TECHNISCHE UNIVERSITÄT DRESDEN

Computational Logic ∴ Group

Hannes Strass

(based on slides by Michael Thielscher)

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

Negation: Model Theory

Lecture 9, 4th Dec 2023 // Foundations of Logic Programming, WS 2023/24

Previously ...

- For every normal logic program P, its **completion** $comp(P)$ replaces the logical implications of clauses by equivalences (to disjunctions of bodies).
- SLDNF resolution w.r.t. P is **sound** for entailment w.r.t. $comp(P)$.
- SLDNF resolution is only **complete** (for entailment w.r.t. $comp(P)$) for certain combinations of classes of programs, queries, and selection rules.
- For a normal program P, its **dependency graph** D_P explicitly shows positive and negative dependencies between predicate symbols.
- A normal program P is **stratified** iff D_P has no cycle with a negative edge.

$$P: \quad \begin{aligned} p &\leftarrow q, \sim r \\ q &\leftarrow r \end{aligned}$$

$$\begin{tikzcd} p/0 & q/0 \arrow[l, "+"'] & r/0 \arrow[l, "+"'] \arrow[ll, bend left, "-"] \end{tikzcd}$$

Completion of P:

$$\begin{aligned} p &\leftrightarrow (q \land \neg r) \\ q &\leftrightarrow r \\ r &\leftrightarrow \mathit{false} \end{aligned}$$

Overview

Consequence Operator for Normal Programs

Standard Models

Perfect Models and Local Stratification

Well-Supported Models

Consequence Operator for Normal Programs

Consequence Operator for Normal Programs

Definition

Let P be a normal logic program and I be a Herbrand interpretation. Then

$$T_P(I) := \{H \mid H \leftarrow \vec{B} \in ground(P),\ I \models \vec{B}\}$$

In case P is a definite program, we know that

- T_P is monotonic,
- T_P is continuous,
- T_P has the least fixpoint $\mathcal{M}(P)$,
- $\mathcal{M}(P) = T_P \uparrow \omega$.

For normal programs, all of these properties are lost.

T_P-Characterisation for Normal LPs (1)

Lemma 4.3

Let P be a normal logic program and I be a Herbrand interpretation. Then

$$I \models P \quad \text{iff} \quad T_P(I) \subseteq I$$

Proof.

$$\begin{array}{ll}
 & I \models P \\
\text{iff} & \text{for every } H \leftarrow \vec{B} \in \mathit{ground}(P)\text{:} \quad I \models \vec{B} \text{ implies } I \models H \\
\text{iff} & \text{for every } H \leftarrow \vec{B} \in \mathit{ground}(P)\text{:} \quad I \models \vec{B} \text{ implies } H \in I \\
\text{iff} & \text{for every ground atom } H\text{:} \\
 & \quad H \leftarrow \vec{B} \in \mathit{ground}(P) \text{ and } I \models \vec{B} \text{ implies } H \in I \\
\text{iff} & \text{for every ground atom } H\text{:} \quad H \in T_P(I) \text{ implies } H \in I \\
\text{iff} & T_P(I) \subseteq I
\end{array}$$

$\square$

T_P-Characterisation for Normal LPs (2)

Definition

Let F and Π be ranked alphabets of function symbols and predicate symbols, respectively, let $= \notin \Pi$ be a binary predicate symbol (equality), and let I be a Herbrand interpretation for F and Π. Then

$$I_= \; := \; I \cup \{=(t,t) \mid t \in HU_F\}$$

is called a **standardised** Herbrand interpretation for F and $\Pi \cup \{=\}$.

Lemma 4.4

Let P be a normal logic program and I a Herbrand interpretation. Then

$$I_= \models comp(P) \quad \text{iff} \quad T_P(I) = I$$

$\rightsquigarrow$ The T_P operator for normal LPs characterises the completion semantics.

T_P-Characterisation for Normal LPs (3)

Proof Idea of Lemma 4.4:

$I_= \models comp(P)$

iff (since $I_=$ is a model for standard axioms of equality and inequality)
for every ground atom H: $I \models \left(H \leftrightarrow \bigvee_{(H \leftarrow \vec{B}) \in ground(P)} \vec{B} \right)$

iff for every ground atom H: $H \in I$ iff $I \models \bigvee_{(H \leftarrow \vec{B}) \in ground(P)} \vec{B}$

iff for every ground atom H: $H \in I$ iff $I \models \vec{B}$ for some $H \leftarrow \vec{B} \in ground(P)$

iff for every ground atom H: $H \in I$ iff $H \in T_P(I)$

iff $T_P(I) = I$

$\square$

$\rightsquigarrow$ Is *comp*(*P*) the "intended" declarative semantics of *P*?

