $\mathcal{S}$-structure with domain D and let $D \subset D'$. Let $\omega_{(R)}$ be the $\mathcal{R}$-structure on ω. If ω has probability 0 because $\omega \models \exists_{\vec{a}\in\vec{D}}\exists_{R\in\mathcal{S}\setminus\mathcal{R}} : R(\vec{a}) \not\Leftrightarrow \phi_R(\vec{a})$, then so will all superstructures of ω since existential formulas are closed under superstructure. So assume this is not the case. Then

$$Q_T^{D'}\left(\left\{\omega'_{(R)} \in \Omega_n | \omega_{(R)} \text{ is the substructure of } \omega'_{(R)} \text{ with domain } \{1,\ldots,n\}\right\}\right) = Q_T^D(\omega_{(R)}).$$

Every $\omega'_{(R)}$ has a unique extension to an $\mathcal{S}$-structure ω' with $R(\vec{a}) \not\Leftrightarrow \phi_R(\vec{a})$ for all $R \in \mathcal{S}\setminus\mathcal{R}$. Since quantifier-free formulas with values in $\omega_{(R)}$ are true in $\omega'_{(R)}$ if and only if they are true in $\omega_{(R)}$, those are exactly the extensions of ω to D' that have non-zero weight. □

Proof of Proposition 11. The first claim follows from the fact that RANDOM($\mathcal{T}$) is the reduct of RANDOM($\mathcal{R}$) to $\mathcal{T}$ for any $\mathcal{T} \subseteq \mathcal{R}$.

To show the second claim, consider the vocabulary $\bar{\mathcal{R}}$ containing $R_{\vec{x}}(\vec{y})$ for every atomic subformula $R(\vec{x},\vec{y})$ of φ and let $\bar{\varphi}$ be the $\bar{\mathcal{R}}$-formula obtained from φ by replacing every occurrence of $R(\vec{x},\vec{y})$ with $R_{\vec{x}}(\vec{y})$. Let M be a model of RANDOM($\mathcal{R}$) and let $\vec{a} \in M$. Then define an $\bar{\mathcal{R}}$-structure on M by setting $R_{\vec{x}}(\vec{y}) :\Leftrightarrow R(\vec{a},\vec{y})$. One can verify that M satisfies the extension axioms in RANDOM($\bar{\mathcal{R}}$). Since RANDOM($\bar{\mathcal{R}}$) is complete, RANDOM($\bar{\mathcal{R}}$) $\vdash \bar{\varphi}$ or RANDOM($\bar{\mathcal{R}}$) $\vdash \neg\bar{\varphi}$. Therefore, either $\varphi(\vec{a})$ or $\neg\varphi(\vec{a})$ holds uniformly for all $\vec{a} \in M$. Therefore, either RANDOM($\mathcal{R}$) $\vdash \forall_{\vec{x}}\varphi(\vec{x})$ or RANDOM($\mathcal{R}$) $\vdash \forall_{\vec{x}}\neg\varphi(\vec{x})$. □

Proof of Theorem 23. By Fact 22 and the finiteness of the vocabulary $\mathcal{V}$, there is a finite set G of extensions axioms over $\mathcal{R}$ such that there are quantifier-free $\mathcal{R}$-formulas ϕ'_R for every $R \in \mathcal{V}\setminus\mathcal{R}$ with $G \vdash \forall_{\vec{x}}\phi_R(\vec{x}) \leftrightarrow \phi'_R(\vec{x})$.

