APPROXIMATE TERMINATION ANALYSIS BY ABSTRACT INTERPRETATION

Two uses of abstract interpretation

- The semantics on which termination can be observed
- The approximation which makes the analysis feasible

Motivations I

Notions of termination

A goal G *terminates* in the program P if the execution of G *in* P terminates in a finite time.

Two different notions of termination for logic programs:

- Existential Termination: *at least one answer for G is obtained in a finite time.*
- Universal Termination: *all the answers for G are obtained in a finite time.*

There are basically two approaches to (Universal) Termination of logic programs:

- Correct and complete methods providing manually verifiable criteria that ensure termination.
- Techniques providing sufficient automatically checkable conditions.

Motivations II

Most of the termination analyses proposed so far prove *a strict decrease of some measure (over a well founded domain) on consecutive procedure calls.*

Several systems exist for automatic termination analysis, such as TermiWeb, TermiLog, cTI, Mercury's Termination analyzer.

- These systems are very powerful:
  - they ensure termination of non-trivial queries.
  - they prove termination of large classes of goals.
- These systems are not able to analyze *all* programs: some problems arise when termination depends on the structure of terms.
- The techniques proposed in this paper can be used to improve the existing methods.

Motivations III

Examples

$$\begin{aligned} P_1: \;& at(telaviv, mary). \\ & at(jerusalem, mary). \\ & at(X, fido) \leftarrow at(X, mary), near(X). \\ & near(jerusalem). \end{aligned}$$

Using TermiLog and cTI, we can not prove that the goal $at(X, Y)$ in P_1 terminates for any X and Y.

Let

$$\begin{aligned} P_2: \;& p(a, b) \\ & p(a, f(Y)) \leftarrow p(a, Y) \\ & q(f(X), Y) \leftarrow p(X, Y) \end{aligned}$$

$p(a, Y)$ terminates *if* Y *is bound* $\Rightarrow$ $q(f(a), Y)$ terminates *if* Y *is bound*,

$p(b, Y)$ terminates *for any* Y $\Rightarrow$ $q(f(b), Y)$ terminates *for any* Y.

Using TermiLog and cTI, we can prove that $q(f(b), Y)$ terminates in P_2 *if* Y *is bound*, while $q(f(b), Y)$ terminates for any Y.

The proposed approach

We propose abstract interpretation in order to

1. systematically derive a suitable semantics to model termination,
2. systematically derive new effective abstraction useful for termination analysis,
3. reconstruct as abstract interpretations of the “termination semantics” most of the existing automatic methods,
4. systematically combine all the different analyses in a more powerful automatic system.

A semantics for reasoning on termination

– Termination is closely related to the existence of infinite derivations.

– To reason about termination in semantic terms we need a fixpoint semantics modelling the infinite behavior.

– To model the infinite behavior in an *And-compositional way*, we have to model the *substitutions computed by infinite and successful* derivations (*exact answers*).

- in fact, $A_1, \ldots, A_n$ has an infinite derivation via a fair selection rule iff
  1. *at least one A_i has an infinite derivation,*
  2. *each A_j has a successful or an infinite derivation,*
  3. *all the chosen derivations for $A_1, \ldots, A_n$ compute compatible substitutions.*

– None of the fixpoint semantics defined in literature models exact answers and is based on a co-continuous operator.

A semantics modelling exact answers I

The semantic domain:

- to represent *possibly infinite* answers of infinite and successful derivations, we use sequences of substitutions.
  - Finite sequences represent answers of successful derivations.

$$:: \vartheta_1 :: \vartheta_2 :: \ldots :: \vartheta_n$$

  - Infinite sequences represent possibly infinite answers of infinite derivations.

$$:: \vartheta_1 :: \vartheta_2 :: \ldots :: \vartheta_n :: \ldots$$

- to model exact answers and not their instances, we have to keep information on the number of rewriting steps.

- To obtain And-compositionality we consider only fixed rewriting steps

$$::_{n_1} \vartheta_1 ::_{n_2} \vartheta_2 :: \ldots ::_{n_n} \vartheta_n :: \ldots$$

A semantics modelling exact answers II

The fixpoint semantics modelling exact answers is:

- correct and fully abstract,
- based on a co-continuos operator,
- And-compositional and compositional **wrt** instantiation.

$$
\begin{aligned}
P_3 :\ & p(a). \\
& p(f(X)) \leftarrow p(X). \\
& q(a) \leftarrow p(X).
\end{aligned}
$$

$$
\begin{aligned}
gfp(\mathcal{P}(P_3))p(X) = \{ & <::_1 X/a > \\
& <::_1 X/f(X_1) ::_1 X/f(a) > \\
& <::_1 X/f(X_1) ::_1 X/f(f(X_2)) ::_1 X/f(f(a)) > \\
& \vdots \\
& <::_1 X/f(X_1) ::_1 X/f(f(X_2)) ::_1 \ldots ::_1 X/f^n(X_n) :: \ldots >\} \\
gfp(\mathcal{P}(P_3))q(X) = \{ & <::_1 X/a ::_1 X/a > \\
& \vdots \\
& <::_1 X/a ::_1 \ldots ::_1 X/a ::_1 \ldots >\}
\end{aligned}
$$

This semantics is systematically obtained by abstract interpretation techniques from a semantics modelling (possibly infinite) SLD-trees.