Completion may be Inadequate

Consider the following normal logic program P:

$$\begin{aligned} &ill \leftarrow {\sim}ill, infection \\ &infection \leftarrow \end{aligned}$$

Its completion $comp(P) \supseteq \{ill \leftrightarrow (\neg ill \wedge infection), \quad infection \leftrightarrow true\}$ is unsatisfiable (it has no models).

Hence, $comp(P) \models healthy$.

But $I = \{infection, ill\}$ is the only Herbrand model of P (taken as a theory):

$$P \equiv \{ill \leftarrow (\neg ill \wedge infection),\ infection\} \equiv \{ill \vee \neg\neg ill \vee \neg infection,\ infection\}$$

Hence, $P \not\models healthy$.

Non-Intended Minimal Herbrand Models

For normal LPs, a unique least Herbrand model is not guaranteed to exist.

Can we at least settle for minimal Herbrand models?

$$P_1: \quad p \leftarrow {\sim}q$$

P_1 has three Herbrand models:

$$M_1 = \{p\},\ M_2 = \{q\}, \text{ and } M_3 = \{p, q\}.$$

P_1 has no least, but two minimal Herbrand models: M_1 and M_2

However: M_1, and not M_2, is the “intended” model of P_1.

Supported Herbrand Interpretations

Definition

A Herbrand interpretation I of P is **supported**
$:\Longleftrightarrow$ for every $H \in I$ there exists some $H \leftarrow \vec{B} \in ground(P)$ such that $I \models \vec{B}$.

If additionally $I \models P$, we say that I is a **supported model** of P.

(Intuitively: $\vec{B}$ is an explanation for H.)

Example

- M_1 is a supported model of P_1. (Literal $\sim q$ is a support for p.)
- M_2 is no supported model of P_1. (Atom $q \in M_2$ has no support.)
- Note (cf. Lemma 4.3) that $T_{P_1}(M_2) = \emptyset \subsetneq M_2$, but in contrast $T_{P_1}(M_1) = M_1$.
- The definite (therefore normal) program $\{p \leftarrow q, \quad q \leftarrow p\}$ has two supported models: $\emptyset$ and $\{p, q\}$. In the second supported model, p is an explanation for q and vice versa. Thus "support" can be cyclic.

T_P-Characterisation for Normal LPs (4)

Lemma 6.2

Let P be a normal program and I be a Herbrand interpretation. Then

$$I \text{ is a supported model of } P \quad \text{iff} \quad T_P(I) = I$$

Proof Idea.

$$
\begin{array}{ll}
 & I \models P \text{ and } I \text{ is supported} \\
\text{iff} & \text{for every } (H \leftarrow \vec{B}) \in \mathit{ground}(P)\text{:} \quad I \models \vec{B} \text{ implies } I \models H \\
 & \quad \text{and for every } H \in I\text{: } I \models \bigvee_{(H \leftarrow \vec{B}) \in \mathit{ground}(P)} \vec{B} \\
\text{iff} & \text{for every ground atom } H\text{: } I \models \left(H \leftarrow \bigvee_{(H \leftarrow \vec{B}) \in \mathit{ground}(P)} \vec{B}\right) \\
 & \quad \text{and } I \models \left(H \rightarrow \bigvee_{(H \leftarrow \vec{B}) \in \mathit{ground}(P)} \vec{B}\right) \\
\text{iff} & \text{for every ground atom } H\text{: } I \models \left(H \leftrightarrow \bigvee_{(H \leftarrow \vec{B}) \in \mathit{ground}(P)} \vec{B}\right) \\
\text{iff} & I_{=} \text{ model for } \mathit{comp}(P) \\
\text{iff} & \text{(Lemma 4.4)} \quad T_P(I) = I
\end{array}
$$

$\square$

Standard Models

Non-Intended Supported Models

$$
\begin{array}{llll}
P_2: & p & \leftarrow & \sim q \\
 & q & \leftarrow & q
\end{array}
$$

P_2 has three Herbrand models:

$$M_1 = \{p\},\ M_2 = \{q\},\ \text{and } M_3 = \{p, q\}$$

P_2 has two supported Herbrand models:

$$M_1 \text{ and } M_2$$

However: M_1, and not M_2, is the “intended” model of P_2.

