TECHNISCHE UNIVERSITÄT DRESDEN

Computational Logic Group

Hannes Strass (based on slides by Bernardo Cuenca Grau, Ian Horrocks, Przemysław Wałęga)

Faculty of Computer Science, Institute of Artificial Intelligence, Computational Logic Group

Non-Monotonic Reasoning II

Lecture 8, 9th Dec 2024 // Foundations of Knowledge Representation, WS 2024/25

Datalog & Least Herbrand Models

We have seen so far:

- It is easy to formalise intuitions about preferred models if we have a least Herbrand model.
- In that case, everyone agrees that the least Herbrand model is the right choice
- Datalog knowledge bases have a least Herbrand model, which can be computed deterministically using forward chaining
- We can successfully formalise the Closed World Assumption

However, we cannot express default statements:

$$\frac{\mathit{hasOrg}(x,y) \wedge \mathit{Heart}(y) \text{ \& consistent to assume } \mathit{hasLocation}(y,\mathit{left})}{\text{deduce } \mathit{hasLocation}(y,\mathit{left})}$$

Going Beyond Datalog

To overcome expressivity limitations we next

1. extend Datalog to a more expressive logic;
2. develop a new mechanism for selecting preferred models.

Idea: First, allow for "negation" $\sim$ in the body of rules:

$$\forall x \forall y \Big((hasOrg(x, y) \land Heart(y) \land {\sim} hasLocation(y, right)) \to hasLocation(y, left) \Big)$$

Then, devise a preferred model selection mechanism such that negation is read non-monotonically, as follows:

- "Deduce that heart is on the left
  unless we can deduce that it is on the right."
- "Deduce that the heart is on the left if ${\sim} hasLocation(y, right)$
  (that is, $hasLocation(y, left)$ is consistent with our knowledge)."

Datalog$^\neg$-Rules

Definition

A Datalog$^\neg$ rule is a function-free, universally quantified implication

$$(L_1 \wedge \ldots \wedge L_n) \to H$$

with L_i a literal (an atom A or negated atom $\sim A$) and H either an atom or $\bot$.
A Datalog$^\neg$ knowledge base is a pair $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$ where $\mathcal{R}$ is a finite set of Datalog$^\neg$ rules and $\mathcal{F}$ is a finite set of facts.

$$\forall x.(\mathit{Heart}(x) \wedge \mathit{hasLoc}(x, \mathit{left}) \to \mathit{SitSolHeart}(x))$$

$$\forall x.(\mathit{Heart}(x) \wedge \mathit{hasLoc}(x, \mathit{right}) \to \mathit{SitInvHeart}(x))$$

$$\forall x.\forall y.(\mathit{Human}(x) \wedge \mathit{hasOrg}(x, y) \wedge \mathit{SitInvHeart}(y) \to \mathit{SitInvPatient}(x))$$

$$\forall x.\forall y.(\mathit{Human}(x) \wedge \mathit{hasOrg}(x, y) \wedge \mathit{SitSolHeart}(y) \to \mathit{Healthy}(x))$$

$$\forall x.(\forall y.(\mathit{hasOrg}(x, y) \wedge \mathit{Heart}(y) \wedge {\sim}\mathit{hasLoc}(y, \mathit{right}) \to \mathit{hasLoc}(y, \mathit{left})))$$

$$\mathit{Human}(\mathit{MaryJones}), \quad \mathit{hasOrg}(\mathit{MaryJones}, \mathit{MJHeart}), \quad \mathit{Heart}(\mathit{MJHeart})$$

Semantics: Stable Models

So far all this is just syntax.

We need to specify the semantics of Datalog$^\neg$.
$\rightsquigarrow$ Which are the preferred models?

There was a "war of semantics" in 1980s and 1990s.

Meaning of $\{\sim B \to A, \sim A \to B\}$? (Infinite negative recursion.)

Single-model vs. multiple-models semantics?