By Fact 12, $\lim\limits_{n\to\infty} Q_T^{(n)}(\{\omega \in \Omega_n | \omega_{\mathcal{R}} \models G\}) = 1$ for any finite subset $G \subseteq$ RANDOM($\mathcal{R}$) and thus $\lim\limits_{n\to\infty} Q_T^{(n)}(\{\omega \in \Omega_n | \forall_{R\in\mathcal{V}\setminus\mathcal{R}}\omega_{\mathcal{R}} \models \phi_R \leftrightarrow \phi'_R\}) = 1$. Let $(Q_T^{(n)})$ be the family of distributions induced by the quantifier-free FO-distribution over $\mathcal{R}$, in which every ϕ_R is replaced by ϕ'_R. By construction, $Q^{(n)}(\omega) = Q'^{(n)}(\omega)$ for every world ω with $\forall_{R\in\mathcal{V}\setminus\mathcal{R}}\omega_{\mathcal{R}} \models \phi_R \leftrightarrow \phi'_R$. Therefore, $\sup\limits_{A\subseteq\Omega_n} |Q^{(n)}(A) - Q'^{(n)}(A)|$ is bounded by above by $1 - Q^{(n)}(\{\omega \in \Omega_n | \forall_{R\in\mathcal{V}\setminus\mathcal{R}}\omega_{\mathcal{R}} \models \phi_R \leftrightarrow \phi'_R\})$, which limits to 0 since $\lim\limits_{n\to\infty} Q^{(n)}(\{\omega \in \Omega_n | \forall_{R\in\mathcal{V}\setminus\mathcal{R}}\omega_{\mathcal{R}} \models \phi_R \leftrightarrow \phi'_R\}) = 1$. □

Proof of Theorem 27. Let $\mathcal{S}\setminus\mathcal{R}$ be the extensional vocabulary of the probabilistic logic program Θ and let Π be its underlying Datalog program. Then for every relation $R \in \mathcal{S}\setminus\mathcal{R}$, $R(\vec{t})$ is given by the Datalog formula $(\Pi, R)\vec{t}$ over any given $\mathcal{R}$-structure. By Fact 21, $(\Pi, R)\vec{t}$ is equivalent to an LFP formula ϕ_R over $\mathcal{R}$. Let T be the abstract LFP distribution over $\mathcal{R}$ in which for every $R \in \mathcal{R}$, q_R is taken from Θ and for every $R \in \mathcal{S}\setminus\mathcal{R}$, this ϕ_R is used. Then T and Θ induce equivalent families of distributions. By Theorem 23, T is asymptotically equivalent to a quantifier-free abstract distribution, which in turn is equivalent to a determinate probabilistic logic program by Fact 25. Therefore Θ itself is asymptotically equivalent to a determinate probabilistic logic program. □

Proof of Proposition 28. We will proceed by contradiction. So assume not. Then there is an m such that $Q^{(m)}$ and $Q'^{(m)}$ are not equal. Let ω_0 be a world of size m which does not have the same probability in $Q^{(m)}$ and $Q'^{(m)}$. Let $a := |Q^{(m)}(\{\omega_0\}) - Q'^{(m)}(\{\omega_0\})|$ For any $n \geq m$, consider the subset

$$A_n := \{\omega \in \Omega_n | \text{the substructure of } \omega \text{ on the domain } \{1, \ldots, m\} \text{ is } \omega_0\}.$$

Since both families are projective, $|Q^{(n)}(A_n) - Q'^{(n)}(A_n)| = |Q^{(m)}(\{\omega_0\}) - Q'^{(m)}(\{\omega_0\})| = a$. Therefore, $(Q^{(n)})$ and $(Q'^{(n)})$ are not asymptotically equivalent. □

Proof of Proposition 30. Let $(Q^{(n)})$ and $(Q'^{(n)})$ be asymptotically equivalent families of distributions over $\mathcal{S}$. Then for any $\mathcal{T} \subseteq \mathcal{S}$, and any $A \subseteq \Omega_n$,

$$|Q^{(n)}_{\mathcal{T}}(A) - Q'^{(n)}_{\mathcal{T}}(A)| = |Q^{(n)}(\{\omega \in \Omega^{\mathcal{S}}_n | \omega_{\mathcal{T}} \in A\}) - Q'^{(n)}(\{\omega \in \Omega^{\mathcal{S}}_n | \omega_{\mathcal{T}} \in A\})|.$$

Therefore, $\lim_{n\to\infty} \sup_{A \subseteq \Omega^{\mathcal{T}}_n} |Q^{(n)}_{\mathcal{T}}(A) - Q'^{(n)}_{\mathcal{T}}(A)| \leq \lim_{n\to\infty} \sup_{A \subseteq \Omega^{\mathcal{S}}_n} |Q^{(n)}(A) - Q'^{(n)}(A)| = 0$ □

