Extending the verification approach to finite failure

- a new method which uses different semantics and specifications

Motivations I

- Assume that the semantics of a program P is defined as least fixpoint of a continuous operator T.
  Let S be an interpretation which specifies the expected program semantics.
  - the program is *partially correct* w.r.t. T iff $\mathsf{lfp}(\mathsf{T}) \subseteq \mathsf{S}$.
  - a sufficient partial correctness condition is $\mathsf{T}(\mathsf{S}) \subseteq \mathsf{S}$.
- Several verification methods, based on semantics modeling different observable properties of logic programming as
  - the ground success set [Shapiro82],
  - the correct answers [Ferrand87],
  - computed answers and their abstractions [Comini et al.99]
- In [Levi et al.98] it has been showed that all the existing methods can be reconstructed as an instance of a general verification technique where the property one wants to verify is simply an abstract semantics on a suitable abstract domain.
- There is one interesting property, finite failure, which is not an abstraction of none of the semantics used above.

Motivations II

- Which semantics for finite failure?
  - The ground finite failure set $\mathsf{FF_P}$ is not correct w.r.t. finite failure.
  - The non ground finite failure set $\mathsf{NGFF_P}$ is correct, fully abstract, w.r.t. finite failure and it is AND- compositional [Gori et al.97].
- **The problem**: there exist no fixpoint characterization of $\mathsf{NGFF_P}$.
- **The idea**: use abstract interpretation to derive such fixpoint semantics.
  - start from a concrete traces semantics, which extends with infinite computations the concrete semantics of the Abstract Interpretation Framework [Comini et al.99]
  - define an abstract domain $\mathcal{S}$, chosen so as to model finite failure and to make the abstract operator $\mathsf{T_P^{ff}}$ precise
  - the corresponding abstract fixpoint semantics $\mathsf{lfp}(\mathsf{T_P^{ff}})$ is the Non-Ground Finite Failure set
  - it correctly models finite failure and is AND-compositional
- we can use the standard condition $\mathsf{T_P^{ff}}(\mathsf{S}) \sqsubseteq \mathsf{S}$ as a sufficient condition for the correctness w.r.t. finite failure
- we can define stronger conditions using Ferrand's approach, based on two specifications

The Semantic Domain

Let P be a program and R be the set of atoms which finitely fail in P. Then

- R is a downward closed set, i.e., if $A \in R \Rightarrow A\vartheta \in R$.
- *The key point:* R enjoys a kind of "upward closure" property.
  Example
  Assume $\{p(a), p(f(a)), p(f(f(X))), p(f(f(a))), \ldots\} \in R$.
  Which behavior for $p(X)$*?*

  - *Suppose* $p(X)$ *has a successful derivation.*
    $p(X) \xrightarrow[c_1]{\sigma_1} G_1 \xrightarrow[c_2]{\sigma_2}, \ldots, G_{n-1} \xrightarrow[c_n]{\sigma_n} \square$
    Let $\vartheta = \sigma_1 \cdot \ldots \cdot \sigma_n$.
    $\forall p(t) \in R,\ \nexists \delta = mgu(p(t), p(X)\vartheta)$, *otherwise* $p(t)\delta$

  - *Suppose* $p(X)$ *has an infinite derivation.*
    $p(X) \xrightarrow[c_1]{\sigma_1} G_1 \xrightarrow[c_2]{\sigma_2}, \ldots, G_{n-1} \xrightarrow[c_n]{\sigma_n} \ldots$
    Let $\vartheta_i = \sigma_1 \cdot \ldots \cdot \sigma_i$.
    $\forall p(t) \in R,\ \forall i\ \nexists \delta_i = mgu(p(t), p(X)\vartheta_i)$, *otherwise* $p(t)\delta_i$

$$\Downarrow$$

if $\forall$ *possible sequences* $\vartheta_1 :: \ldots :: \vartheta_n :: \ldots\ p(X)\vartheta_i \leq p(X)\vartheta_{i+1}$

$\exists p(t) \in R$, *s.t.* $\forall i\ \exists \delta_i = mgu(p(t), p(X)\vartheta_i)$,

then

$p(X) \in R$.