To date, we have the following:

- Well-founded Semantics
- Stable Model Semantics (a/ka Answer Set Semantics)

Semantics: Stable Models

We will focus on Stable Model Semantics.

Preferred models are given through so-called Stable Models (SM).

It thus follows that

$$\mathcal{K} \models\!\approx \alpha \quad \text{iff} \quad \mathcal{I} \models \alpha \quad \text{for each stable model } \mathcal{I} \text{ of } \mathcal{K}$$

We will see that $\mathcal{K}$ may have

- no stable models, or
- one stable model, or
- several stable models.

Furthermore, if $\mathcal{K}$ contains only Datalog rules (i.e., no negation), then $\mathcal{K}$ has exactly one stable model (the least Herbrand model).

Semantics: Stable Models

We proceed as follows:

1. Define stable models for the propositional case.
2. Extend to the case with variables using grounding.

A simple propositional example $\mathcal{K}$ with one rule and one fact:

$$\begin{aligned} \mathit{Suspect} \land {\sim}\mathit{Guilty} \rightarrow \mathit{Innocent} \\ \mathit{Suspect} \end{aligned}$$

Intuitively, the rule says the following:

"A suspect is innocent unless they can be proved guilty."

We only know that Suspect holds, so we intuitively expect that

$$\mathcal{K} \mathrel{|\!\!\approx} \mathit{Innocent} \qquad \text{and } \mathcal{K} \cup \{\mathit{Guilty}\} \mathrel{|\!\!\not\approx} \mathit{Innocent}$$

Semantics: Stable Models

Our example: $\mathit{Suspect} \wedge \sim\mathit{Guilty} \to \mathit{Innocent}$

$\mathit{Suspect}$

Intuitively, the following (Herbrand-style) model should be stable:

$$\mathcal{I}_1 = \{\mathit{Suspect}, \mathit{Innocent}\}$$

To check this, we first compute the reduct $\mathcal{K}^{\mathcal{I}_1}$ of $\mathcal{K}$ by $\mathcal{I}_1$:

1. Remove all rules with negative body literal $\sim A$ such that the (positive) literal A is in $\mathcal{I}_1$.
2. Remove all negative literals from the remaining rules.

The result is always a (negation-free) Datalog knowledge base.

In our example, we do not remove any rule since $\mathit{Guilty} \notin \mathcal{I}_1$:

$$\mathit{Suspect} \to \mathit{Innocent}$$

$$\mathit{Suspect}$$

Semantics: Stable Models

Once we have the reduct $\mathcal{K}^{\mathcal{J}_1}$

$$\begin{aligned} \mathit{Suspect} \rightarrow \mathit{Innocent} \\ \mathit{Suspect} \end{aligned}$$

We check whether $\mathcal{J}_1$ is the least Herbrand Model of $\mathcal{K}^{\mathcal{J}_1}$, in which case $\mathcal{J}_1$ is a stable model.

Indeed, by using forward chaining we can see that

$$\mathcal{J}_1 \quad = \quad \{\mathit{Suspect}, \mathit{Innocent}\}$$

is the least Herbrand model of $\mathcal{K}^{\mathcal{J}_1}$ and hence $\mathcal{J}_1$ is a stable model of $\mathcal{K}$.

But this is not sufficient to show $\mathcal{K} \mid\!\approx \mathit{Innocent}$.

$\leadsto$ We need to look at all stable models of $\mathcal{K}$.

Semantics: Stable Models

Let us check the remaining possibilities:

$$\begin{aligned}\mathcal{J}_2 &= \{\mathit{Suspect}, \mathit{Guilty}\}\\ \mathcal{J}_3 &= \{\mathit{Suspect}, \mathit{Innocent}, \mathit{Guilty}\}\\ \mathcal{J}_4 &= \{\mathit{Suspect}\}\end{aligned}$$

The reducts $\mathcal{K}^{\mathcal{J}_2}$ and $\mathcal{K}^{\mathcal{J}_3}$ are the same and contain just the fact:

$$\mathit{Suspect}$$

This is so because $\mathit{Guilty} \in \mathcal{J}_2, \mathcal{J}_3$ and hence the reduct does not include the only rule we have in $\mathcal{K}$.

