Semantics for Conditional Literals via the SM Operator

Zach Hansen   Yuliya Lierler

University of Nebraska Omaha

September 2022

Table of Contents

Table of Contents

Motivation

Program Verification

We are interested in the verification of nonground ASP programs.
Thus, we rely on translations to first-order logic and the SM operator.

Meta programming

Conditional literals are expressive, non-standard ASP constructs.
They are frequently used in meta-programming (Kaminski et al. 2021).
Examples include:

- optimization statements;
- reasoning about actions;
- reasoning about preferences;
- guess-and-check programming.

Running Example

The Graph Coloring Problem (CS Lecture Notes)

$$\{asg(V,I)\} \text{ :- } vtx(V);\ color(I).$$
$$\text{:- }\ not\ asg(V,r);\ not\ asg(V,g);\ not\ asg(V,b);\ vtx(V).$$
$$\text{:- }\ asg(V,I);\ asg(V,J);\ I \neq J;\ vtx(V);\ color(I;J).$$
$$\text{:- }\ asg(V,I);\ asg(W,I);\ vtx(V;W);\ color(I);\ edge(V,W).$$

The Graph Coloring Problem (Rewritten)

$$\{asg(V,I)\} \text{ :- } vtx(V);\ color(I).$$
$$\text{:- }\ not\ asg(V,I) :\ color(I);\ vtx(V).$$
$$\text{:- }\ asg(V,I);\ asg(V,J);\ I \neq J;\ vtx(V);\ color(I;J).$$
$$\text{:- }\ asg(V,I);\ asg(W,I);\ vtx(V;W);\ color(I);\ edge(V,W).$$

Table of Contents

Abstract Program Syntax

Conditional Logic Programs

Terms are variables, object constants, or $f(t_1, \ldots, t_k)$

Atomic formulas are $t_1 = t_2$ or **atoms** $p(t_1, \ldots, t_k)$

Basic literals are atomic formulas (optionally negated)

Conditional literals are $H : l_1, \ldots, l_m$ (abbreviated $H : \mathbf{L}$)

Rules have the form $H_1 \mid \cdots \mid H_m \leftarrow B_1; \ldots; B_n$.

A *(conditional logic) program* is a finite set of rules.

Basic Choice Rules

$$\{p(\mathbf{t})\} \leftarrow B_1; \ldots; B_n$$

is considered to be shorthand for

$$p(\mathbf{t}) \mid \mathit{not}\ p(\mathbf{t}) \leftarrow B_1; \ldots; B_n$$

Program Syntax Continued

Explicit Program Signatures

For a signature $\sigma = (\mathcal{O}, \mathcal{F}, \mathcal{P})$ of a first-order language:

- $\mathcal{O}$ is the set of object constants;
- $\mathcal{F}$ is the set of function symbols (non-zero arity);
- $\mathcal{P}$ is the set of predicate constants;
- G_σ is the set of all ground terms constructed from $\mathcal{O}$ and $\mathcal{F}$ of σ.

For a program Π:

- $\sigma = (\mathcal{O}_\Pi, \mathcal{F}_\Pi, \mathcal{P}_\Pi)$, where $\mathcal{O}_\Pi$, $\mathcal{F}_\Pi$, and $\mathcal{P}_\Pi$ contain all the object constants, function symbols, and predicate constants occurring in Π;
- $\mathcal{G}_\Pi$ denotes $\mathcal{G}_{(\mathcal{O}_\Pi, \mathcal{F}_\Pi, \mathcal{P}_\Pi)}$.

Table of Contents

Global vs. Local Variables

Global vs. Local Variables

- A variable is *global* in a conditional literal $H : \mathbf{L}$ if it occurs in H but not in $\mathbf{L}$;
- All other variables occurring in a conditional literal are *local*.
- All variables are global in basic literals;
- A variable is global in a rule if it is global in at least one rule literal.

Example

In rule

$$\text{:- } \ not\ asg(V, I) : \ color(I);\ vtx(V).$$

with conditional literal

$$not\ asg(V, I) : \ color(I);$$

V is a global variable, whereas I is local.