The Semantics Domain II

$$up^{ff}_{p(x)}(R) = R \cup \{p(x)\vartheta \mid \begin{array}{l} \text{for all (possibly infinite) sequences} \\ \vartheta_1 :: \ldots\ldots :: \vartheta_n :: \ldots, p(x)\vartheta_i \leq p(x)\vartheta_{i+1} \\ \exists p(t) \in R \text{ s.t.} \\ \forall\, i,\ p(t) \text{ unifies with } p(x)\vartheta\vartheta_i \end{array} \}.$$

$\cup_{p(x)} up^{ff}_{p(x)}$ is a closure operator.

$\mathcal{S}$ is the domain of downward closed sets of atoms, which are also closed w.r.t. $\cup_{p(x)} up^{ff}_{p(x)}$.

$(\mathcal{S}, \subseteq)$ is a complete lattice,

- the least upper bound of $R_1, R_2 \in \mathcal{S}$ is $\cup_{p(x)} up^{ff}_{p(x)}(R_1 \cup R_2)$
- the greatest lower bound of $R_1, R_2 \in \mathcal{S}$ is $(R_1 \cap R_2)$

$$
\begin{aligned}
T_P^{ff}(I) = \{\, p(\tilde{t}) \mid{} & \text{for every clause defining the procedure } p, \\
& p(t) :- B \in P \\
& p(\tilde{t}) \in up_{p(x)}^{ff}(Nunif_{p(x)}(p(t)) \cup \\
& \qquad \{p(t)\tilde{\vartheta} \mid \tilde{\vartheta} \text{ is a relevant for } p(t), \\
& B\tilde{\vartheta} \in up_B^{ff}(\{B\sigma \mid B = B_1, \ldots, B_n \; \exists B_i\sigma \in I\})\})
\end{aligned}
$$

- T_P^{ff} is continuous $\Rightarrow lfp(T_P^{ff}) = up_{p(x)}^{ff}(\cup_{i<\omega} T_P^{ff} \uparrow i)$

Example

P

$q(a) :- p(X)$
$p(f(X)) :- p(X)$

$$
\begin{aligned}
T_P^{ff} \uparrow 1 = \{ \; & q(f(X)), q(f(f(X))), \ldots \\
& q(f(a)), q(f(f(a))), \ldots \\
& p(a) \qquad \}
\end{aligned}
$$

$$
\begin{aligned}
T_P^{ff} \uparrow 2 = \; & T_P^{ff} \uparrow 1 \cup \{p(f(a))\} \\
& \vdots \\
T_P^{ff} \uparrow \omega = \; & T_P^{ff} \uparrow 2 \cup \{p(f(f(a))), p(f(f(f(a)))), \ldots\}
\end{aligned}
$$

$p(X) \notin up_{p(X)}^{ff}(T_P^{ff} \uparrow \omega)$ *since*

$\exists \vartheta_1 = \{X/f(Y)\} :: \vartheta_2 = \{X/f(f(Y))\} :: \vartheta_3 = \{X/f(f(f(Y)))\} :: \ldots,$

and $\forall p(t) \in T_P^{ff} \uparrow \omega \; \forall i \;\; \nexists \delta_i = mgu(p(t), p(X)\vartheta_i).$

$$\Downarrow$$

$$q(a) \notin T_P^{ff} \uparrow \omega + 1$$

Ferrand's approach

- Ferrand in [Ferrand93] uses the standard ground consequence operator $\mathsf{T_P}$
- The specifications are
  - S, intended $\mathsf{lfp}(\mathsf{T_P})$
  - $\mathsf{S'}$, intended $\mathsf{gfp}(\mathsf{T_P})$
- $\mathsf{lfp}(\mathsf{T_P}) \subseteq \mathsf{S}$.
  The standard sufficient condition for partial correctness $\mathsf{T_P}(\mathsf{S}) \subseteq \mathsf{S}$ allows us to reason about the ground success set
- $\mathsf{S'} \subseteq \mathsf{gfp}(\mathsf{T_P})$.
  The new sufficient condition $\mathsf{S'} \subseteq \mathsf{T_P}(\mathsf{S'})$ is somewhat related to missing answers

