# Lecture III

June 7, 2016

## Structure of this Lecture

- ASP and FO(.) compared: the informal semantics
- ASP and FO(.) compared: examples
- Lazy model expansion: interleaving grounding with search

- What do LP's and ASP's language constructs mean?
  - negation as failure
  - rule operator
  - disjunction in the head
  - explicit negation
- A problem started in 1975
- These are problems of informal semantics
  - Can we find a precise informal explanation of programs, that explains the conclusions made by solvers
- Unresolved, I believe

## ±1990: ASP is Autoepistemic/Default reasoning

Example: Grant policy

- Every student for whom we do not know whether he is eligible for grant will be interviewed.

$\downarrow$

$$Interview(x) \leftarrow \texttt{not}\ Eligible(x), \texttt{not}\ \neg Eligible(x).$$

| Informal semantics (G&L) |
| --- |
| $\texttt{not}$ = "I don't know that" |
| $\leftarrow$ = material implication |

$\downarrow$

Every $x$ for which I don't know that $x$ is eligible and I don't know that $x$ is not eligible is to be interviewed.

This view explained:

- The program is the theory of a rational introspective agent
  - The theory is all he knows
- A stable model represents a possible belief state
  - More precisely: the set of literals believed in this belief state
- Backed up by mappings to autoepistemic logic (Moore 85) and Default Logic (Reiter 80)

$$P \leftarrow Q, \texttt{not}\ R$$

$$\downarrow$$

$$Q \wedge \neg K(R) \Rightarrow P \qquad \frac{Q:\neg R}{P} \qquad .$$

it is consistent (according to my theory) that $\neg R$ is true
it is consistent to assume that $\neg R$ is true
I do not know $R$

## ±2000 - now: Search problems

Using ASP for encoding search problems

$\Rightarrow$ new language constructs

$\Rightarrow$ new methodology: GDT-programs
(Generate-Define-Test methodology — [Lifschitz 2002])

## Hamiltonian cycle in ASP

| | |
|---|---|
| GENERATE | $\{Path(x,y)\} \leftarrow Edge(x,y).$ |
| DEFINE | $Node(V). \ \ldots \ Node(W).$ |
| | $Edge(A_1, A_2). \ \ldots \ Edge(A_{n-1}, A_n).$ |
| | $T(x,y) \leftarrow Path(x,y).$<br>$T(x,y) \leftarrow T(x,z), T(z,y).$ |
| TEST | $\leftarrow Path(x,y), Path(x,z), y \neq z.$ |
| | $\leftarrow Path(x,z), Path(y,z), x \neq y.$ |
| | $\leftarrow Node(x), Node(y), \texttt{not}\ T(x,y).$ |

## Hamiltonian cycle in FO(.)

| | |
|---|---|
| GENERATE | $\forall x \forall y (Path(x,y) \Rightarrow Edge(x,y)).$ |
| DEFINE | $\{Node(V).\ \ldots\ Node(W).\}$ |
| | $\{Edge(A_1, A_2).\ \ldots\ Edge(A_{n-1}, A_n).\}$ |
| | $\left\{ \begin{array}{l} \forall x \forall y\ T(x,y) \leftarrow Path(x,y). \\ \forall x \forall y\ T(x,y) \leftarrow T(x,z) \wedge T(z,y). \end{array} \right\}$ |
| TEST | $\forall x \forall y \forall z\ Path(x,y) \wedge Path(x,z) \Rightarrow y = z.$ |
| | $\forall x \forall y \forall z\ Path(x,z) \wedge Path(y,z) \Rightarrow x = y.$ |
| | $\forall x \forall y\ Node(x) \wedge Node(y) \Rightarrow T(x,y).$ |

- The epistemic informal semantics was never satisfactory extended to full ASP?
  - Constraints, choice rules
  - $\leftarrow Node(x), Node(y), \texttt{not}\ T(x,y).$
    "There is no x and y such that I believe that x and y are vertices and I do not know $(x, y) \in T$?
- Answer sets semantics de facto was interpreted as a possible world semantics
  - An answer set represents a possible state of affairs (as everywhere else in logic and science)
- Terminology was not adapted
- The influence on the meaning of `not` and $\leftarrow$ not investigated?