Proof of Theorem 31. By asymptotic quantifier elimination, we can choose T_q to be asymptotically equivalent to T. Since T_q is a quantifier-free distribution, its induced family of distributions $(P^{(n)})$ is projective. By Proposition 30, $Q^{(n)}_{\mathcal{S}'}$ and $P^{(n)}_{\mathcal{S}'}$ are asymptotically equivalent. However, since they are both projective, this implies that they are actually equivalent everywhere. □

Proof of Proposition 33. Let the abstract distribution be defined over $\mathcal{R}$. Then by replacing occurrences of other relations with their quantifier-free definitions, we can reduce to the case where all formulas and structures mentioned are $\mathcal{R}$-formulas and $\mathcal{R}$-structures. Since by definition of the abstract distribution semantics, $\{1, \ldots, n\} \models \varphi$ and $\{n+1, \ldots, n+m\} \models \psi$ are independent for $\mathcal{R}$-structures, this suffices to show IP.

To show CIP, observe first that for all atoms $R(x_1, \ldots, x_k)$ and $n_1, \ldots, n_k \in \{1, \ldots, n-1\}$, either $\omega \models R(a_1, \ldots, a_k)$ or $\omega \models \neg R(a_1, \ldots, a_k)$. Therefore, we can replace all occurrences of atoms with entries in $x_1, \ldots x_{n-1}$ with $\top$ or $\bot$, depending on whether ω satisfies them under the substitution $x_i \to n_i$. As only atoms remain in which x_n occurs freely, their interpretations refer to n or $n+1$ in $\{1, \ldots, n\}$ and $\{1, \ldots, n+1\} \setminus \{n\}$ respectively. Now we can conclude as for IP above. □

Proof of Corollary 34. Such a projective family is in fact induced by a determinate probabilistic logic program, which is equivalent to a quantifier-free family of distributions. □

Proof of Corollary 35. $(Q^{(n)}_{\mathcal{S}'})$ satisfies IP since $(Q^{(n)})$ does and IP transfers to reducts. □

Proof of Proposition 36. As in the previous proofs, we can reduce to the situation where nullary predicates are the propositional facts. Since there are only finitely many nullary predicates in $\mathcal{S}'$, there are only finitely many possible configurations of those nullary predicates. For every such configuration φ, let q_φ be the probability of that configuration. Then the distribution itself is given by the finite sum of the conditional distributions on φ, weighted by q_φ , and every such conditional distribution is given by the probabilistic logic program without nullary relations obtained by substituting $\top$ or $\bot$ for the nullary propositions, depending on whether they are true or false in the configuration φ. $\square$

Proof of Proposition 38. Assume there is such a probabilistic logic program. Since $\mathfrak{m}^*$ is projective, it would have to be a finite sum of distributions satisfying IP. For each of these finite components $(P^{(n)})$, let p_P be the unconditional probability of $R(x)$ for any x (well-defined by projectivity). We observe from Equation 5.1 that for variable n, the infimum of $\mathfrak{m}^*\left(R(a_{n+1})|\left\{R(a_i)\right\}_{i\in I\subseteq\{1,\dots,n\}}\cup\left\{\neg R(a_i)\right\}_{i\in\{1,\dots,n\}\setminus I}\right)$ is 0, even if we assume that there is at least one i with $R(a_i)$. As there are only finitely many components $(P^{(n)})$, the infimum c of the nonzero p_P is greater than 0. By the IP for the $(P^{(n)})$, $R(a_{n+1})$ is conditionally independent of $\{R(a_i)\}_{i\in I\subseteq\{1,\dots,n\}}\cup\{\neg R(a_i)\}_{i\in\{1,\dots,n\}\setminus I}$ under $(P^{(n)})$. Thus, the conditional probability of $R(a_{n+1})$ given $\left\{R(a_i)\right\}_{i\in I\subseteq\{1,\dots,n\}}\cup\left\{\neg R(a_i)\right\}_{i\in\{1,\dots,n\}\setminus I}$ is a weighted mean of the non-zero p_P and therefore bounded below by $c > 0$, *in contradiction to 0 being the infimum of* $\mathfrak{m}^*\left(R(a_{n+1})|\left\{R(a_i)\right\}_{i\in I\subseteq\{1,\dots,n\}}\cup\left\{\neg R(a_i)\right\}_{i\in\{1,\dots,n\}\setminus I}\right)$. $\square$