Verification conditions based on $\mathsf{T}_{\mathsf{P}}^{\mathsf{ff}}$

- $\mathsf{T}_{\mathsf{P}}^{\mathsf{ff}}$ is not co-continuous
  - this is also the case for Ferrand's T_{P}
- we replace $\mathsf{gfp}(\mathsf{T}_{\mathsf{P}}^{\mathsf{ff}})$ by $\mathsf{T}_{\mathsf{P}}^{\mathsf{ff}} \downarrow \omega$
  - we have proved that $\mathsf{T}_{\mathsf{P}}^{\mathsf{ff}} \downarrow \omega$ is the complement of the set of (possibly non-ground) atoms which have a successful derivation
- the standard verification condition
  - S is the intended set of (possibly non-ground) atoms which have a finite failure
  - correctness
    $\mathsf{lfp}(\mathsf{T}_{\mathsf{P}}^{\mathsf{ff}}) \subseteq \mathsf{S}$
  - sufficient condition for correctness
    $\mathsf{T}_{\mathsf{P}}^{\mathsf{ff}}(\mathsf{S}) \subseteq \mathsf{S}$
- the new verification condition
  - S' is the intended set of (possibly non-ground) atoms which do not have a successful derivation
  - correctness
    $\mathsf{S}' \subseteq \mathsf{T}_{\mathsf{P}}^{\mathsf{ff}} \downarrow \omega \Rightarrow \quad \mathsf{H}_{\mathsf{v}} \backslash \mathsf{T}_{\mathsf{P}}^{\mathsf{ff}} \downarrow \omega \subseteq \mathsf{H}_{\mathsf{v}} \backslash \mathsf{S}'$
  - sufficient condition for correctness
    $\mathsf{S}' \subseteq \mathsf{T}_{\mathsf{P}}^{\mathsf{ff}}(\mathsf{S}')$

Towards effective verification conditions

- the sufficient conditions $T_P^{ff}(S) \subseteq S$ and $S' \subseteq T_P^{ff}(S')$ are not effective because
  - T_P^{ff} is not finitary
  - both S and S' are infinite sets
- the analysis and verification of properties of finite failure, can be based on effective approximations of the operator T_P^{ff}
- since we have two semantics and two specifications, we can use two different (related) abstractions
  - an upward approximation (of the least fixpoint semantics)
  - a downward approximation (of $T_P^{ff} \downarrow \omega$)

The depth-k domain

- we define the function $\mathtt{depth}$ on terms, atoms and goal of a program.

$$|t| = \begin{cases} 1 & t \text{ is a constant or a variable} \\ max\{|t_1|, \ldots, |t_n|\} + 1 & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

The downward approximation

- $< \alpha^{bl}, \gamma^{bl} >$ is a *reversed* Galois insertion, i.e., $\alpha^{bl}(\cap X_i) = \cap(\alpha^{bl}(X_i))$.

- We can define the optimal abstract fixpoint operator $T_P^{ff^{bl}}$ on D^{bl}.

Example

$$\begin{array}{l} P \\ q(a) :- p(X) \\ p(f(X)) :- p(X) \end{array}$$

for $k = 3$,

$lfp(T_P^{ff^{bl}}) = \{q(f(f(X))), q(f(f(a))), q(f(X)), q(f(a)), p(a), p(f(a)), p(f(f(a)))\}$

$\gamma^{bl}(lfp(T_P^{ff^{bl}})) \subseteq lfp(T_P^{ff})$

The upward approximation

- $< \alpha^{up}, \gamma^{up} >$ is a Galois insertion.

- We can define the optimal abstract fixpoint operator $T_P^{ff^{up}}$ on D^{up}.