## Hamiltonian cycle in ASP

| | |
|---|---|
| GENERATE | $\{Path(x,y)\} \leftarrow Edge(x,y).$ |
| DEFINE | $Node(V). \ \ldots \ Node(W).$ |
| | $Edge(A_1, A_2). \ \ldots \ Edge(A_{n-1}, A_n).$ |
| | $T(x,y) \leftarrow Path(x,y).$ <br> $T(x,y) \leftarrow T(x,z), T(z,y).$ |
| TEST | $\leftarrow Path(x,y), Path(x,z), y \neq z.$ |
| | $\leftarrow Path(x,z), Path(y,z), x \neq y.$ |
| | $\leftarrow Node(x), Node(y), \texttt{not}\ T(x,y).$ |

## Hamiltonian cycle in FO(.)

| | |
|---|---|
| GENERATE | $\forall x \forall y (Path(x,y) \Rightarrow Edge(x,y)).$ |
| DEFINE | $\{Node(V).\ \ldots\ Node(W).\}$ |
| | $\{Edge(A_1,A_2).\ \ldots\ Edge(A_{n-1},A_n).\}$ |
| | $\left\{ \begin{array}{l} \forall x \forall y\ T(x,y) \leftarrow Path(x,y). \\ \forall x \forall y\ T(x,y) \leftarrow T(x,z) \wedge T(z,y). \end{array} \right\}$ |
| TEST | $\forall x \forall y \forall z\ Path(x,y) \wedge Path(x,z) \Rightarrow y = z.$ |
| | $\forall x \forall y \forall z\ Path(x,z) \wedge Path(y,z) \Rightarrow x = y.$ |
| | $\forall x \forall y\ Node(x) \wedge Node(y) \Rightarrow T(x,y).$ |

## Informal semantics of GDT-programs

$$\{Path(x, y)\} \leftarrow Edge(x, y).$$
$$\text{versus}$$
$$\forall x \forall y (Path(x, y) \Rightarrow Edge(x, y).$$

Global versus local closure

- ASP: global closure of all predicates
  - *Path* is closed unless opened by ... choice rule or whatever
- FO(.): definitions act as local closure of the defined predicates.
  - *Path* is open and the only predicate without a definition. So, all other predicates are closed.

## Informal semantics of GDT-programs

$$\text{DEFINE} \quad \begin{array}{l} Node(V_1).\ \ldots \\ \hline Edge(A_1, A_2).\ldots\ldots. \\ \hline T(x,y) \leftarrow Path(x,y). \\ T(x,y) \leftarrow T(x,z), T(z,y). \end{array}$$

What information is encoded here?

- Definitions
- Definitions include negative information

## Informal semantics of GDT-programs

$$\begin{array}{ll}\textrm{TEST} & \ldots \\ & \leftarrow \mathit{Node}(x), \mathit{Node}(y), \texttt{not } T(x,y).\end{array}$$

What information is encoded here?

- Every pair of nodes is connected.
- $\forall x \forall y (\mathit{Node}(x) \wedge \mathit{Node}(y) \Rightarrow T(x,y))$.

What does this mean according to ASP's informal semantics?

- x, y known to be nodes, I know they are connected.

Marc Denecker, Yuliya Lierler, Miroslaw Truszczynski, Joost Vennekens: A Tarskian Informal Semantics for Answer Set Programming. ICLP (Technical Communications) 2012: 277-289

## There is definitely a difference

$$\Pi = \{P \leftarrow \texttt{not}\ Q\}$$

- ASP, originally: P if I do not know Q (and this is all I know)
- LP as defs: P is defined to be not Q and Q is false (implicitly) (and this is all I know)
- FO(.): P is defined to be not Q (and this is all I know)

Three different meanings.

- Possible world semantics immediately reveals the difference.

- Currently, negation as failure `not` in ASP is still called called default negation
- I suspect, not because it has a different informal meaning than classical negation
- But because this is the "default" value of every atom:
  - an atom is false unless it has support
  - there is no support for deriving falsity of the literal.
- But this does not mean that `not` $\varphi$ is $\neg K\varphi$ from AEL or DL.