M_1 will be called the standard model of P_2 (cf. slide 19).

Stratifications

Definition

Let P be a normal program with dependency graph D_P.

- A predicate symbol p is **defined** in P
  $:\Longleftrightarrow \quad P$ contains a clause $p(t_1, \ldots, t_n) \leftarrow \vec{B}$.
- $P_1 \cup \ldots \cup P_n = P$ is a **stratification** of P
  $:\Longleftrightarrow$
  1. $P_i \neq \emptyset$ for every $i \in [1, n]$
  2. $P_i \cap P_j = \emptyset$ for every $i, j \in [1, n]$ with $i \neq j$

  $\Big\}$ a partition of P

  3. for every p defined in P_i and edge $q \xrightarrow{+} p$ in D_P: q is not defined in $\bigcup_{j=i+1}^{n} P_j$
  4. for every p defined in P_i and edge $q \xrightarrow{-} p$ in D_P: q is not defined in $\bigcup_{j=i}^{n} P_j$

Lemma 6.5

A normal program P is stratified iff there exists a stratification of P.

Note: A stratified program may have different stratifications.

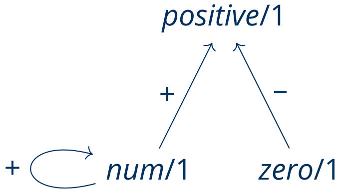
Example (1)

The normal logic program P is the following:

$$
\begin{array}{rcl}
zero(0) & \leftarrow & \\
positive(x) & \leftarrow & num(x),\ \sim zero(x) \\
num(0) & \leftarrow & \\
num(s(x)) & \leftarrow & num(x)
\end{array}
$$

$P_1 \cup P_2 \cup P_3$ is a stratification of P, where

$$
\begin{array}{rcl}
P_1 & = & \{num(0) \leftarrow, \quad num(s(x)) \leftarrow num(x)\} \\
P_2 & = & \{zero(0) \leftarrow\} \\
P_3 & = & \{positive(x) \leftarrow num(x),\ \sim zero(x)\}
\end{array}
$$



Example (2)

$$
\begin{aligned}
num(0) &\leftarrow \\
num(s(x)) &\leftarrow num(x) \\
even(0) &\leftarrow \\
even(x) &\leftarrow {\sim}odd(x),\ num(x) \\
odd(s(x)) &\leftarrow even(x)
\end{aligned}
$$

***P* admits no stratification.**

$$
\begin{tikzcd}
num/1 \arrow[loop left, "+"] \arrow[r, "+"] & even/1 \arrow[r, bend left, "+"] & odd/1 \arrow[l, bend left, "-"]
\end{tikzcd}
$$

Quiz: Stratifications

Recall: A normal logic program P is *stratified* iff its dependency graph D_P has no cycle involving a negative edge.

Quiz

Consider the normal logic program P where x is the only variable: ...

Standard Models (Stratified Programs)

Definition

Let I be an Herbrand interpretation, Π be a set of predicate symbols.

$$I \mid \Pi \; := \; I \cap \{p(t_1, \ldots, t_n) \mid p \in \Pi,\ t_1, \ldots, t_n \text{ ground terms}\}$$

Let $P_1 \cup \ldots \cup P_n$ be a stratification of the normal program P. Define:

$$\begin{aligned}
M_1 \quad &:= \quad \text{least Herbrand model of } P_1 \text{ such that} \\
&\qquad M_1 \mid \{p \mid p \text{ not defined in } P \text{not defined in } \; P_1 \cup \ldots \cup P_n\} \; = \; \emptyset \\
M_2 \quad &:= \quad \text{least Herbrand model of } P_2 \text{ such that} \\
&\qquad M_2 \mid \{p \mid p \text{ defined nowhere or in } P_1 \text{not defined in } \; P_2 \cup \ldots \cup P_n\} \; = \; M_1 \\
&\vdots \\
M_n \quad &:= \quad \text{least Herbrand model of } P_n \text{ such that} \\
&\qquad M_n \mid \{p \mid p \text{ defined nowhere or in } P_1 \cup \ldots \cup P_{n-1} \text{not defined in } \; P_n\} \; = \; M_{n
\end{aligned}$$

We call $M_P = M_n$ the **standard model** of P.