Example

$$
\begin{array}{l}
P \\
q(a) :- p(X) \\
p(f(X)) :- p(X)
\end{array}
$$

for $k = 3$,

$$
\begin{aligned}
lfp(T_P^{ff^{up}}) = \{ \; & q(f(f(K))), q(f(X))\{X/f(X)\}, q(f(f(a))), q(f(X)), q(f(a)), \\
& p(a), p(f(a)), p(f(f(a))), p(f(f(K)))\}
\end{aligned}
$$

$$
lfp(T_P^{ff}) \subseteq \gamma^{up}(lfp(T_P^{ff^{up}}))
$$

$\mathsf{depth}-\mathsf{k}$ **correctness and sufficient conditions**

- the two abstractions are used to get finite approximations of the Non-Ground Finite Failure set and of the complement of the success set.
- the specifications
  - $S_{\alpha^{up}}$ is the α^{up} abstraction of the intended Non-Ground Finite Failure set.
  - $S'_{\alpha^{bl}}$ is the α^{bl} abstraction of the intended set of atoms which either finitely fail or (universally) do not terminate.
    * the complement of the set of atoms (of depth $\leq$ k) which have a successful derivation.
- a program P is $\mathsf{depth}-\mathsf{k}$ correct if

c1 $\alpha^{up}(lfp(T_P^{ff})) \subseteq S_{\alpha^{up}}$.

c2 $S'_{\alpha^{bl}} \subseteq \alpha^{bl}(T_P^{ff} \downarrow \omega)$.

- sufficient conditions for the $\mathsf{depth}-\mathsf{k}$ correctness

sc1 $T_P^{ff^{up}}(S_{\alpha^{up}}) \subseteq S_{\alpha^{up}}$.

sc2 $S'_{\alpha^{bl}} \subseteq T_P^{ff^{bl}}(S'_{\alpha^{bl}})$.

Examples I

- **Example 1**

$$\begin{array}{ll} P_1: & \mathtt{append}([\,], X, X) :- \mathtt{list}([X]) \quad \textit{instead of} \; \mathtt{append}([\,], X, X) :- \mathtt{list}(X) \\ & \mathtt{append}([X|Y], Z, T) :- \mathtt{append}(Y, Z, [X|T]). \\ & \mathtt{list}([\,]). \\ & \mathtt{list}([X|Y]) :- \mathtt{list}(Y). \end{array}$$

  - The program is *not* correct w.r.t. the intended depth-k success set.
    We can detect this error.
    $\mathtt{append}([\,], a, a) \in S'_{\alpha^{bl}}$ yet $\mathtt{append}([\,], a, a) \notin T_{P_1}^{ff^{bl}}(S'_{\alpha^{bl}})$.
    Therefore $\mathbf{sc}_2$ does not hold.

- **Example 2**

$$\begin{array}{ll} P_2: & \mathtt{append}([X|Y], Z, T) :- \mathtt{append}(Y, Z, [X|T]). \\ & \mathtt{list}([\,]). \\ & \mathtt{list}([X|Y]) :- \mathtt{list}(Y). \end{array}$$

  - The program is *not* correct w.r.t. the intended depth-k finite failure set.
    We can detect this error.
    $\mathtt{append}([\,], [a], [a]) \in T_{P_2}^{ff^{up}}(S_{\alpha^{up}})$, yet
    $\mathtt{append}([\,], [a], [a]) \notin S_{\alpha^{up}}$.
    Therefore $\mathbf{sc}_1$ does not hold.

Examples II

- **Example 3**

$$\begin{aligned} P_3: \; & \mathtt{append}([\,], X, X) :- \mathtt{list}(X). \\ & \mathtt{append}([X|Y], Z, T) :- \mathtt{append}(Y, Z, [X|T]). \\ & \mathtt{list}([\,]). \\ & \mathtt{list}([X|Y]) :- \mathtt{list}(Y). \end{aligned}$$

  – $\mathbf{sc}_1$ holds.
    $\Downarrow$
    The program is correct w.r.t. the intended depth-k finite failure set.

  – $\mathbf{sc}_2$ holds.
    $\Downarrow$
    The program is correct w.r.t. the intended depth-k successful set.

Future Work

- how to extend the approach to other abstract domains which might be useful for reasoning about finite failure (e.g. assertions).