

# Zugzwang

## *Logic and Artificial Intelligence*

Francisco Coelho          Bruno Dinis  
fc@uevora.pt      bruno.dinis@uevora.pt

January 19, 2023

### Abstract

( **rewrite** ) A major limitation of logical representations is the implicit assumption that the Background Knowledge (BK) is perfect. This assumption is problematic if data is noisy, which is often the case. Here we aim to explore how ASP specifications with probabilistic facts can lead to characterizations of probability functions on the specification's domain.

## 1 Introduction and Motivation

( **rewrite** ) Answer Set Programming (ASP) is a logic programming paradigm based on the Stable Model semantics of Normal Logic Programs (NP) that can be implemented using the latest advances in SAT solving technology. ASP is a truly declarative language that supports language constructs such as disjunction in the head of a clause, choice rules, and hard and weak constraints.

The Distribution Semantics (DS) is a key approach to extend logical representations with probabilistic reasoning. Probabilistic Facts (PF) are the most basic stochastic DS primitive and they take the form of logical facts,  $a$ , labelled with a probability, such as  $p :: a$ ; Each probabilistic fact represents a boolean random variable that is true with probability  $p$  and false with probability  $1 - p$ . A (consistent) combination of the PFs defines a *total choice*  $c = \{p :: a, \dots\}$  such that

$$P(C = x) = \prod_{a \in c} p \prod_{a \notin c} (1 - p). \quad (1)$$

Our goal is to extend this probability, from total choices, to cover the specification domain. We can foresee two key applications of this extended probability:

1. Support any probabilistic reasoning/task on the specification domain.
2. Also, given a dataset and a divergence measure, now the specification can be scored (by the divergence w.r.t. the *empiric* distribution of the dataset), and sorted amongst other specifications. This is a key ingredient in algorithms searching, for example, an optimal specification of the dataset.

This goal faces a critical problem concerning situations where multiple standard models result from a given total choice, illustrated by the following example. The specification

$$\begin{aligned} 0.3 &:: a, \\ &b \vee c \leftarrow a. \end{aligned} \tag{2}$$

has three stable models,  $\bar{a}, ab$  and  $ac$ . While it is straightforward to set  $P(\bar{a}) = 0.7$ , there is *no further information* to assign values to  $P(ab)$  and  $P(ac)$ . At best, we can use a parameter  $x$  such that

$$\begin{aligned} P(ab) &= 0.3x, \\ P(ac) &= 0.3(1 - x). \end{aligned}$$

This uncertainty is inherent to the specification, but can be mitigated with the help of a dataset: the parameter  $x$  can be estimated from the empirical distribution.

In summary, if an ASP specification is intended to describe some observable system then:

1. The observations can be used to estimate the value of the parameters (such as  $x$  above and others entailed from further clauses).
2. With a probability set for the stable models, we want to extend it to all the events of the specification.
3. This extended probability can then be related to the *empirical distribution*, using a probability divergence, such as Kullback-Leibler; and the divergence value used as a *performance* measure of the specification with respect to the observations.

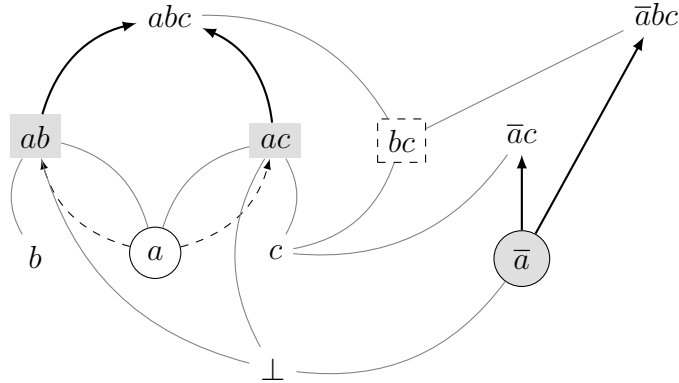


Figure 1: Extending values, *e.g.* probabilities, from total choice nodes to stable models and then to general events in a node-wise process quickly leads to coherence problems concerning probability, with no clear systematic approach.