Syntactic Transformation ϕ

$\phi_{\mathbf{z}}$

For a rule R with global variables $\mathbf{z}$:

- for a conditional literal $H : \mathbf{L}$ occurring in the body of R with local variables $\mathbf{x}$, $\phi_{\mathbf{z}}(H : \mathbf{L})$ is $\forall \mathbf{x}\,(\phi_{\mathbf{z}}(\mathbf{L}) \rightarrow \phi_{\mathbf{z}}(H))$;
- for a conditional literal $H : \mathbf{L}$ occurring in the head of R with local variables $\mathbf{x}$, $\phi_{\mathbf{z}}(H : \mathbf{L})$ is $\exists \mathbf{x}\big(\,(\phi_{\mathbf{z}}(\mathbf{L}) \rightarrow \phi_{\mathbf{z}}(H)) \wedge \neg\neg\phi_{\mathbf{z}}(\mathbf{L})\big)$.

ϕ

$\phi(R)$ is the formula $\forall \mathbf{z}(\phi_{\mathbf{z}}(B_1) \wedge \cdots \wedge \phi_{\mathbf{z}}(B_n) \rightarrow \phi_{\mathbf{z}}(H_1) \vee \cdots \vee \phi_{\mathbf{z}}(H_m))$ where $\mathbf{z}$ is the list of the global variables of R

Graph Coloring Example

Conditional Literal Translation

Transformation ϕ applied to the conditional literal

$$not\ asg(V, I)\ :\ color(I)$$

produces formula

$$\forall i(color(i) \rightarrow \neg asg(v, i))$$

$\phi(\Pi)$

$$\forall vi((vtx(v) \wedge color(i)) \rightarrow asg(v, i) \vee \neg asg(v, i))$$
$$\forall v(\ (\forall i(color(i) \rightarrow \neg asg(v, i)) \wedge vtx(v)) \rightarrow \bot)$$
$$\forall vij((asg(v, i) \wedge asg(v, j) \wedge i \neq j \wedge vtx(v) \wedge color(i) \wedge color(j)) \rightarrow \bot)$$
$$\forall viw((asg(v, i) \wedge asg(w, i) \wedge vtx(v; w) \wedge color(i) \wedge edge(v, w)) \rightarrow \bot)$$

Semantics of Conditional Logic Programs

Semantics via the SM operator

An interpretation is a **p**-*stable model* of a first-order sentence F when it is a model of $\text{SM}_{\mathbf{p}}[F]$.

For a conditional logic program Π and a Herbrand interpretation I over the signature $(\mathcal{O}_\Pi, \mathcal{F}_\Pi, \mathcal{P}_\Pi)$, I is an *answer set* of Π when I is a $\mathcal{P}_\Pi$-stable model of $\phi(\Pi)$.

Table of Contents

Conditional Programs via Infinitary Propositional Logic

Traditional Characterization of Conditional Literals:

(i) Syntactic reduction to IPL formulas[1];

(ii) Semantics defined by IPL stable model semantics[2].

Closed Conditional Literal Transformations

If $\mathbf{x}$ is the list of variables occurring in a conditional literal $H : \mathbf{L}$:

Body: $\tau(H : \mathbf{L})$ is the *conjunction* of the formulas

$$\tau(\mathbf{L}^{\mathbf{x}}_{\mathbf{r}}) \rightarrow \tau(H^{\mathbf{x}}_{\mathbf{r}})$$

over all tuples of ground terms $\mathbf{r} \in \mathcal{G}^{|\mathbf{x}|}$.

[1] A. Harrison, V. Lifschitz, and F. Yang. "The Semantics of Gringo and Infinitary Propositional Formulas". In: *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning (KR'14)*. Ed. by C. Baral, G. De Giacomo, and T. Eiter. AAAI Press, 2014

[2] M. Truszczyński. "Connecting First-Order ASP and the Logic FO(ID) through Reducts". In: *Correct Reasoning: Essays on Logic-Based AI in Honour of Vladimir Lifschitz*. Ed. by E. Erdem et al. Vol. 7265. Lecture Notes in Computer Science. Springer-Verlag, 2012, pp. 543–559

Rules to Closed Rules

Instantiations of rule R w.r.t. a set $\mathcal{G}$ of ground terms

$inst_{\mathcal{G}}(R) = \{R^{\mathbf{z}}_{\mathbf{u}} \mid \mathbf{u} \in \mathcal{G}^{\mathbf{z}}\}^{\wedge}$