The least model of $\mathcal{K}^{\mathcal{J}_2}$ (or $\mathcal{K}^{\mathcal{J}_3}$) is $\mathcal{J}_4$, thus neither $\mathcal{J}_2$ nor $\mathcal{J}_3$ are stable.

Semantics: Stable Models

We finally check whether

$$\mathcal{I}_4 = \{\mathit{Suspect}\}$$

is a stable model of

$$\begin{aligned} \mathit{Suspect} \land {\sim}\mathit{Guilty} &\rightarrow \mathit{Innocent} \\ & \mathit{Suspect} \end{aligned}$$

The reduct $\mathcal{K}^{\mathcal{I}_4}$ is the same as $\mathcal{K}^{\mathcal{I}_1}$, namely

$$\begin{aligned} \mathit{Suspect} &\rightarrow \mathit{Innocent} \\ & \mathit{Suspect} \end{aligned}$$

But then $\mathcal{I}_4$ is not even a model of $\mathcal{K}^{\mathcal{I}_4}$.

Thus, $\mathcal{I}_1 = \{\mathit{Suspect}, \mathit{Innocent}\}$ is the only stable model of $\mathcal{K}$ and so

$$\mathcal{K} \mathrel{|\!\approx} \mathit{Innocent}.$$

Example (1)

Consider $\mathcal{K}$ as follows:

$$\begin{aligned} {\sim}\mathit{Guilty} &\quad\to\quad \mathit{Innocent} \\ {\sim}\mathit{Innocent} &\quad\to\quad \mathit{Guilty} \end{aligned}$$

Recall that we compute the reduct $\mathcal{K}^{\mathcal{J}}$ of $\mathcal{K}$ by $\mathcal{J}$ as follows:

1. Remove all rules with negative body literal ${\sim}A$ such that the (positive) literal A is in $\mathcal{J}$.
2. Remove all negative literals from the remaining rules.

SM candidates: $\emptyset$, $\{\mathit{Guilty}\}$, $\{\mathit{Innocent}\}$, $\{\mathit{Guilty}, \mathit{Innocent}\}$

Stable models: $\{\mathit{Guilty}\}$, $\{\mathit{Innocent}\}$

$\leadsto$ A KB can have several stable models.

Example (2)

Consider $\mathcal{K}$ as follows:

$$\sim Guilty \quad \rightarrow \quad Guilty$$

Recall that we compute the reduct $\mathcal{K}^{\mathcal{J}}$ of $\mathcal{K}$ by $\mathcal{J}$ as follows:

1. Remove all rules with negative body literal $\sim A$ such that the (positive) literal A is in $\mathcal{J}$.
2. Remove all negative literals from the remaining rules.

Stable model candidates: $\emptyset$, $\{Guilty\}$

$\rightsquigarrow$ A KB may have no stable models.

Non-monotonic vs. Classical Negation

Consider again our propositional example $\mathcal{K}$:

$$\begin{aligned} \mathit{Suspect} \land \neg \mathit{Guilty} \rightarrow \mathit{Innocent} \\ \mathit{Suspect} \end{aligned}$$

Let us check whether

$$\mathcal{K} \models \mathit{Innocent}$$

for $\models$ being entailment under monotonic PL semantics.

Clearly, $\mathcal{K}$ is equivalent in standard propositional logic to

$$\begin{aligned} \mathit{Suspect} \rightarrow \mathit{Innocent} \lor \mathit{Guilty} \\ \mathit{Suspect} \end{aligned}$$

Hence $\mathcal{I} = \{\mathit{Suspect}, \mathit{Guilty}\}$ is a model of $\mathcal{K}$ with $\mathcal{I} \not\models \mathit{Innocent}$, thus:

$$\mathcal{K} \not\models \mathit{Innocent}$$

Properties

Let $\mathcal{K}$ be a (propositional) Datalog$^{\neg}$ knowledge base. Then:

Theorem

Every stable model of $\mathcal{K}$ is a classical model of $\mathcal{K}$.

