which is a finite set of rules with modal atoms. Following [3] our focus here is on nondisjunctive rules.

A (normal MKNF) rule $r$ is of the form: $\mathbf{K}H \leftarrow \mathbf{K}A_1, \ldots, \mathbf{K}A_m, \mathbf{not}B_1, \ldots, \mathbf{not}B_n$, where $H$, $A_i$, and $B_j$ are function-free first-order atoms. Given a rule $r$, we let $hd(r) = \mathbf{K}H$, $bd^+(r) = \{\mathbf{K}A_i \mid i = 1..m\}$, and $bd^-(r) = \{B_i \mid i = 1..n\}$. We use the notation $\mathbf{K}(bd^-(r)) = \{\mathbf{K}B_1, \ldots, \mathbf{K}B_n\}$.

For the interpretation of a hybrid MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$, a transformation $\pi(\mathcal{K}) = \mathbf{K}\pi(\mathcal{O}) \wedge \pi(\mathcal{P})$ is performed to transform $\mathcal{O}$ into a first-order formula and rules $r \in \mathcal{P}$ into a conjunction of first-order implications to make each of them coincide syntactically with an MKNF formula, i.e.,

$$\begin{aligned}
&\pi(r) = \forall \vec{x} : (\mathbf{K}H \subset \mathbf{K}A_1 \wedge \ldots \wedge \mathbf{K}A_m \wedge \mathbf{not}B_1 \wedge \ldots \wedge \mathbf{not}B_n)\\
&\pi(\mathcal{P}) = \textstyle\bigwedge_{r \in \mathcal{P}} \pi(r)\\
&\pi(\mathcal{K}) = \mathbf{K}\pi(\mathcal{O}) \wedge \pi(\mathcal{P})
\end{aligned}$$

where $\vec{x}$ is the vector of free variables in $r$.

Under the additional assumption of DL-safety a first-order rule base is semantically equivalent to a finite ground rule base under both two-valued [5] and three-valued semantics [3]; hence decidability is guaranteed. In the following, we assume that a given rule base is DL-safe and already grounded.

Given a hybrid MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$, let $\mathsf{KA}(\mathcal{K})$ be the set of all (ground) $\mathbf{K}$-atoms $\mathbf{K}\varphi$ such that either $\mathbf{K}\varphi$ occurs in $\mathcal{P}$ or $\mathbf{not}\varphi$ occurs in $\mathcal{P}$. A *partition* of $\mathsf{KA}(\mathcal{K})$ is a pair $(T, P)$ such that $T, P \subseteq \mathsf{KA}(\mathcal{K})$; if $T \subseteq P$, then $(T, P)$ is said to be consistent, otherwise it is inconsistent.

Knorr et al. [3] define an alternating fixpoint construction to compute the well-founded MKNF model. Their construction can be equivalently formulated by stable revisions with the following approximator.

*Definition* Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be a normal hybrid MKNF knowledge base. Define an operator $\Phi_\mathcal{K}$ on $(2^{\mathsf{KA}(\mathcal{K})})^2$ as follows: $\Phi_\mathcal{K}(T, P) = (\Phi_\mathcal{K}(T, P)_1, \Phi_\mathcal{K}(T, P)_2)$, where

$$\begin{aligned}
\Phi_\mathcal{K}(T, P)_1 = \{&\mathbf{K}a \in \mathsf{KA}(\mathcal{K}) \mid \mathsf{OB}_{\mathcal{O},T} \models a\} \cup\\
&\{\mathbf{K}a \mid r \in \mathcal{P} : hd(r) = \{\mathbf{K}a\}, bd^+(r) \subseteq T, \mathbf{K}(bd^-(r)) \cap P = \emptyset\}\\
\Phi_\mathcal{K}(T, P)_2 = \{&\mathbf{K}a \in \mathsf{KA}(\mathcal{K}) \mid \mathsf{OB}_{\mathcal{O},P} \models a\} \cup\\
&\{\mathbf{K}a \mid r \in \mathcal{P} : hd(r) = \{\mathbf{K}a\}, \mathsf{OB}_{\mathcal{O},T} \not\models \neg a, bd^+(r) \subseteq P, \mathbf{K}(bd^-(r)) \cap T = \emptyset\}
\end{aligned}$$

