The University of Texas at Austin

Final Research Paper

Transformation, Simplification, and Completion of clingo Programs Using Formal Methods

Raeeca Narimani

supervised by
Dr. Vladimir Lifschitz

May 15, 2017

Abstract

Answer Set Programming (ASP) is a mathematical approach for building a set of stable models from a set of logical rules. clingo is a declarative programming language that involves ASP, and the set of rules in each clingo program has an underlying logic which for the purpose of verification is better explained using formal methods. anthem is a software system aiming to automate verifications such as proofs of programming correctness. In this paper, the process of transformation, simplification, and completion of clingo programs is expanded, and the results are compared to the output of anthem.

Introduction

ASP evolves around the idea that a logic program can have one or more sets of answers. For a positive definite program, starting from null, a possible answer set builds up with processing every rule in the program until an expected fixed point is reached. The answer set of a negative program where negation as failure applies to at least one predicate in the body is computed after generating the reduct [Gelfond and Lifschitz, 1988].

1 Transformation Phi

Producing a formula representation from a rule in CLINGO is achieved by applying transformation ϕ [Harrison *et al.*, 2017, Section 4] to the rule. Initially in this paper examples of applying transformation ϕ to a variety of rules in CLINGO programming language have been demonstrated. Later on, this will lead to simplification and completion applied to the same examples.

The formula representation R is denoted by ϕR. Hence looking at:

```
p(1..5).                                  (1)
```

we see an example of a CLINGO rule with no body. Its formula representation $\phi(p(1..5))$ is equivalent to:

$$V \in 1..5 \rightarrow p(V).$$

This simply states that there exists a value V belonging to the set of integers 1..5 such that $p(V)$ holds. Now consider a CLINGO with a body:

```
p(N) :- N = 1..5.                         (2)
```

with $\phi(p(N) \leftarrow N = 1..5)$ arriving at its formula representation

$$V \in N \wedge \phi(N = 1..5) \rightarrow p(V).$$

knowing that transformation ϕ apples to the body rules in the following fashion:

$$V \in N \wedge \exists X_1, X_2(X_1 \in N \wedge X_2 \in 1..5 \wedge X_1 = X_2) \rightarrow p(V).$$

This formula representation is correct, but unsimplified and overly complicated. The simplification step is described later on this paper. However, this formula correctly states that there exists a value V belonging to a set of values N, and there exist values X_1 and X_2 such that X_1 belongs to N, X_2 belongs to the integer set 1..5, and X_1 is equal to X_2. In another example:

```
p(N + 1) :- q(N).                    (3)
```

we see a CLINGO rule such that its $\phi(p(N+1) \leftarrow q(N))$ is:

$$V \in N+1 \wedge \phi(q(N)) \rightarrow p(V).$$

which can be further written as:

$$V \in N+1 \wedge \exists X(X \in N \wedge q(X)) \rightarrow p(V).$$

Also here $q(N)$ in the body is described as there exist a value X belonging to N such that $q(X)$ holds. Continuing on the examples:

```
p(N + 1) :- q(N), not t(N).          (4)
```

is the representation of a CLINGO rule, containing negation as failure applied to a predicate in the body. Its $\phi(p(N+1) \leftarrow q(N) \wedge \mathit{not}\, t(N))$ is:

$$V \in N+1 \wedge \phi(q(N) \wedge \mathit{not}\, t(N)) \rightarrow p(V),$$

with transformation ϕ being applied separately to each body cause, the rule can be further written as:

$$V \in N+1 \wedge \exists X(X \in N \wedge q(X)) \wedge \exists X(X = N \wedge \neg\, t(X)) \rightarrow p(V).$$

Once transformation is completed, the notation "not" is changed to the symbol "$\neg$". In the following example is a constraint rule, implying an empty head for the rule:

```
:- p(M), M = 1..5.                   (5)
```

that its $\phi(\bot \leftarrow p(M) \wedge M = 1..5)$ is:

$$\phi(p(M)) \wedge \phi(M = 1..5).$$

which can be further written as:

$$\exists X(X \in N \wedge p(X)) \wedge \exists X_1, X_2(X_1 \in M \wedge X_2 \in 1..5 \wedge X_1 = X_2) \rightarrow \bot.$$