Corollary

If $\mathcal{K} \models \alpha$, then $\mathcal{K} \mathrel{|\!\!\approx} \alpha$.

Theorem

If a proposition P holds in some stable model of $\mathcal{K}$, then P is a head of some rule (or a fact) in $\mathcal{K}$.

Theorem

If $\mathcal{I}_1$ and $\mathcal{I}_2$ are stable models of $\mathcal{K}$, then neither $\mathcal{I}_1 \subsetneq \mathcal{I}_2$ nor $\mathcal{I}_2 \subsetneq \mathcal{I}_1$.

Stable Models: Non-Propositional Case

So far, all this is propositional. What about …

$$\forall x(\mathit{Heart}(x) \land \mathit{hasLoc}(x, \mathit{left}) \rightarrow \mathit{SitSolHeart}(x))$$

$$\forall x(\mathit{Heart}(x) \land \mathit{hasLoc}(x, \mathit{right}) \rightarrow \mathit{SitInvHeart}(x))$$

$$\forall x \forall y(\mathit{Human}(x) \land \mathit{hasOrg}(x, y) \land \mathit{SitInvHeart}(y) \rightarrow \mathit{SitInvPatient}(x))$$

$$\forall x \forall y(\mathit{Human}(x) \land \mathit{hasOrg}(x, y) \land \mathit{SitSolHeart}(y) \rightarrow \mathit{Healthy}(x))$$

$$\forall x(\forall y(\mathit{hasOrg}(x, y) \land \mathit{Heart}(y) \land {\sim}\mathit{hasLoc}(y, \mathit{right}) \rightarrow \mathit{hasLoc}(y, \mathit{left})))$$

$$\mathit{Human}(\mathit{MJ})$$

$$\mathit{hasOrg}(\mathit{MJ}, h)$$

$$\mathit{Heart}(h)$$

Fortunately, we are still within the Bernays-Schönfinkel class.*

$\rightsquigarrow$ We can apply grounding and reduce to the propositional case.

*: Bernays-Schönfinkel formulas are of the form $\exists x_1 \ldots \exists x_m \forall y_1 \ldots \forall y_n \varphi$ with φ quantifier-free.

Stable Models: Non-Propositional Case

So, to compute all the stable models of $\mathcal{K}$:

1. Compute the grounding of $\mathcal{K}$ over the Herbrand universe.
2. Compute all the stable models of the resulting propositional KB.

Obviously, the grounding could be of exponential size.

But this is a computational hazard, not a conceptual one.

Intuitively, the following Herbrand model should be stable:

$$\begin{aligned}\mathcal{I}_1 = \{&\mathit{Human}(\mathit{MJ}), \mathit{hasOrg}(\mathit{MJ}, h), \mathit{Heart}(h), \mathit{hasLoc}(h, \mathit{left}), \\ &\mathit{SitSolHeart}(h), \mathit{Healthy}(\mathit{MJ})\}\end{aligned}$$

On the other hand, the following one should not be stable:

$$\begin{aligned}\mathcal{I}_2 = \{&\mathit{Human}(\mathit{MJ}), \mathit{hasOrg}(\mathit{MJ}, h), \mathit{Heart}(h), \mathit{hasLoc}(h, \mathit{right}), \\ &\mathit{SitInvHeart}(h), \mathit{SitInvPatient}(\mathit{MJ})\}\end{aligned}$$