Appendix B Background and notation

B.1 First-order logic

This paper follows the notation of Ebbinghaus and Flum (2006), which we outline here. Full information can be found in Chapter 1 there. A *vocabulary*, sometimes called a *relational* vocabulary for emphasis, is a finite set of relation symbols, each of which are assigned a natural number arity, and of constant symbols, but does not contain function symbols. We also assume an infinite set of first-order variables, customarily referred to by lower-case letters from the end of the alphabet, i. e. u to z. For a vocabulary $\mathcal{S}$, an *atomic $\mathcal{S}$-formula* or *$\mathcal{S}$-atom* is an expression of the form $R(t_1,\dots,t_n)$, where R is a relation symbol of arity n and every t_i is either a variable or a constant. An *$\mathcal{S}$-literal* is either an atom φ or its negation $\neg\varphi$. A *quantifier-free $\mathcal{S}$-formula* is a Boolean combination of atoms, where conjunction is indicated by $\wedge$, disjunction by $\vee$ and logical implication by $\rightarrow$. We use the big operators $\bigwedge$ A *first-order* or *FOL-formula* is made up from atoms using Boolean connectives as well as existential and universal quantifiers over variables x, indicated by $\exists_x$ and $\forall_x$ respectively. To simplify the notation for longer strings of quantifiers, we use the shorthand $\forall_{x_1,\dots,x_n}$ for $\forall_{x_1}\dots\forall_{x_n}$, and analogously for $\exists$.

In Section 3 we also refer to second-order formulas, which are introduced there.

Let φ be a first-order formula. An occurrence of a variable x in φ is called *bound* if it is in the scope of a quantifier annotated with that variable and *free* otherwise. x is called *free in* φ if it occurs freely. We use the notation $\varphi(x_1,\dots,x_n)$ for φ to assert that every free variable in φ is from $x_1,\dots,x_n$. We also abuse notation and write $R(\vec{x},\vec{c})$ for an atomic formula with constants $\vec{c}$ and free variables $\vec{x}$, even though they don't necessarily appear in that order. A formula with no free variables is called a *sentence*, and a set of sentences is called a *theory*. Sentences making up a given theory are also called its *axioms*.

Since tuples such as $x_1,\dots,x_n$ occur frequently and their exact length is often not important, we use the notation $\vec{x}$ to indicate a tuple of arbitrary finite length in many contexts.

If $\mathcal{S}$ is a vocabulary, then an *$\mathcal{S}$-structure* ω consists of a finite non-empty set D, the *domain* of ω, along with an interpretation of the relation symbols and constants of $\mathcal{S}$ as relations and elements of D respectively. If R is a relation symbol and c a constant, then we write R_ω and c_ω for their respective interpretations in ω. A bijective map $f : D \to D'$ between the domains of two $\mathcal{S}$-structures ω and ω' respectively is an *$\mathcal{S}$-isomorphism* if it maps the interpretation of every relation symbol and constant in ω to the interpretation of the same symbol in ω'. Given a subset $D' \subseteq D$ of the domain of a structure ω, we call a structure ω' with domain D' the *substructure* of ω on D' if the interpretation of the relation symbols in ω' are obtained by restricting the interpretations of the symbols in ω to D'. Let $\mathcal{S}' \subseteq \mathcal{S}$ be two vocabularies. Then the *reduct* $\omega_{\mathcal{S}'}$ of an $\mathcal{S}$-structure ω is given by simply omitting the interpretations of the symbols not in $\mathcal{S}$. In this situation, we call ω an *extension* of $\omega_{\mathcal{S}'}$.

Let ω be an $\mathcal{S}$-structure, let $\varphi(x_1, \dots, x_n)$ be a first-order $\mathcal{S}$-formula and let $a_1, \dots, a_n$ be a tuple of elements of the domain D of ω. Then we write $\omega \models \varphi(a_1, \dots, a_n)$ whenever $\varphi(x_1, \dots, x_n)$ *holds* with respect to the interpretation of $a_1, \dots, a_n$ for $x_1, \dots x_n$, and call ω a *model* of $\varphi(a_1, \dots, a_n)$. Similarly, for a theory T, we call ω a *model* of T if $\omega \models \varphi$ for all axioms φ in T. We also express this situation by saying that ω *satisfies* φ or T.