Program Π, rule R in Π

$$\tau(R) = \{\tau(r) \mid r \in inst_{\mathcal{G}_\Pi}(R)\}^{\wedge}.$$

Similarly,

$$\tau(\Pi) = \{\tau(R) \text{ w.r.t. } \mathcal{G}_\Pi \mid R \in \Pi\}^{\wedge}.$$

Table of Contents

Connecting Semantics for Conditional Literals

Summary

- We have presented our translation ϕ, which produces *nonground* first-order formulas from logic programs.
- We have reviewed translation τ, which produces infinitary propositional formulas from logic programs.
- We use a previously established result to show that the $\mathcal{P}_{\Pi}$-stable models of $\phi(\Pi)$ coincide with the stable models of $\tau(\Pi)$.

Grounding

F is a FO sentence over σ

$gr(\bot) = \bot$;

$gr(A) = A$ for a ground atom A;

$gr(t_1 = t_2)$ is $\top$ if t_1 is identical to t_2, and $\bot$ otherwise, for ground terms t_1, t_2;

If $F = G \vee \ldots \vee H$, then $gr(F) = gr(G) \vee \cdots \vee gr(H)$;

If $F = G \wedge \ldots \wedge H$, then $gr(F) = gr(G) \wedge \cdots \wedge gr(H)$;

If $F = G \rightarrow H$, then $gr(F) = gr(G) \rightarrow gr(H)$;

If $F = \exists \mathbf{x} G(\mathbf{x})$, then $gr(F) = \{gr(G(\mathbf{u})) \mid \mathbf{u} \in \mathcal{G}^{\mathbf{x}}_{\sigma}\}^{\vee}$;

If $F = \forall \mathbf{x} G(\mathbf{x})$, then $gr(F) = \{gr(G(\mathbf{u})) \mid \mathbf{u} \in \mathcal{G}^{\mathbf{x}}_{\sigma}\}^{\wedge}$.

Main Results

Theorem (Syntactic Identity)

For any conditional logic program Π *containing at least one object constant,* $gr\big(\phi(\Pi)\big)$ *is identical to* $\tau(\Pi)$.

Theorem (Main Theorem)

For any conditional logic program Π *containing at least one object constant and any Herbrand interpretation* I *over* $(\mathcal{O}_\Pi, \mathcal{F}_\Pi, \mathcal{P}_\Pi)$*, the following conditions are equivalent:*

- I *is a* $\mathcal{P}_\Pi$*-stable model of* $\phi(\Pi)$*;*
- I *is a clingo answer set of* Π.

Graph Coloring Example

$\mathcal{G}_\Pi = \{1, g\}$

Take rule R to be $\;$:- $\;$ *not* $asg(V, I) : color(I);\; vtx(V)$. The grounding of $\phi(R)$ replaces global variables:

$$gr(\phi(R)) = \{\big(gr\,(\forall i(color(i) \to \neg asg(1, i)) \wedge vtx(1)) \to \bot\big),$$
$$\big(gr\,(\forall i(color(i) \to \neg asg(g, i)) \wedge vtx(g)) \to \bot\big)\}^\wedge$$

Take closed rule r to be $\;$:- $\;$ *not* $asg(1, I) : color(I);\; vtx(1)$. Then, $\tau(r) = ((color(1) \to \neg asg(1,1)) \wedge (color(g) \to \neg asg(1,g))) \wedge vtx(1) \to \bot$

Comparing Transformations w.r.t. the set of ground terms $\{1, g\}$

$$(\forall i(color(i) \to \neg asg(1, i)) \wedge vtx(1)) \to \bot$$

is equivalent to

$$((color(1) \to \neg asg(1,1)) \wedge (color(g) \to \neg asg(1,g))) \wedge vtx(1) \to \bot$$

Conclusion

Contribution

We have provided semantics for conditional literals via the SM operator which do not refer to grounding.
This enables us to create modular proofs of correctness for a broader class of nonground programs.

Future Work

We are potentially interested in integrating this work into Anthem. For instance, the graph coloring example explored here could be verified by an extended version of Anthem.