Stable revision computes the least fixpoints of the projection operators $\Phi_\mathcal{K}(\cdot, P)_1$ and $\Phi_\mathcal{K}(T, \cdot)_2$, respectively. Intuitively, given a partition $(T, P)$, the operator $\Phi_\mathcal{K}(\cdot, P)_1$, with $P$ fixed, computes the set of true $\mathbf{K}$-atoms w.r.t. $(T, P)$ and operator $\Phi_\mathcal{K}(T, \cdot)_2$, with $T$ fixed, computes the set of $\mathbf{K}$-atoms that are possibly true w.r.t. $(T, P)$. The condition $\mathsf{OB}_{\mathcal{O},T} \not\models \neg a$ attempts to avoid the generation of a contradiction. We can show that $\Phi_\mathcal{K}$ is an approximator on the bilattice $(2^{\mathsf{KA}(\mathcal{K})})^2$ and preserves all consistent stable fixpoints (and is thus a strong approximator). Note that operator $\Phi_\mathcal{K}$ is not symmetric and it can map a consistent pair to an inconsistent one. A major advantage of AFT is that the stable fixpoints of $\Phi_\mathcal{K}$ satisfying a consistency condition characterize all three-valued MKNF models.

*Theorem* Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be a normal hybrid MKNF knowledge base and $(T, P)$ be a partition. Also let $(M, N) = (\{I \mid I \models \mathsf{OB}_{\mathcal{O},T}\}, \{I \mid I \models \mathsf{OB}_{\mathcal{O},P}\})$. Then, $(M, N)$ is a three-valued MKNF model of $\mathcal{K}$ iff $(T, P)$ is a consistent stable fixpoint of $\Phi_\mathcal{K}$ and $\mathsf{OB}_{\mathcal{O},\mathsf{lfp}(\Phi_\mathcal{K}(\cdot,T)_1)}$ is satisfiable.

Thus, given a hybrid MKNF KB $\mathcal{K}$, we can compute the least stable fixpoint of $\Phi_\mathcal{K}$ and check if $\mathsf{OB}_{\mathcal{O},\mathsf{lfp}(\Phi_\mathcal{K}(\cdot,T)_1)}$ is satisfiable to determine the existence of a well-founded MKNF model. The process is polynomial if so is the underlying DL. Since the problem of computing the well-founded MKNF model is in general intractable, even for normal hybrid MKNF KBs [4], a question is whether we can enlarge the class of hybrid MKNF KBs whose well-founded MKNF model can be tractably computed this way. We answer this question positively by formulating a strictly more precise approximator.

### References


1. Marc Denecker, Victor Marek & Miroslaw Truszczynski (2000): *Approximations, stable operators, wellfounded fixpoints and applications in nonmonotonic reasoning*. In: Logic-Based Artificial Intelligence, Springer, pp. 127–144, doi: 10.1007/978-1-4615-1567-8_6.
2. Marc Denecker, Victor W. Marek & Miroslaw Truszczynski (2004): *Ultimate approximation and its application in nonmonotonic knowledge representation systems*. Information and Computation 192(1), pp. 84–121, doi: 10.1016/j.ic.2004.02.004.
3. Matthias Knorr, Jose Julio Alferes & Pascal Hitzler (2011): *Local closed world reasoning with description logics under the well-founded semantics*. Artificial Intelligence 175(9–10), pp. 1528–1554, doi:10.1016/j.artint.2011.01.007.
4. Fangfang Liu & Jia-Huai You (2017): *Three-valued semantics for hybrid MKNF knowledge bases revisited*. Artificial Intelligence 252, pp. 123–138, doi:10.1016/j.artint.2017.08.003.
5. Boris Motik & Riccardo Rosati (2010): *Reconciling Description Logics and Rules*. Journal of the ACM 57(5), pp. 1–62, doi:10.1145/1754399.1754403.


# Summary of Statistical Statements in Probabilistic Logic Programming


**Damiano Azzolini** (*University of Ferrara*)
**Elena Bellodi** (*University of Ferrara*)
**Fabrizio Riguzzi** (*University of Ferrara*)



### Abstract

Statistical statements, also called Type 1 statements by Halpern, have a new representation in terms of Probabilistic Answer Set Programming under the credal semantics. In this extended abstract we summarize that contribution.