The approximate semantics

The idea is to use depth-k substitutions, i.e. substitutions whose terms are cut at depth-k,

Example *For* $k = 2$,

$X/f(g(a), g(Y)) \Rightarrow X/f(V, W)$

The abstraction:

- we lose information on the number of steps,
- we approximate a finite sequence

$$:: \vartheta_1 :: \vartheta_2 :: \ldots :: \vartheta_n \text{ with } < \vartheta, \Box >$$

  where $\vartheta = \alpha_k(\vartheta_i) = \alpha_k(\vartheta_j)$ for all $j > i$,

- we approximate an infinite sequence

$$:: \vartheta_1 :: \vartheta_2 :: \ldots :: \vartheta_n :: \ldots \text{ with } < \vartheta, \Diamond >$$

  where $\vartheta = \alpha_k(\vartheta_i) = \alpha_k(\vartheta_j)$ for all $j > i$.

$$\begin{array}{ll} P_3: & p(a). \\ & p(f(X)) \leftarrow p(X). \\ & q(a) \leftarrow p(X). \end{array}$$

$$\begin{array}{ll} gfp(\mathcal{P}^k(P_3)(p(X)) & = \{< X/a, \Box >, < X/f(a), \Box >, \\ & < X/f(f(\tilde{V})), \Box >, < X/f(f(\tilde{V})), \Diamond >\} \\ gfp(\mathcal{P}^k(P_3)(q(X)) & = \{< X/a, \Box >, < X/a, \Diamond >\} \end{array}$$

Toward a termination analysis

With our abstract semantics, we can determine *a superset of the goals having at least an infinite derivation*.

$$
\begin{aligned}
Inf_P^k = \{G \mid\ & G = (A_1, \ldots, A_n)\vartheta \\
& \exists \text{ an } mgu(G, (A_1\sigma_1, \ldots, A_n\sigma_n)) \text{ such that} \\
& \exists \bar{i} \in \{1, \ldots, n\}, < \sigma_{\bar{i}}, \diamond > \in gfp(\mathcal{P}^k(P))(A_{\bar{i}}) \\
& \text{for } i = \{1, \ldots, n\}, < \sigma_i, \_ > \in gfp(\mathcal{P}^k(P))(A_i)\}
\end{aligned}
$$

For our approximation the following properties hold:

1. If G has an infinite derivation in P
   $\Rightarrow \forall k, G \in Inf_P^k$.
2. If G does not have an infinite derivation in P
   $\Rightarrow \exists l$, s.t. $\forall k > l, G \notin Inf_P^k$.

We can use Inf_P^k for:

- universal termination analysis,
- define an analysis which allows us to ensure that replacing the breadth-first with depth-first search rule is “safe”.

Universal Termination

Definition[Ruggieri 1999]
A goal G in P $\exists$-universally terminates if there exists a selection rule s such that every derivation of G (via s) is finite.

Result[Ruggieri 1999]
A goal G in P $\exists$-universally iff it universally terminates **wrt** the set of fair selection rules.

Our result:
G in P $\exists$-universally terminates *iff* there exists a k such that $G \notin Inf_P^k$.

Examples I

Let us go back to the first two examples:

$$\begin{aligned} P_1 :\ & at(telaviv, mary). \\ & at(jerusalem, mary). \\ & at(X, fido) \leftarrow at(X, mary), near(X). \\ & near(jerusalem). \end{aligned}$$

Our analysis for $k = 2$:

$$\begin{aligned} gfp(\mathcal{P}^2(P_1))(at(X,Y)) = &\ \{< \{X/telaviv, Y/fido\}, \Box >, \\ & < \{X/jerusalem, Y/fido\}, \Box >\} \\ gfp(\mathcal{P}^2(P_1))(near(X)) = &\ \{< \{X/jerusalem\}, \Box >\} \end{aligned}$$

$at(X, Y)$ terminates for any X and Y,

$at(X, Y) \notin Inf^2_{P_1}$, since $Inf^2_{P_1} = \emptyset$.

Examples II

Let

$$
\begin{aligned}
P_2 :\ & p(a,b) \\
& p(a,f(Y)) \leftarrow p(a,Y) \\
& q(f(X),Y) \leftarrow p(X,Y)
\end{aligned}
$$

Our analysis for $k = 4$:
$gfp(\mathcal{P}^4(P_2))(p(X,Y)) = \{\ <\{X/a, Y/f(f(f(W)))\}, \diamond>, <\_, \square>\}$
$gfp(\mathcal{P}^4(P_2))(q(X,Y)) = \{\ <\{X/f(a), Y/f(f(f(W)))\}, \diamond>, <\_, \square>\}$

$q(X,Y) \in Inf^4_{P_2}$ iff it unifies with $q(f(a), f(f(f(W))))$.

This allow us to prove that

1. $q(f(b), Y)$ terminates for any Y,
2. $q(f(a), Y)$ terminates only if Y is a ground depth-4 term.

Note:

1. our analysis allows us to prove that $q(f(b), Y)$ terminates for any Y,
2. anyway, using TermiLog or cTI, we can prove that $q(f(a), Y)$ terminates for a larger set of Y instances.

Safely replacing the breadth-first search

The search rules:

- breadth-first is complete but inefficient.
- depth-first is efficient but incomplete.

Using Inf_k^P, we define an analysis which allows us to safely replace the breadth-first with the depth-first rule.

Conclusions and Future Work

We have:

- introduced a semantic foundation for an abstract interpretation approach to termination,
- developed a new abstract domain useful for termination analysis.

We think that:

- most of the existing automatic methods can be reconstructed in this framework as abstractions of the "exact answers" semantics on suitable abstract domains,
- using the framework different abstractions can be combined together obtaining more precise results,
- abstract interpretation theory provides a rigorous theoretical background for combining domains and, therefore, analyses,
- the resulting method can be viewed as a theoretical basis for the design of a refined system able to analyze termination of real Prolog programs.