Let T be a theory and φ a sentence. We use the notation $T \vdash \varphi$ to indicate that $\omega \models \varphi$ for all models ω of T.

B.2 Logic Programming

Our terminology for logic programs is taken from Chapter 9 of Ebbinghaus and Flum (2006), where one can find a more detailed exposition.

A *general logic program* in a vocabulary $\mathcal{S}$ is a finite set Π of clauses of the form $\gamma \leftarrow \gamma_1, \dots, \gamma_n$, where $n \geq 0$, γ is an atomic formula and $\gamma_1, \dots, \gamma_n$ are literals. We call γ the *head* and $\gamma_1, \dots, \gamma_n$ the *body* of the logic program. The *intensional* relation symbols are those that occur in the head of any clause of a program, while the relation symbols occurring only in the body of clauses are called *extensional*. We write $(\mathcal{S}, \Pi)_{\text{ext}}$ for the extensional vocabulary, or $\mathcal{S}_{\text{ext}}$ where the logic program is clear from context.

An *acyclic* logic program is one in which no intensional relation symbol occurs in the body of any clause (This is at first glance a stronger condition than the usual definition of acyclicity, but by successively unfolding head atoms used in the body of a clause every acyclic logic program in the usual sense can easily be brought to this form).

A *pure Datalog program* is a general logic program in which no intensional relation symbol appears within any negated literal. To affix a meaning to a pure Datalog program, consider it as a function which take as input a finite $\mathcal{S}_{\text{ext}}$-structure ω and returns as output an extension ω_Π of ω to $\mathcal{S}$. Starting from an empty interpretation of the relation symbols not in $\mathcal{S}_{\text{ext}}$, we successively expand them by applying the rules of Π. We give an informal description of this process:

Let $\gamma(\vec{x})$ be the head of a clause and $\gamma_1(\vec{x}, \vec{y}), \dots, \gamma_n(\vec{x}, \vec{y})$ the body, and let $\vec{a}$ be a tuple of elements of the domain D of length equal to $\vec{x}$. Then whenever there is a tuple $\vec{b}$ such that $\omega \models \gamma_i(\vec{a}, \vec{b})$ for every i, we add $\gamma(\vec{a})$ to the interpretation of the relation symbol R of the atom γ. Successively proceed in this manner until nothing can be added by applying any of the clauses of Π. Since the domain of ω is finite, this is bound to happen eventually.

As the restriction for no intensional relation symbol to occur negated in the body of a clause turns out to be too strong for many practical applications, we consider a generalisation to *stratifiable* Datalog programs. A general logic program Π in a vocabulary $\mathcal{S}$ is called a *stratifiable Datalog program* if there is a partition of $\mathcal{S}$ into subvocabularies $\mathcal{S}_{\text{ext}} = \mathcal{S}_0, \ldots \mathcal{S}_n$ such that the following holds:

The corresponding logic programs $\Pi_1, \ldots, \Pi_n$, where Π_i is defined as the set of clauses whose head atom starts with a relation symbol from $\mathcal{S}_i$, are pure Datalog programs, and the extensional vocabulary of Π_i is contained in $\mathcal{S}_0 \cup \cdots \cup \mathcal{S}_{i-1}$.

If Π is a stratifiable Datalog program in $\mathcal{S}$ and ω an $\mathcal{S}_{\text{ext}}$-structure, then ω is obtained by applying $\Pi_1, \ldots, \Pi_n$ successively.

A *(pure/stratified) Datalog formula* is an expression of the form $(\Pi, P)\vec{x}$, where Π is a (pure/stratified) Datalog program, P an intensional relation symbol and $\vec{x}$ a tuple of variables. We say that a Datalog formula *holds* in an $\mathcal{S}_{\text{ext}}$-structure ω for a tuple of elements $\vec{a}$ of the same length as $\vec{x}$, written $\omega \models (\Pi, P)\vec{a}$, if $P(\vec{a})$ is true in ω_Π.