4. If that specification is only but one of many possible candidates then that performance measure can be used, *e.g.* as fitness, by algorithms searching (optimal) specifications of a dataset of observations.

Currently, we are on the step two above: Extending a probability function (with parameters such as  $x$ ), defined on the stable sets of a specification, to all the events of the specification. This must, of course, respect the axioms of probability so that probabilistic reasoning is consistent with the ASP specification.

## 2 Extending Probabilities

Given an ASP specification, we consider the *atoms*  $a \in \mathcal{A}$  and *literals*,  $z \in \mathcal{L}$ , *events*  $e \in \mathcal{E} \iff e \subseteq \mathcal{L}$  and *worlds*  $w \in \mathcal{W}$  (consistent events), *total choices*  $c \in \mathcal{C} \iff c = a \vee \neg a$  and *stable models*  $s \in \mathcal{S}$ .

Our path starts with a perspective of stable models as playing a role similar to *prime* factors. The stable models of specification are the irreducible events entailed from that specification and any event must be interpreted under its relation with the stable models. This stance leads to definition 1:

**Definition 1.** Let  $e, u, v \in \mathcal{E}$ , and  $\mathcal{S}, \mathcal{W}$  the set of stable models, resp. consistent events, of some specification. Define

$$\langle e | = \{s \in \mathcal{S} \mid e \subseteq s\}, \quad (3)$$

$$|e\rangle = \{s \in \mathcal{S} \mid s \subseteq e\} \quad (4)$$

and

$$u \sim v \iff u, v \notin \mathcal{W} \vee (\langle u| = \langle v| \wedge |u\rangle = |v\rangle). \quad (5)$$

This equivalence relation defines a partition of the events space, where each class holds a unique relation with the stable models. In particular, we can denote each class by

$$e/\sim = \begin{cases} \otimes & \text{if } e \in \mathcal{E} \setminus \mathcal{W}, \\ \langle e|e\rangle & \text{otherwise.} \end{cases} \quad (6)$$

Consider the example from 2. The stable models are  $\mathcal{S} = \bar{a}, ab, ac$  so the quotient set of this relation is

$$\begin{aligned} & \otimes, \langle | \rangle, \\ & \langle \bar{a}|\bar{a} \rangle = \{\bar{a}\}, \langle ab|ab \rangle = \{ab\}, \langle ac|ac \rangle = \{ac\} \\ & \langle \bar{a}| \rangle, \langle ab| \rangle, \langle ac| \rangle, \langle |\bar{a} \rangle, \langle |ab \rangle, \langle |ac \rangle, \\ & \langle \bar{a}, ab| \rangle, \langle \bar{a}, ac| \rangle, \langle ab, ac| \rangle, \langle |\bar{a}, ab \rangle, \langle |\bar{a}, ac \rangle, \langle |ab, ac \rangle, \\ & \langle \bar{a}, ab, ac| \rangle, \langle |\bar{a}, ab, ac \rangle. \end{aligned} \quad (7)$$

For example,  $a/\sim = \langle ab, ac| \rangle$ ,  $abc/\sim = \langle |ab, ac \rangle$  and  $bc/\sim = \langle | \rangle$ .

- Since all events within a equivalence class have the same relation with the stable models, probability assignment should be constant for the elements of that class.
- So, instead of dealing with  $2^6$  events, we need only to handle 19 classes, well defined in terms of combinations of the stable models.
- The extended probability *events* are the *classes*.
- The physical system might have *latent* variables, possibly also represented in the specification. These variables are never observed, so observations should be concentrated in the  $\langle e|/\emptyset$  classes.

( **must adapt** ) Our path, traced by equations (1) and (8 — 13), starts with the probability of total choices,  $P(C = c)$ , expands it to stable models,  $P(S = s)$ , and then to worlds  $P(W = w)$  and events  $P(E = e)$ .

1. **Total Choices.** This case is given by  $P(C = c)$ , from equation 1. Each total choice  $C = c$  (together with the facts and rules) entails some stable models,  $s \in S_c$ , and each stable model  $S = s$  contains a single total choice  $c_s \subseteq s$ .