### Contribution

This is a summary of the paper "Statistical Statements in Probabilistic Logic Programming" [1] presented at LPNMR2022. In First-Order knowledge bases, we can adopt statistical statements, also known as Type 1 statements [6], to represent statistical information about the domain, such as "40% of the elements have a particular property". The authors of [7] propose to encode probabilistic conditionals (i.e., statistical statements) with formulas of the form $(C \mid A)[\Pi]$, where $C$ and $A$ are First-Order formulas and $\Pi \in [0, 1]$ with the meaning that $100 \cdot \Pi\%$ of elements that satisfy formula $A$ also satisfy $C$, and compute the probability of a query by adopting the maximum entropy principle [8].

Differently, in [1] the authors propose to encode statistical statements with the syntax

$$(C \mid A)[\Pi_l, \Pi_u]$$

with the meaning that at least $100 \cdot \Pi_l\%$ and at most $100 \cdot \Pi_u\%$ of the elements that satisfy formula $A$ also satisfy $C$. To assign a semantics to these and to perform inference, the authors propose to adopt the credal semantics of probabilistic answer set programs [3] and encode statements with answer set rules and constraints [2]. Probabilistic answer set programming under the credal semantics represents uncertainty with answer set programming extended with ProbLog probabilistic facts [4]

$\Pi :: f$, with the meaning that the probability of $f$ of being true if $\Pi$ ($\Pi \in [0,1]$). A world is a selection of a truth value for every probabilistic fact and is an answer set program with one or more answer sets (the semantics requires that every world has at least one answer set). A world is associated with a probability value computed as

$$P(w) = \prod_{f_i \text{ selected}} \Pi_i \cdot \prod_{f_i \text{ not selected}} (1 - \Pi_i).$$

Since every world has one or more stable models, the probability of a query $P(q)$ is defined by a range: a world contributes to the upper bound if the query is true in at least one answer set and also contributes to the lower bound if it is true in every answer set. It is required that every world has at least one answer set.

The meaning of a statistical statement is given in terms of models for every world. That is, the models of a world where the constraint

$$\Pi_l \leq \frac{\#count\{\mathbf{X} : C(\mathbf{X}), A(\mathbf{X})\}}{\#count\{\mathbf{X} : A(\mathbf{X})\}} \leq \Pi_u$$

is false should be excluded. Practically speaking, consider the following small knowledge base:
*0.4::bird(1). 0.4::bird(2). 0.4::bird(3). 0.4::bird(4).*
*(fly(X)|bird(X))[0.6,1].*
The first line contains four probabilistic facts stating that each of the four elements (indexed with 1, ..., 4) can be birds with probability 0.4. The statistical statement states that at least 60\% (and at most 100\%) of the birds fly. The statement is converted into two answer set rules:
*fly(X) ; not_fly(X) :- bird(X).*
*:- #count{X:bird(X)}=H, #count{X:fly(X),bird(X)}=FH, 100*FH<60*H.*
Only two rules suffice, since $\Pi_u = 1$. From this, we can compute the probability of a query, such as $fly(1)$. This requires checking its validity in the answer sets of each world. Since we are interested only in knowing if it holds in all the answer sets of a world or in at least one, and not in the number of answer sets, we can adopt projection [5] : every probabilistic fact $\Pi :: f$ is converted into three rules $f(PT) : -f.$, $nf(PF) : -not\ f.$, $\{f\}.$, where $PT = \Pi \cdot 10^n$ and $PF = (1 - \Pi) \cdot 10^n$ (where $n$ can be selected and is needed since ASP cannot deal with floating point values), and then we can compute the projective solutions on the query, $f/1$ and, $nf/1$ atoms. For the previous program, we get $P(fly(1)) = [0.2592, 0.4]$.

#### Acknowledgements

This research was partly supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No. 952215.

