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, 5th Dec 2022 // Foundations of Knowledge Representation, WS 2022/23

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:

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

TECHNISCHE UNIVERSITÄT DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 2 of 22

Computational Logic ∴ Group

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 in the body of rules:

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

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"

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 3 of 22

Computational
Logic ∴ Group

Datalog¬-Rules

A Datalog¬ rule is a function-free, universally quantified implication of the form

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

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

$$\forall x.(Heart(x) \land hasLoc(x, left) \to SitSolHeart(x))$$
$$\forall x.(Heart(x) \land hasLoc(x, right) \to SitInvHeart(x))$$
$$\forall x.\forall y.(Human(x) \land hasOrg(x, y) \land SitInvHeart(y) \to SitInvPatient(x))$$
$$\forall x.\forall y.(Human(x) \land hasOrg(x, y) \land SitSolHeart(y) \to Healthy(x))$$
$$\forall x.(\forall y.(hasOrg(x, y) \land Heart(y) \land \sim hasLoc(y, right) \to hasLoc(y, left)))$$
$$Human(MaryJones), \quad hasOrg(MaryJones, MJHeart), \quad Heart(MJHeart)$$

TECHNISCHE UNIVERSITÄT DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 4 of 22

Computational Logic ∴ Group

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 things like $\sim B \rightarrow A$ and $\sim A \rightarrow B$

Single vs. multiple-models semantics

To date, we have the following:

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

TECHNISCHE UNIVERSITÄT DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 5 of 22

Computational Logic ∴ Group

Semantics: Stable Models

We will focus on Stable Model Semantics.

Preferred models are called Stable Models (SM).

It thus follows that

$$\mathcal{K} \not\hspace{-0.3em}\models\hspace{-0.8em}\sim \alpha \quad \text{iff} \quad \mathcal{I} \models \alpha \quad \text{for each stable model } \mathcal{I} \text{ of } \mathcal{K}$$

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).

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 6 of 22

TECHNISCHE UNIVERSITÄT DRESDEN

Computational Logic ∴ Group

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:

$$Suspect \wedge \sim Guilty \rightarrow Innocent$$

$$Suspect$$

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

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

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 7 of 22

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:

$$Suspect \wedge \sim Guilty \rightarrow Innocent$$

$$Suspect$$

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

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 7 of 22

Computational
Logic ∴ Group

Semantics: Stable Models

Our example: $Suspect \wedge \sim Guilty \rightarrow Innocent$

$Suspect$

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

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

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 remaining rules

The result is always a Datalog knowledge base.

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

$$Suspect \rightarrow Innocent$$

$$Suspect$$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 8 of 22

Computational
Logic ∴ Group

Semantics: Stable Models

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

$$Suspect \rightarrow Innocent$$
$$Suspect$$

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

Indeed, by using forward chaining we can see that

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

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

But this is not sufficient to show $\mathcal{K} \mathrel{\not\!\sim} Innocent$.

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 9 of 22

Computational
Logic ∴ Group

Semantics: Stable Models

Let us check the remaining possibilities:

$$\mathcal{I}_2 = \{\textit{Suspect}, \textit{Guilty}\}$$
$$\mathcal{I}_3 = \{\textit{Suspect}, \textit{Innocent}, \textit{Guilty}\}$$
$$\mathcal{I}_4 = \{\textit{Suspect}\}$$

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

$$\textit{Suspect}$$

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

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 10 of 22

Computational
Logic ∴ Group

Semantics: Stable Models

We finally check whether

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

is a stable model of

$$Suspect \wedge {\sim}Guilty \to Innocent$$
$$Suspect$$

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

$$Suspect \to Innocent$$
$$Suspect$$

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

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

$$\mathcal{K} \not\hspace{0.1em}\models Innocent.$$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 11 of 22

Computational
Logic ∴ Group

Example (1)

Consider $\mathcal{K}$ as follows:

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

Recall that we compute the reduct $\mathcal{K}^{\mathcal{I}}$ of $\mathcal{K}$ by $\mathcal{I}$ as follows:
1. Remove all rules with negative body literal $\sim A$ such that the (positive) literal A is in $\mathcal{I}$
2. Remove all negative literals from remaining rules

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

Example (1)

Consider $\mathcal{K}$ as follows:

$$\sim Guilty \quad \to \quad Innocent$$
$$\sim Innocent \quad \to \quad Guilty$$

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

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

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

Reduct $\mathbf{K}^{\emptyset}$:

$$Innocent$$
$$Guilty$$

Least model of $\mathbf{K}^{\emptyset}$: $\{Guilty, Innocent\} \neq \emptyset \rightsquigarrow$ no stable model

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 12 of 22

Computational
Logic ∴ Group

Example (1)

Consider $\mathcal{K}$ as follows:

$$\sim\!Guilty \quad \rightarrow \quad Innocent$$
$$\sim\!Innocent \quad \rightarrow \quad Guilty$$

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

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

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

Reduct $\mathbf{K}^{\{Guilty\}}$:

$$Guilty$$

Least model of $\mathbf{K}^{\{Guilty\}}$: $\{Guilty\} \rightsquigarrow$ stable model

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 12 of 22

Computational
Logic ∴ Group

Example (1)