Stable Models: Non-Propositional Case

To check whether

$$\mathcal{I}_1 = \{Human(MJ), hasOrg(MJ, h), Heart(h), hasLoc(h, left), SitSolHeart(h), Healthy(MJ)\}$$

is stable, notice that even though the grounding is huge, the only PL formulas that matter are the following:

$$Heart(h) \wedge hasLoc(h, left) \rightarrow SitSolHeart(h)$$
$$Human(MJ) \wedge hasOrg(MJ, h) \wedge SitSolHeart(h) \rightarrow Healthy(MJ)$$
$$hasOrg(MJ, h) \wedge Heart(h) \wedge {\sim}hasLoc(h, right) \rightarrow hasLoc(h, left)$$
$$Human(MJ),\ hasOrg(MJ, h),\ Heart(h)$$

The reduct of $\mathcal{I}_1$ over those formulas is

$$Heart(h) \wedge hasLoc(h, left) \rightarrow SitSolHeart(h)$$
$$Human(MJ) \wedge hasOrg(MJ, h) \wedge SitSolHeart(h) \rightarrow Healthy(MJ)$$
$$hasOrg(MJ, h) \wedge Heart(h) \rightarrow hasLoc(h, left)$$
$$Human(MJ),\ hasOrg(MJ, h),\ Heart(h)$$

And clearly, $\mathcal{I}_1$ is the least model.

Stable Models: Non-Propositional Case

To check whether

$$\mathcal{I}_2 = \{Human(MJ), hasOrg(MJ,h), Heart(h), hasLoc(h,right), SitInvHeart(h), SitInvPatient(MJ)\}$$

is stable, the relevant PL formulas are the following:

$$\begin{aligned}
Heart(h) \land hasLoc(h,right) &\to SitInvHeart(h) \\
Human(MJ) \land hasOrg(MJ,h) \land SitInvHeart(h) &\to SitInvPatient(MJ) \\
hasOrg(MJ,h) \land Heart(h) \land {\sim}hasLoc(h,right) &\to hasLoc(h,left) \\
Human(MJ),\ hasOrg(MJ,h),\ Heart(h) &
\end{aligned}$$

The reduct of $\mathcal{I}_2$ over those formulas is

$$\begin{aligned}
Heart(h) \land hasLoc(h,right) &\to SitInvHeart(h) \\
Human(MJ) \land hasOrg(MJ,h) \land SitInvHeart(h) &\to SitInvPatient(MJ) \\
Human(MJ),\ hasOrg(MJ,h),\ Heart(h) &
\end{aligned}$$

And clearly, $\mathcal{I}_2$ is not the least model.

Quick Recap

We have seen that by using Datalog with non-monotonic negation

1. We can formalise the closed-world assumption
2. We can express default statements

The key notion is that of a Stable Model as a “preferred” model.

Checking whether a propositional model is stable involves

1. Eliminating negation by computing the reduct
2. Checking if the candidate is the least model of the reduct

Checking whether a FOL Herbrand interpretation is a stable model involves

1. Computing the propositional grounding of the KB
2. Checking whether the candidate is stable for the grounding

Note: Stable models in the FOL case are always Herbrand models.

What have we left out?

Much more than we have covered!

The field of NMR is huge and we have just seen the tip of the iceberg.

Extensions related to what we have seen:

- Stable models and disjunctive rules (disjunction in the head), e.g.

$$Professor(x), Semester(s) \rightarrow Teaches(x, s) \vee Sabbatical(x, s)$$

- Stable models and general propositional formulas
- Combinations of classical and non-monotonic negation, e.g.

$$Suspect(x), {\sim}Guilty(x) \rightarrow \neg Guilty(x)$$
$$Heart(x), {\sim}\neg SitSolH(x) \rightarrow SitSolH(x)$$

Relationships with other areas

What we have seen is not only relevant to KR.

There are strong connections with other fields:

- Answer Set Programming (ASP)

    Using negation we can encode search problems

- Deductive databases

    Database systems that can conclude new data using rules

- Logic programming (Prolog)

    Negation as failure can help write shorter programs