### References

1. Damiano Azzolini, Elena Bellodi, and Fabrizio Riguzzi (2022): Statistical Statements in Probabilistic Logic Programming. In Georg Gottlob, Daniela Inclezan, and Marco Maratea, editors: Logic Programming and Non-monotonic Reasoning, Springer International Publishing, Cham, pp. 43-55, doi:10.1007/978-3-031-15707-3_4.
2. Gerhard Brewka, Thomas Eiter, and Miroslaw Truszczynski (2011): Answer Set Programming at a Glance. Communications of the ACM 54(12), pp. 92-103, doi:10.1145/2043174.2043195.
3. Fabio Gagliardi Cozman and Denis Deratani Mauá (2020): The joy of Probabilistic Answer Set Programming: Semantics, complexity, expressivity, inference. International Journal of Approximate Reasoning 125, pp. 218-239, doi:10.1016/j.ijar.2020.07.004.
4. Luc De Raedt, Angelika Kimmig, and Hannu Toivonen (2007): ProbLog: A Probabilistic Prolog and Its Application in Link Discovery. In Manuela M. Veloso, editor: 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), 7, AAAI Press, pp. 2462-2467.
5. Martin Gebser, Benjamin Kaufmann, and Torsten Schaub (2009): Solution Enumeration for Projected Boolean Search Problems. In Willem-Jan van Hoeve and John Hooker, editors: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, Springer Berlin Heidelberg, pp. 71-86, doi:10.1007/978-3-642-01929-6_7.
6. Joseph Y. Halpern (1990): An Analysis of First-Order Logics of Probability. Artificial Intelligence 46(3), pp. 311-350, doi:10.1016/0004-3702(90)90019-V.
7. Gabriele Kern-Isberner and Matthias Thimm (2010): Novel Semantical Approaches to Relational Probabilistic Conditionals. In: Proceedings of the Twelfth International Conference on Principles of Knowledge Representation and Reasoning, AAAI Press, pp. 382-392.
8. Marco Wilhelm, Gabriele Kern-Isberner, Marc Finthammer, and Christoph Beierle (2019): Integrating Typed Model Counting into First-Order Maximum Entropy Computations and the Connection to Markov Logic Networks. In Roman Barták and Keith W. Brawner, editors: Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference, AAAI Press, pp. 494-499

---

# Error-free Smart Legal Contracts without Programmers

**Kevin Purnell** (*Macquarie University*)
**Rolf Schwitter** (*Macquarie University*)

### 1 Introduction

This research investigates the potential of declarative languages to advance what can be achieved by domain experts producing their own automation. We illustrate this potential by exploring the issue of domain experts implementing their own executable legal documents as both human- and machine-readable smart legal contracts deployable on a blockchain [7,11]. Smart contracts have the potential to enable automation in previously automation-resistant areas (e.g., legal contracts and contract management). While some legal documents can probably never be fully automated (e.g., subjective performance criteria), many legal documents and contracts use forms of logic that are computable, can be agreed by electronic signature, have performance that can be delivered and administered electronically and use money as the consideration. The current reality; however, is that this type of automation is usually impractical because of: an inability to build custom smart contracts economically; the risk of errors and security breaches; and, inflexibility in regard to implementing contract modifications. These hurdles motivate the research question that we answer: Is it possible for **domain experts** to **design, build, test** and deploy their own **custom legal documents** as **smart contracts** which are **free from errors** and **security exploits** and can also support **contract modifications**? After pure first-order logic was shown to be inadequate for expressing legal concepts [4], non-monotonic declarative languages like Answer Set Programming (ASP), TOAST, SPINdle [1], Turnip [2], and a controlled natural language, Logical English [6] were developed and used to model legal logic. Following this stream of research, we investigate the use of ASP for modelling legal logic by iteratively developing a proof-of-concept smart document editor (SDE editor) which reads an existing legal document (in HTML) and transforms it in five domain expert-guided steps into an error-free executable logic program [8,9,10]. Domain experts model the ontology, legal logic, generate the assertional data, and perform all testing tasks for amenable legal documents. The editor communicates visually and verbally in domain familiar concepts; a feature made possible by the formal correspondences achieved between graphics, text and declarative representations, and automation of steps not requiring domain knowledge. Our approach allows automatic construction of ASP code and enables highly automated testing.

### 2 System Description

The SDE editor uses ASP to represent the ontology and the legal processes derived from a legal document. Firstly, the artefacts in the document and their relationships are captured in representations by a process where three different notations (a term, a graphic and an ASP-like string) are allocated the same meaning by the user. This formal correspondence and the use of a grammar facilitates the automatic generation of representations, rules and the automation of model validation and program verification. The user sees the ontology being built up as they work through the legal document, and can use ontology elements in rules which are then tested in