2. **Stable Models.** Given a stable model  $s \in \mathcal{S}$ , and variables/values  $x_{s,c} \in [0, 1]$ ,

$$P(S = s \mid C = c) = \begin{cases} x_{s,c} & \text{if } s \in \mathcal{S}_c, \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

such that  $\sum_{s \in \mathcal{S}_c} x_{s,c} = 1$ .

3. **Worlds.** Each world  $W = w$  either:

(a) Is a *stable model*. Then

$$P(W = w \mid C = c) = P(S = s \mid C = c). \quad (9)$$

(b) *Contains* some stable models. Then

$$P(W = w \mid C = c) = \prod_{s \subset w} P(S = s \mid C = c). \quad (10)$$

(c) *Is contained* in some stable models. Then

$$P(W = w \mid C = c) = \sum_{s \supset w} P(S = s \mid C = c). \quad (11)$$

(d) Neither contains nor is contained by a stable model. Then

$$P(W = w) = 0. \quad (12)$$

4. **Events.** For each event  $E = e$ ,

$$P(E = e \mid C = c) = \begin{cases} P(W = e \mid C = c) & e \in \mathcal{W}, \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

Since stable model are minimal, there is no proper chain  $s_1 \subset w \subset s_2$  so each world folds into exactly one of the four cases of point 3 above.

Equation (8) expresses the lack of knowledge about the probability assignment when a single total choice entails more than one stable model. In this case, how to distribute the respective probability? Our answer to this problem consists in assigning an unknown probability,  $x_{s,c}$ , conditional on the total choice,  $c$ , to each stable model  $s$ . This approach allow the expression of an unknown quantity and future estimation, given observed data.

The stable model case, in equation (9), identifies the probability of a stable model *as a world* with its probability as defined previously in equation (8), as a stable model.

Equation 10 results from conditional independence of the stable models  $s \subset w$ . Conditional independence of stable worlds asserts a least informed strategy that we make explicit:

**Assumption 1.** *Stable models are conditionally independent, given their total choices.*

Consider the stable models  $ab, ac$  from the example above. They result from the clause  $b \vee c \leftarrow a$  and the total choice  $a$ . These formulas alone impose no relation between  $b$  and  $c$  (given  $a$ ), so none should be assumed. Dependence relations are further discussed in Subsection (2.1).

I'm not sure about what to say here.

todo

My first guess was

$$P(W = w \mid C = c) = \sum_{s \supset w} P(S = s \mid C = c).$$

$P(W = w \mid C = c)$  already separates  $P(W)$  into **disjoint** events!

Also, I am assuming that stable models are independent.

This would entail  $p(w) = p(s_1) + p(s_2) - p(s_1)p(s_2)$  if I'm bound to set inclusion. But I'm not. I'm defining a relation

Also, if I set  $p(w) = p(s_1) + p(s_2)$  and respect the laws of probability, this entails  $p(s_1)p(s_2) = 0$ .

So, maybe what I want is (1) to define the cover  $\hat{w} = \cup_{s \supset w} s$

$$P(W = w \mid C = c) = \sum_{s \supset w} P(S = s \mid C = c) - P(W = \hat{w} \mid C = c).$$

But this doesn't work, because we'd get  $P(W = a \mid C = a) < 1$ .

---

A world that neither contains nor is contained in a stable model describes a case that, according to the specification, should never be observed. So the respective probability is set to zero, per equation (12).

## 2.1 Dependence

Dependence relations in the underlying system can be explicitly expressed in the specification.

For example,  $b \leftarrow c \wedge d$ , where  $d$  is an atomic choice, explicitly expressing this dependence between  $b$  and  $c$ . One would get, for example, the specification

$$0.3 :: a, b \vee c \leftarrow a, 0.2 :: d, b \leftarrow c \wedge d.$$

with the stable models  $\overline{ad}, \overline{ad}, \overline{adb}, \overline{adc}, \overline{adb}$ .