Here, falsity ($\bot$) is the simbol for truth value of the constraint's head.

```
P.                                                   (6)
```

that its $\phi(P)$ is: $V \in P \to V$. In the following example, a comparison is embedded in the rule's body:

```
p(M, N) :- M..M + 1 = N..N + 1, q(M, N).           (7)
```

that its $\phi(p(M,N) \leftarrow M..M+1 = N..N+1 \wedge q(M,N))$ is:

$$V_1 \in M \wedge V_2 \in N \wedge \phi(M..M+1 = N..N+1 \wedge q(M,N)) \to p(V1, V2),$$

and ϕ is applied to each separate formula and clause in the body:

$$V_1 \in M \wedge V_2 \in N \wedge \phi(M..M+1 = N..N+1) \wedge \phi(q(M,N)) \to p(V_1, V_2),$$

which is written as:

$$V_1 \in M \wedge V_2 \in N \wedge \exists X_1, X_2(X_1 \in M..M+1 \wedge X_2 \in N..N+1 \wedge X_1 = X_2)$$

$$\wedge \exists X_3, X_4(X_3 \in M \wedge X_4 \in N \wedge q(X_3, X_4)) \to p(V_1, V_2).$$

2 Simplification

Below is the step-by-step process of simplifying [Harrison *et al.*, 2017, Section 4] for (1) through (7). At this step, we use equality whevever possible to replace redunded values in a formula representation.

Looking at (1) with the formula representation $V \in 1..5 \to p(V)$, we see that it's not simplifiable any further- the only singleton value here is V, which belongs to the integer set 1..5.

For (2) with an unsimplified formula representation:

$$V \in N \wedge \exists X_1, X_2(X_1 \in N \wedge X_2 \in 1..5 \wedge X_1 = X_2) \to p(V).$$

Looking at $V \in N$ we conclude that singleton values V and N are in fact equal, and so are X_1 and N. Also since $X_1 = X_2$ already holds in the formula, we use it to replace every value of X_2 with a value X_1.

$$V = N \wedge \exists X1_1(X_1 = N \wedge X_1 \in 1..5) \to p(V).$$

Now since $X_1 = N$ holds, the new form of the formula representation for (2) is even further simplified as following:

$$V = N \wedge N \in 1..5 \to p(V).$$

For (3) with unsimplified formula representation:

$$V \in N + 1 \wedge \exists X(X \in N \wedge q(X)) \to p(V).$$

Here again, $X \in N$ is implying $X = N$, hence:

$$V \in N + 1 \wedge \exists X(X = N \wedge q(X)) \to p(V).$$

Now by replacing values of X with N as well as removing the existential quantifier, the final representation stated below is as simplified as possible:

$$V \in N + 1 \wedge q(N) \to p(V).$$

For (4) and its unsimplified formula representation:

$$V \in N + 1 \wedge \exists X(X \in N \wedge q(X)) \wedge \exists X(X \in N \wedge \neg t(X)) \to p(V).$$

Its obvious that $X = N$ inside both existential quantifiers:

$$V \in N + 1 \wedge \exists X(X = N \wedge q(X)) \wedge \exists X(X = N \wedge \neg t(X)) \to p(V).$$

which again allows a replacement for all instances of X and removal of the existential quantifiers:

$$V \in N + 1 \wedge q(N) \wedge \neg t(N) \to p(V).$$

Here, the formula $V \in N + 1$ beholds unbound values V and $N + 1$. To set V and $N + 1$ equal, we need to have them as bound values. This step is later described in the completion part for (4).

The unsimplified formula representation for (5):

$$\exists X(X \in N \wedge p(X)) \wedge \exists X_1, X_2(X_1 \in M \wedge X_2 \in 1..5 \wedge X1_1 = X_2) \to \bot$$

is further simplified as:

$$\exists X(X = N \wedge p(X)) \wedge \exists X_1(X_1 = M \wedge X_1 \in 1..5) \to \bot$$

which after replacing X with N and X_1 with M, implies:

$$p(N) \wedge M \in 1..5 \rightarrow \bot.$$

Using the implication law, this representation is also equivalent to:

$$\neg\,(p(M) \wedge M \in 1..5).$$

For (6) we have $V \in P \rightarrow V$, which is further simplified to $V = P \rightarrow V$.