Example (1)

Let $P_1 \cup P_2 \cup P_3$ with

$$\begin{aligned}
P_1 &= \{num(0) \leftarrow, \quad num(s(x)) \leftarrow num(x)\} \\
P_2 &= \{zero(0) \leftarrow\} \\
P_3 &= \{positive(x) \leftarrow num(x), \ {\sim}zero(x)\}
\end{aligned}$$

be a stratification of P. Then:

$$\begin{aligned}
M_1 &= \{num(t) \mid t \in HU_{\{s,0\}}\} & &= \{num(0), num(s(0)), \ldots\} \\
M_2 &= \{num(t) \mid t \in HU_{\{s,0\}}\} \cup \{zero(0)\} & &= \{zero(0), num(0), num(s(0)), \ldots\} \\
M_3 &= \{num(t) \mid t \in HU_{\{s,0\}}\} \cup \{zero(0)\} & &= \{zero(0), num(0), num(s(0)), \ldots\} \\
&\quad \cup \{positive(t) \mid t \in HU_{\{s,0\}} \setminus \{0\}\} & &\quad \cup \{positive(s(0)), positive(s(s(0))), \ldots\}
\end{aligned}$$

Hence $M_P = M_3$ is the standard model of P.

Properties of Standard Models

Theorem 6.7

Consider a stratified program P. Then:

- M_P does not depend on the chosen stratification of P,
- M_P is a minimal model of P,
- M_P is a supported model of P.

Corollary

For a stratified program P, $comp(P)$ admits a Herbrand model.

Perfect Models and Local Stratification

Stratification may be too demanding

Consider the first-order program P_1 over $\Pi_1 = \{even/1\}$ and $F_1 = \{s/1, 0/0\}$:

- P_1 is not stratified, since $even/1$ depends negatively on itself.

$$even(0) \leftarrow$$
$$even(s(x)) \leftarrow \sim even(x)$$

Observation

P_1 has a clear intended model: $\{even(0), even(s(s(0))), even(s(s(s(s(0))))), \ldots\}$.

Consider, in contrast, the propositional program P_0 over $\Pi_0 = HB_{\{even\},\{s,0\}} = \{even(0)/0, even(s(0))/0, \ldots\}$ and $F_0 = \emptyset$:

- P_0 is stratified.
- The standard model of P_0 is the intended model of P_1.

$$even(0) \leftarrow$$
$$even(s(0)) \leftarrow \sim even(0)$$
$$even(s(s(0))) \leftarrow \sim even(s(0))$$
$$\vdots$$

Perfect Models

Definition

Let P be a normal program over Π and F, and let $\prec$ be a well-founded order on $HB_{\Pi,F}$. Further, let M and N be Herbrand interpretations of P.

- N is **preferable** to M (written $N \triangleleft M$)
  $:\Longleftrightarrow$ for every $B \in N \setminus M$ there exists an $A \in M \setminus N$ such that $A \prec B$.
- A Herbrand model M of P is **perfect** (w.r.t. $\prec$)
  $:\Longleftrightarrow$ there is no Herbrand model of P that is preferable to M.

Well-founded orders admit no infinite descending chains $\ldots \prec c_2 \prec c_1 \prec c_0$.

Example

$$\begin{array}{lcl} p & \leftarrow & \sim q \\ q & \leftarrow & q \end{array}$$

For the well-founded order $q \prec p$, we obtain $\{p\} \triangleleft \{q\}$.

The Standard Model is Perfect

Lemma 6.9

Let P be a normal program and $\prec$ be a well-founded order on $HB_{\Pi,F}$.

- If $N \subsetneq M$ then $N \lhd M$.
- Every perfect model of P is minimal.
- The relation $\lhd$ is a partial order on Herbrand interpretations.

Theorem 6.10

Let P be a stratified normal program over Π and F and for $A, B \in HB_{\Pi,F}$ define $A \prec B \;:\Longleftrightarrow\;$ the predicate symbol of B depends negatively on the predicate symbol of A.

Then M_P is a unique perfect model of P (w.r.t. $\prec$).