The interesting case is the subtree of the total choice  $ad$ . Notice that no stable model  $s$  contains  $adc$  because (1)  $adb$  is a stable model and (2) if  $adc \subset s$  then  $b \in s$  so  $adb \subset s$ .

Following equations (9) and (12) this sets

$$\begin{cases} P(W = adc | C = ad) = 0, \\ P(W = adb | C = ad) = 1 \end{cases}$$

which concentrates all probability mass from the total choice  $ad$  in the  $adb$  branch, including the node  $W = adb$ . This leads to the following cases:

$x$	$P(W = x   C = ad)$
$ad$	1
$adb$	1
$adc$	0
$adbc$	1

so, for  $C = ad$ ,

$$\begin{aligned} P(W = b) &= \frac{2}{4} \\ P(W = c) &= \frac{1}{4} \\ P(W = bc) &= \frac{1}{4} \\ &\neq P(W = b)P(W = c) \end{aligned}$$

*i.e.* the events  $W = b$  and  $W = c$  are dependent and that dependence results directly from the segment 0.2 ::  $d, b \leftarrow c \wedge d$  in the specification.

---

Prove the four world cases (done), support the product (done)  
and sum (tbd) options, with the independence assumptions.

Todo

### 3 Developed Example

We continue with the specification from equation 2.

**Step 1: Total Choices.** The total choices, and respective stable models, are

Total Choice ( $c$ )	$P(C = c)$	Stable Models ( $s$ )
$a$	0.3	$ab$ and $ac$ .
$\bar{a} = \neg a$	$\overline{0.3} = 0.7$	$\bar{a}$ .

**Step 2: Stable Models.** Suppose now that

Stable Models ( $s$ )	Total Choice ( $c$ )	$P(S = c   C = c)$
$\bar{a}$	1.0	$\bar{a}$ .
$ab$	0.8	$a$ .
$ac$	$0.2 = \overline{0.8}$	$a$ .

**Step 3: Worlds.** Following equations 9 — 12 we get:

Occ. ( $o$ )	S.M. ( $s$ )	Relation	T.C. ( $c$ )	$P(W = w)$
$\emptyset$	all	contained	$a, \bar{a}$	1.0
$a$	$ab, ac$	contained	$a$	$0.8 \times 0.3 + 0.2 \times 0.3 = 0.3$
$b$	$ab$	contained	$a$	$0.8 \times 0.3 = 0.24$
$c$	$ac$	contained	$a$	$0.2 \times 0.3 = 0.06$
$\bar{a}$	$\bar{a}$	stable model	$\bar{a}$	$1.0 \times 0.3 = 0.3$
$\bar{b}$	none	independent	none	0.0
$\bar{c}$	none	...		
$ab$	$ab$	stable model	$a$	0.24
$ac$	$ac$	stable model	$a$	0.06
$\bar{a}\bar{b}$	none	...		
$\bar{a}\bar{c}$	none	...		
$\bar{a}b$	$\bar{a}$	contains	$\bar{a}$	1.0
$\bar{a}c$	$\bar{a}$	...		
$\bar{a}\bar{b}$	$\bar{a}$	...		
$\bar{a}\bar{c}$	$\bar{a}$	...		
$abc$	$ab, ac$	contains	$a$	$0.8 \times 0.2 = 0.016$

## References

## References

- [1] Fabio Gagliardi Cozman and Denis Deratani Mauá. “The joy of probabilistic answer set programming: semantics, complexity, expressivity, inference”. In: *International Journal of Approximate Reasoning* 125 (2020), pp. 218–239.
- [2] Andrew Cropper et al. “Inductive logic programming at 30”. In: *Machine Learning* 111.1 (2022), pp. 147–172.
- [3] Martin Gebser et al. “Answer set solving in practice”. In: *Synthesis lectures on artificial intelligence and machine learning* 6.3 (2012), pp. 1–238.
- [4] Fabrizio Riguzzi. *Foundations of probabilistic logic programming: Languages, semantics, inference and learning*. CRC Press, 2022.
- [5] Victor Verreet et al. “Inference and learning with model uncertainty in probabilistic logic programs”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 9. 2022, pp. 10060–10069.