The same behavior of negation can be found also in definitions, where it is definitely not epistemic negation

We define $\mathfrak{A} \models \varphi$ by induction on the structure ...:

- ...
- $\mathfrak{A} \models \neg\alpha$ if $\mathfrak{A} \not\models \alpha$
  (i.e., it is not the case that $\mathfrak{A} \models \alpha$).

$$\Downarrow$$

$$\Delta_{\models} = \left\{ \begin{array}{l} \forall i \forall p (Sat(i, p) \leftarrow Atom(p) \wedge In(p, i)) \\ \forall i \forall f \forall g (Sat(i, And(f, g)) \leftarrow Sat(i, f) \wedge Sat(i, g)) \\ \forall i \forall f \forall g (Sat(i, Or(f, g)) \leftarrow Sat(i, f) \vee Sat(i, g)) \\ \forall i \forall f (Sat(i, Not(f)) \leftarrow \neg Sat(i, f)) \end{array} \right\}$$

- I define that a person is dead if he is not alive.
- I define that a person is dead if I do not know that he is alive.

Do a possible world analysis.

Confer the IDP webpage
`https://dtai.cs.kuleuven.be/software/idp/ASPComparison`
This webpage contains a comparison of 4 different applications from the ASP competition 2013.

- N01 - Permutation Pattern Matching
- N06 - Bottle Filling Problem
- N07 - Nomystery
- N12 - Strategic Companies

`https://people.cs.kuleuven.be/~bart.bogaerts/presentations/src/2014/LazyGrounding.html#/`

- Slides of invited talk at LaSh2014, given by Bart Bogaerts
- Based on the paper:

  *Broes de Cat, Marc Denecker, Peter J. Stuckey, Maurice Bruynooghe: Lazy Model Expansion: Interleaving Grounding with Search. J. Artif. Intell. Res. (JAIR) 52: 235-286 (2015)*

The view of knowledge representation:

- formalizing "information"

The view of knowledge representation language:

- a formal language,
  - formal syntax
  - formal semantics
- a theory of informal semantics:
  - a general theory of what information is expressed by formal expressions of the logic

If so, a logic is a formal, exact scientific theory of the informal meaning of the language constructs that it contains.

- E.g., $\land, \lor, \neg, \forall, \exists, \Rightarrow, \Leftrightarrow$ in FO.
- Aggregates
- Definitions

The plan:

- building logics with language constructs derived from the language used in formal science and mathematics.
- Formal possible world semantics:
  - Structures are abstractions of potential states of affairs.
  - A mathematical theory formalizing the value of formal expressions in structures
  - Models = abstractions of possible states of affairs
  - Non-models = abstractions of impossible states of affairs.
- Simulates the methods of formal science (confer Newtons gravitation theory)

A refutable theory

- the informal semantics of an FO(.) theory is a mathematical theory
- We can compare the two and analyse whether they express the same
- As such, every theory is an experiment that can refute or corroborate the theory

We define $\mathfrak{A} \models \varphi$ by induction on the structure . . . :

- . . .
- $\mathfrak{A} \models \neg\alpha$ if $\mathfrak{A} \not\models \alpha$ (i.e., it is not the case that $\mathfrak{A} \models \alpha$).

$$\Downarrow$$

$$\Delta_{\models} = \left\{ \begin{array}{l} \forall i \forall p (Sat(i,p) \leftarrow Atom(p) \wedge In(p,i)) \\ \forall i \forall f \forall g (Sat(i, And(f,g)) \leftarrow Sat(i,f) \wedge Sat(i,g)) \\ \forall i \forall f \forall g (Sat(i, Or(f,g)) \leftarrow Sat(i,f) \vee Sat(i,g)) \\ \forall i \forall f (Sat(i, Not(f)) \leftarrow \neg Sat(i,f)) \end{array} \right\}$$

The experiment:

- Use FO's informal semantics theory to verify mathematically that the formal definition "reads" as the definition
- Compare *Sat* in the well-founded model with the defined relation.