

Wiki or Bust
- in support of Bob Kowalski's

“Logic Programming in Wikipedia Update”

Luís Moniz Pereira
Universidade Nova de Lisboa
15 June 2007

I fully agree with Bob's “Logic Programming in Wikipedia Update” in this issue.

He writes ‘The stable model semantics article has been tagged since November 2006 as in need of expert attention. In February of this year, it was suggested that it be merged with the article on answer set programming. However, in the discussion about the proposed merger, an anonymous user claimed that “From stable model semantics you get both ASP (credulous reasoning) and Well-founded Semantics as implemented e.g. by XSB using the SLG-WAM which are two competing ways to represent knowledge. ASP is not synonymous with stable models.”¹ This claim, stated with such authority, stopped the proposal in its tracks.’

However, it is clearly wrong: the skeptical version of the stable model semantics is defined as the intersection of all the stable models, if there are any. First, there are cases where none exists, such as when there are odd loops over default negation, e.g. $p :- \text{not } p$, as well as cases of infinite descending chains identified by François Fages (that need not concern us here). Second, floating conclusions belong to the intersection of stable models, but not to the well-founded model, such as ‘c’ in the program $\{c :- a. c :- b. a :- \text{not } b. b :- \text{not } a\}$. So one does not get the well-founded semantics from the stable models semantics, nor does one get the well-founded semantics with strong (or explicit) negation from answer-sets semantics.

Of course, ASP is not synonymous with the stable model semantics, nor does XSB coincide with the well-founded semantics: They are systems, with various limitations, but also with add-ons. However, they invoke those semantics for justification of their results. Apart from that, the semantics of ASP is stable models syntactically extended with strong negation, i.e. Answer-Sets, from whence it gets its name.

Naturally, ASP first computes the well-founded model (at least the well-known Smodels and DLV systems do), which approximates the stable models semantics since the former is contained in all stable models (but not coinciding with their intersection, as we have seen). It does this prior to computing the stable models themselves, if there are any, and is limited to the ground case.

However, answer-set semantics do not produce the well-founded model of programs with strong (or explicit) negation. Indeed, in the three-valued well-founded semantics with explicit negation, strong negation is defined explicitly as entailing default negation – the so-called “coherency” principle. That this entailment is a crucial condition is well-known.

¹ http://en.wikipedia.org/wiki/Talk:Answer_set_programming

Now, answers-sets based semantics, like ASP, need not explicitly state that strong negation entails default negation, for it follows as a theorem in the two-valued semantics. Hence, the three-valued well-founded model computed by ASP, and defined by answer-sets semantics does not provide this entailment, and does not coincide with the corresponding well-founded model. To see this consider program P, where ‘-a’ is the strong negation of ‘a’:

$$- a \quad a :- \text{not } b \quad b :- \text{not } a$$

In the well-founded model, if ‘-a’ did not entail ‘not a’, then ‘a’ would be false by the first rule, but ‘not a’ would remain undefined, a non-intuitive undesirable result. So the coherency principle must be explicitly imposed. In answer-set semantics, however, because it is two-valued, ‘not a’ must be either true or false, and the false case is ruled out because inconsistent with the falsity of ‘a’ imposed by the first rule. But answer set semantics, in its definition, does not impose coherency, i.e. $L \Rightarrow \text{not } L$ for every objective literal L, and so leaves ‘not a’ and ‘not b’ undefined in its “well-founded model with strong negation” approximation. All it does is throw away models that are inconsistent.

However, ASP systems like Smodels, that computes and provides on demand the well-founded model, do in fact produce the well-founded model with explicit negation, making sure coherency is complied with, a need justified by published research on the well-founded semantics with explicit negation (ECAI’94), by José Alferes and myself.

In conclusion, to understand fully and properly ASP one needs to understand answer-set semantics, and to understand the answer-set semantics one needs to understand stable model semantics, and even the well-founded semantics.

Furthermore, to understand the problem representation issues and limitations of ASP, one needs to understand the well-founded semantics and XSB-Prolog. All the more so because there are systems nowadays (such as XASP) supporting applications that combine them to get the best of both worlds. This will surely increase in the future, and the joining together in the same Wikipedia article of all such concerns will only help to make the most of our community’s effort investment, and to remove artificial barriers. But this is neither the place nor the occasion to elaborate at length about these promising developments.

Science should come before marketing and the building of artificial walls. Hence, the stable models and the ASP articles should be joined and, moreover, should aim to connect substantially to a well-founded semantics article, for a deeper, more thorough, and productive understanding.

Now for an organizational comment: ALP could setup a subscribed wiki for members. One problem would presumably be to move it later from to the Wikipedia, as not everyone may agree to be tagged as an author. This can be skirted around by letting people remain anonymous at the time of the move.

Another problem is that this new wiki (there are plenty off-the-shelf systems) might not be as rich for editing and keeping track of editing as the Wikipedia, but in any event are probably good enough.

A final problem is that this will delay affecting the Wikipedia as there will always be people dissatisfied with any on-going state of the private wiki. Also, we may want to continue editing the private wiki, but by then, after the first move, the Wikipedia will have gone its own way and future compatibility will surely be jeopardized.

All in all, I believe our community should raise to my previous general call to arms concerning Logic Programming in the Wikipedia and, as Bob Kowalski says, use an ALP mailing list for exchanging views, irrespective of any Wikipedia editing, but not foreign to it either.

I believe what is needed at this stage is a critical mass of editors from our community to get the ball rolling, and soon more will follow. Perhaps the suggestion that ALP should take up the cause is the right course to follow. We need to identify ALP members who are willing to form the core group and to proselytize for its expansion. This issue could be raised too at the next ICLP, in September.

What we need is a public commitment of some of us to take up arms together, for building a Wikipedia trove on Logic Programming.