Finally (7), with an unsimplified formula representation:

$$V_1 \in M \wedge V_2 \in N \wedge \exists X_1, X_2(X_1 \in M..M+1 \wedge X_2 \in N..N+1 \wedge X_1 = X_2)$$

$$\wedge \exists X_3, X_4(X_3 \in M \wedge X_4 \in N \wedge q(X_3, X_4)) \rightarrow p(V_1, V_2).$$

By letting some value X equal to X_1, and replacing for all values of X_1 and X_2, the formula can be further simplified to:

$$V_1 = M \wedge V_2 = N \wedge \exists X(X \in M..M+1 \wedge X \in N..N+1) \wedge$$

$$\exists X_3, X_4(X_3 = M \wedge X_4 = N \wedge q(X_3, X_4)) \wedge \rightarrow p(V_1, V_2).$$

and by replacing X_3 for M and X_4 for N, the simplified formula can be rewritten as:

$$V_1 = M \wedge V_2 = N \wedge \exists X(X \in M..M+1 \wedge X \in N..N+1) \wedge q(M, N) \rightarrow p(V_1, V_2).$$

3 Completion

Completion is a rather abstract process for unifying programming rules consisting of two or more scattered parts. This step the purpose of readability in verification for users as well as for theorem prover software systems.

If the definition of p/n (literal p with arity n) in a program $\mathbf{P}$ has the rule set $\{R_1, .., R_n\}$ with an arbitrary rule $R_i : \mathbf{F}_i \rightarrow p(\mathbf{V})$ and $\mathbf{V}$ is a tuple with i arguments, then the completion rule of p/n is as following [Harrison *et al.*, 2017, Section 5]:

$$\forall \mathbf{V}(p(\mathbf{V}) \leftrightarrow \vee_{k=1}^{n} \exists \mathbf{U}_i \mathbf{F}_i)$$

For rule (1), the completion representation of $V \in 1..5 \rightarrow p(V)$ has the completion form:

$$\forall V(q(V) \leftrightarrow \exists N(V \in N \wedge N \in 1..5).$$

which can be expanded as the following conjunction:

$$\forall V((q(V) \rightarrow \exists N(V \in N \wedge N \in 1..5)) \wedge (\exists N(V \in N \wedge N \in 1..5) \rightarrow q(V))).$$

and is equivalent to:

$$\forall V(q(V) \leftrightarrow V = 1 \wedge V = 2 \wedge V = 3 \wedge V = 4 \wedge V = 5).$$

For rule (2), the completion representation of $V = N \wedge N \in 1..5 \rightarrow p(V)$ is as following:

$$\forall V(p(V) \leftrightarrow \exists N(V = N \wedge N \in 1..5)).$$

This representation can be expanded as:

$$\forall V((p(V) \rightarrow \exists N(V = N \wedge N \in 1..5)) \wedge (\exists N(V = N \wedge N \in 1..5) \rightarrow p(V))).$$

Hence in regards to the simplification rules when $V = N$, it can be rewritten as following:

$$\forall V((p(V) \rightarrow V \in 1..5) \wedge (V \in 1..5) \rightarrow p(V)).$$

Resulting:

$$\forall V(p(V) \leftrightarrow V \in 1..5)$$

Rule (3), with the simplified formula $V \in N + 1 \wedge q(N) \rightarrow p(V)$ has the completion form:

$$\forall V(p(V) \leftrightarrow \exists N(V \in N + 1 \wedge q(N)))$$

Here, it's possible to rewrite the formula $V \in N + 1$ as $V = N + 1$:

$$\forall V(p(V) \leftrightarrow \exists N(V = N + 1 \wedge q(N)))$$

and the results can be expanded as:

$$\forall V(p(V) \rightarrow \exists N(V = N + 1 \wedge q(N)) \wedge \exists N(V = N + 1 \wedge q(N)) \rightarrow p(V))$$

Since $N = V - 1$, we can replace the value of V in the completion form as following:

$$\forall V(p(V) \rightarrow \exists N(V - 1 = N \wedge q(N))) \wedge \exists N(V - 1 = N \wedge q(N)) \rightarrow p(V))$$

Hence by replacing N with $V - 1$, the completion form is simplified to:

$$\forall V(p(V) \rightarrow q(V - 1)) \wedge (q(V - 1) \rightarrow p(V))$$

Which is finally presented as:

$$\forall V(p(V) \leftrightarrow q(V - 1))$$

Rule (4), with the simplified form $V \in N + 1 \wedge q(N) \wedge \neg t(N) \rightarrow p(V)$ has the completion form:

$$\forall V(p(V) \leftrightarrow \exists N(V \in N \wedge \neg t(N) \wedge q(N)))$$

and can be expanded as:

$$\forall V((p(V) \rightarrow \exists N(V \in N \wedge \neg t(N) \wedge q(N)) \wedge (\exists N(V \in N \wedge \neg t(N) \wedge q(N)) \rightarrow p(V)))$$