The standard model M_P is thus the $\lhd$-least Herbrand model of P.

But how to come up with an order $\prec$ for non-stratified programs?

Local Stratification

Definition

Let P be a normal program over Π and F.

- A **local stratification** for P is a function *strat* from $HB_{\Pi,F}$ to the countable ordinals.
- For a given local stratification *strat* and $A \in HB_{\Pi,F}$, we define $strat({\sim}A) := strat(A) + 1$.
- A clause $c \in P$ is **locally stratified w.r.t.** *strat*
  $:\Longleftrightarrow$ for every $A \leftarrow \vec{K}, L, \vec{M} \in ground(c)$, we have $strat(A) \geq strat(L)$.
- P is **locally stratified w.r.t.** *strat*
  $:\Longleftrightarrow$ all $c \in P$ are locally stratified w.r.t. *strat*.
- P is **locally stratified**
  $:\Longleftrightarrow$ it is locally stratified w.r.t. to some local stratification.

$\rightsquigarrow$ A first-order program is locally stratified iff its ground version is stratified.

Locally Stratified Programs & Perfect Models

Lemma 6.12

Every stratified program is locally stratified.

Example

The program

$$
\begin{aligned}
even(0) &\leftarrow \\
even(s(x)) &\leftarrow \ \sim even(x)
\end{aligned}
$$

is locally stratified (via $\{even(s^n(0)) \mapsto n\}$), but not stratified.

Theorem 6.13

Let P be a normal logic program (over Π and F) that is locally stratified (w.r.t. $strat$), and for $A, B \in HB_{\Pi,F}$ define $A \prec B \ :\Longleftrightarrow \ strat(A) < strat(B)$.

Then P has a unique perfect model (w.r.t. $\prec$).

Well-Supported Models

From Supported to Well-Supported Models

$$p \leftarrow q$$
$$q \leftarrow p$$

has two supported models, $\emptyset$ and $\{p, q\}$.

Only the minimal supported model is intended.

$$p \leftarrow {\sim}q$$
$$q \leftarrow q$$

has two minimal supported models, $\{p\}$ and $\{q\}$.

Only $\{p\}$ is intended: the support of q ("q because q") is unfounded.

$$p \leftarrow {\sim}q$$
$$q \leftarrow {\sim}p$$

has two minimal supported models, $\{p\}$ and $\{q\}$.

- The program is not (locally) stratified.
- The situation is symmetric, so why should we prefer one model over the other?

Well-Supported Models

Definition

Let P be a normal logic program over vocabulary Π, F.

A Herbrand interpretation $I \subseteq HB_{\Pi,F}$ is **well-supported**

$:\Longleftrightarrow$

there is a well-founded order $\prec$ on $HB_{\Pi,F}$ such that:

for each $A \in I$ there is a clause $A \leftarrow \vec{B} \in ground(P)$ with:

- $I \models \vec{B}$
- for every positive atom $C \in \vec{B}$, we have $C \prec A$.

If additionally $I \models P$, then I is a **well-supported model** of P.

Intuitively: Well-supported models disallow circular justifications.

Theorem 6.20

Any locally stratified normal logic program P has a unique well-supported model that coincides with its perfect model.

Well-Supported Models: Examples

$$p \leftarrow {\sim}q$$
$$q \leftarrow q$$

has $\{p\}$ as only well-supported model.

$$p \leftarrow {\sim}q$$
$$q \leftarrow {\sim}p$$

has two well-supported models, $\{p\}$ and $\{q\}$.

$$p \leftarrow q$$
$$p \leftarrow {\sim}q$$
$$q \leftarrow p$$
$$q \leftarrow {\sim}p$$

has no well-supported model.

Preview: Well-supported models are also known as stable models.

Conclusion

Summary

- The immediate consequence operator T_P for a normal logic program P characterizes the **supported models** of P ($=$ the models of $comp(P)$).
- The **stratification** of a program P partitions the program in layers (strata) such that predicates in one layer only negatively/positively depend on predicates in strictly lower/lower or equal layers.
- Programs P that are **stratified** have an intended **standard model** M_P.
- A program is **locally stratified** iff its ground instantiation is stratified.
- Locally stratified programs allow for a unique **perfect model**.
- A normal program P may have zero or more **well-supported models**.

Suggested action points:

- Prove Lemma 6.5; show that every well-supported model is supported.