Consider $\mathcal{K}$ as follows:

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

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

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

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

Reduct $\mathbf{K}^{\{Innocent\}}$:

$$Innocent$$

Least model of $\mathbf{K}^{\{Innocent\}}$: $\{Innocent\} \rightsquigarrow$ stable model

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 12 of 22

TECHNISCHE
UNIVERSITÄT
DRESDEN

Computational
Logic ∴ Group

Example (1)

Consider $\mathcal{K}$ as follows:

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

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

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

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

Reduct $\mathbf{K}^{\{Guilty, Innocent\}}$:

(empty set)

Least model of $\mathbf{K}^{\{Guilty, Innocent\}}$: $\emptyset \neq \{Guilty, Innocent\} \rightsquigarrow$ no stable model

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 12 of 22

Computational
Logic ∴ Group

Example (1)

Consider $\mathcal{K}$ as follows:

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

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

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

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

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

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 12 of 22

TECHNISCHE
UNIVERSITÄT
DRESDEN

Computational
Logic ∴ Group

Example (1)

Consider $\mathcal{K}$ as follows:

$$\sim\!\textit{Guilty} \quad \rightarrow \quad \textit{Innocent}$$
$$\sim\!\textit{Innocent} \quad \rightarrow \quad \textit{Guilty}$$

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

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

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

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

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

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 12 of 22

TECHNISCHE
UNIVERSITÄT
DRESDEN

Computational
Logic ∴ Group

Example (2)

Consider $\mathcal{K}$ as follows:

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

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

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

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

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 13 of 22

Example (2)

Consider $\mathcal{K}$ as follows:

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

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

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

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

Reduct $\mathbf{K}^{\emptyset}$:

$$Guilty$$

Least model of $\mathbf{K}^{\emptyset}$: $\{Guilty\} \neq \emptyset \rightsquigarrow$ no stable model

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 13 of 22

Computational
Logic ∴ Group

Example (2)

Consider $\mathcal{K}$ as follows:

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

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

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

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

Reduct $\mathbf{K}^{\{Guilty\}}$: (empty set)

Least model of $\mathbf{K}^{\{Guilty\}}$: $\emptyset \neq \{Guilty\} \rightsquigarrow$ no stable model

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 13 of 22

Example (2)

Consider $\mathcal{K}$ as follows:

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

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

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

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

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

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 13 of 22

TECHNISCHE
UNIVERSITÄT
DRESDEN

Computational
Logic ∴ Group

Non-monotonic vs. Classical Negation

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

$$Suspect \wedge \neg Guilty \rightarrow Innocent$$
$$Suspect$$

Let us check whether

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

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

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

$$Suspect \rightarrow Innocent \vee Guilty$$
$$Suspect$$

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

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 14 of 22

Computational
Logic ∴ Group

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 $\mathcal{K} \mathrel{\vnottilde} \alpha$.

Theorem

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

Theorem

If $\mathcal{I}_1$ and $\mathcal{I}_2$ are stable models of $\mathcal{K}$, then neither $\mathcal{I}_1 \subsetneq \mathcal{I}_2$ or $\mathcal{I}_2 \subsetneq \mathcal{I}_1$.

TECHNISCHE UNIVERSITÄT DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 15 of 22

Computational Logic Group

Stable Models: Non-Propositional Case

So far, all this is propositional. What about …

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

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.

TECHNISCHE UNIVERSITÄT DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 16 of 22

Computational Logic ∴ Group

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.

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 17 of 22

Computational
Logic ∴ Group

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:

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 17 of 22

Computational
Logic ∴ Group

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:

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

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

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 17 of 22

Computational
Logic ∴ Group

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:

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

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

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

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

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 18 of 22

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:

$$Heart(h) \wedge hasLoc(h, right) \rightarrow SitInvHeart(h)$$
$$Human(MJ) \wedge hasOrg(MJ, h) \wedge SitInvHeart(h) \rightarrow SitInvPatient(MJ)$$
$$hasOrg(MJ, h) \wedge Heart(h) \wedge {\sim}hasLoc(h, right) \rightarrow hasLoc(h, left)$$
$$Human(MJ), \quad hasOrg(MJ, h), \quad Heart(h)$$

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

$$Heart(h) \wedge hasLoc(h, right) \rightarrow SitInvHeart(h)$$
$$Human(MJ) \wedge hasOrg(MJ, h) \wedge SitInvHeart(h) \rightarrow SitInvPatient(MJ)$$
$$Human(MJ), \quad hasOrg(MJ, h), \quad Heart(h)$$

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

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 19 of 22

TECHNISCHE
UNIVERSITÄT
DRESDEN

Computational
Logic ∴ Group

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.

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 20 of 22

Computational
Logic ∴ Group

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.

$$professor(x), semester(s) \rightarrow teaches(x, s) \lor sabbatical(x, s)$$

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

$$suspect(x), \sim guilty(x) \rightarrow \neg guilty(x)$$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 21 of 22

Computational
Logic ∴ Group

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

Non-Monotonic Reasoning II (Lecture 8)
Computational Logic Group // Hannes Strass
Foundations of Knowledge Representation, WS 2022/23

Slide 22 of 22

Computational
Logic ∴ Group