With the use of simplification, this can be rewritten as:

$$\forall V((p(V) \rightarrow \exists N(V = N \wedge \neg t(N) \wedge q(N)) \wedge (\exists N(V = N \wedge \neg t(N) \wedge q(N)) \rightarrow p(V)))$$

Hence by replacing N with V, the completion form is simplified to:

$$\forall V((p(V) \rightarrow \neg t(V) \wedge q(V)) \wedge (\neg t(V) \wedge q(V) \rightarrow p(V)))$$

Which is finally presented as:

$$\forall V((p(V) \leftrightarrow \neg t(V) \wedge q(V))$$

Rule (5), with the simplified form $p(N) \wedge M \in 1..5 \rightarrow \perp$ has the completion form:

$$\forall V_1 V_2(\perp \leftrightarrow \exists M, N(V_1 \in N \wedge p(N) \wedge V_2 \in M \wedge M \in 1..5)$$

and can be expanded as:

$$\forall V_1 V_2((\perp \rightarrow \exists M, N(V_1 \in N \wedge p(N) \wedge V_2 \in M \wedge M \in 1..5)$$

$$\wedge\,(\exists M,N(V_1 \in N \wedge p(N) \wedge V_2 \in M \wedge M \in 1..5 \rightarrow \bot))$$

Using simplification, this can be rewritten as:

$$\forall V_1 V_2((\bot \rightarrow \exists M,N(V_1 = N \wedge p(N) \wedge V_2 = M \wedge M \in 1..5)$$

$$\wedge\,(\exists M,N(V_1 = N \wedge p(N) \wedge V_2 = M \wedge M = 1..5 \rightarrow \bot))$$

By replacing N with V_1 and with M V_2:

$$\forall V_1 V_2((\bot \leftrightarrow (p(V_1) \wedge V_2 \in 1..5) \wedge\ (p(V_1) \wedge V_2 \in 1..5 \rightarrow \bot))$$

Rule (6), with the simplified form $V = P \rightarrow V$, has the completion form:

$$\forall V(V \leftrightarrow \exists N(N \in P \wedge P = V))$$

and can be simplified to:

$$\forall V(V \leftrightarrow \exists N(N = P \wedge P = V))$$

and expressed as:

$$\forall V(V \leftrightarrow P = V)$$

Rule (7), with the simplified form:

$$V_1 = M \wedge V_2 = N \wedge \exists X(X \in M..M{+}1 \wedge X \in N..N{+}1) \wedge q(M,N) \rightarrow p(V_1,V_2)$$

has the completion form:

$$\forall V_1,V_2(p(V_1,V_2) \leftrightarrow \exists N,M(V_1 = M \wedge V_2 = N \wedge \exists X(X \in M..M{+}1 \wedge X \in N..N{+}1)$$

$$\wedge q(M,N))).$$

which can be expanded:

$$\forall V_1,V_2((p(V_1,V_2) \rightarrow \exists N,M(V_1 = M \wedge V_2 = N \wedge \exists X(X \in M..M{+}1 \wedge X \in N..N{+}1) \wedge q(M,N)))$$

$$\wedge(\exists N,M(V_1 = M \wedge V_2 = N \wedge \exists X(X \in M..M{+}1 \wedge X \in N..N{+}1) \wedge q(M,N) \rightarrow p(V_1,V_2)))$$

By replacing values V_1 for M and V_2 for N, the completion form can be further simplified to:

$$\forall V_1,V_2((p(V_1,V_2) \rightarrow \exists X(X \in V_1..V_1 + 1 \wedge X \in V_2..V_2 + 1) \wedge q(V_1,V_2)))$$

$$\wedge(\exists X(X \in V_1..V_1 + 1 \wedge X \in V_2..V_2 + 1) \wedge q(V_1,V_2) \rightarrow p(V_1,V_2)))$$

and rewritten as:

$$\forall V_1 V_2((p(V_1,V_2) \leftrightarrow \exists X(X \in V_1..V_1 + 1 \wedge X \in V_2..V_2 + 1)))$$

References

[Gelfond and Lifschitz, 1988] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In Robert Kowalski and Kenneth Bowen, editors, *Proceedings of International Logic Programming Conference and Symposium*, pages 1070–1080. MIT Press, 1988.

[Harrison *et al.*, 2017] Amelia Harrison, Vladimir Lifschitz, and Dhananjay Raju. Program completion in the input language of GRINGO. Submitted for publication, 2017.