Programmazione logica

Analisi statica

- l'interpretazione astratta è molto popolare in programmazione logica
  - Il modello computazionale si presta a varie ottimizzazioni, basate sui risultati dell'analisi
  - è relativamente facile da definire, poiché le semantiche sono naturalmente collecting ed il dominio concreto (insiemi di sostituzioni o insiemi di insiemi di equazioni) è piuttosto semplice
- esistono molte importanti proprietà (groundness, freeness, sharing, *depth(k)*)
  - per alcune proprietà (groundness e sharing) esisono molti domini astratti differenti
    - tecniche per confrontare la precisione relativa di domini astrattiimportantresults on techniques for the systematic designdomains, which can probably be applied to other paradigms as well
  - compilazione astratta in CLP (Giacobazzi, Debray & Levi, JLP 95)
    - la computazione astratta è una computazione CLP standard su un differente sistema di vincoli

Programmazione logica

Semantica comparativa

$$\alpha(\mathrm{lfp}\ \mathcal{F}) = \mathrm{lfp}\ \mathcal{F}^{\alpha}$$

- nessuno dei due punti fissi è calcolabile in un numero finito di passi
- utile per ragionare su differenti semantiche e per derivare sistematicmente semantiche più astratte
  - **scelta della semantica più adeguata alla particolare analisi e/o verifica**
- $\mathcal{F}^{\alpha}$ è meno costoso di $\mathcal{F}$ nel calcolo della proprietà osservabile modellata da α
  - no junk

- ricostruzione sistematica di diverse semantiche punto fisso
  (Comini, Levi & Meo, Info. & Comp. 00)

Groundness in Programmazione logica

Analisi di groundness

- versione CLP
- dominio concreto
  - $(\mathcal{P}(\mathcal{E}qns), \subseteq)$, insiemi di insiemi di equazioni in forma risolta
- semantica concreta
  - la versione CLP della s-semantica (vincoli di risposta)
- 3 domini astratti
  - $\mathcal{G}$: the property of being ground
  - $\mathcal{DEF}$: dipendenze funzionali di groundness
  - $\mathcal{POS}$: $\mathcal{DEF}$ + un po' di informazione disgiuntiva
    - i reticoli sono mostrati nel caso di 2 variabili

Un esempio

- programma

  p(X,Y) :- X=a.
  p(X,Y) :- Y=b.
  q(X,Y) :- X=Y.
  r(X,Y) :- p(X,Y),q(X,Y).

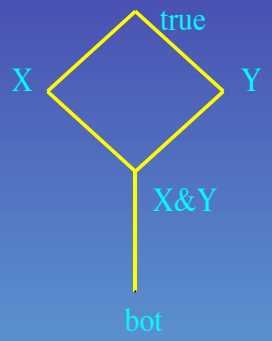
- semantica concreta

  p(X,Y) -> {{X=a},{Y=b}}
  q(X,Y) -> {{X=Y}}
  r(X,Y) -> {{X=a,Y=a},{X=b,Y=b}}

- nella semantica concreta di r
  - entrambi gli argomenti sono legati a terminiground (in tutti i vincoli di risposta)

Il dominio $\mathcal{G}$

Analisi di groundness



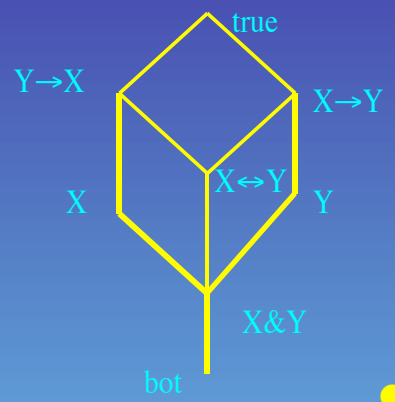
$\gamma_{\mathcal{G}}$ (v) =

- $\varnothing$, se v= bot
- {e $\in$ *Eqns* | X è legato ad un termine ground in e }, se v= X — X è sempre ground
- *Eqns*, se v= true — nessuna informazione di groundness

- programma
  p(X,Y) :- X=a.
  p(X,Y) :- Y=b.
  q(X,Y) :- X=Y.
  r(X,Y) :- p(X,Y),q(X,Y).
- semantica concreta
  p(X,Y) -> {{X=a},{Y=b}}
  q(X,Y) -> {{X=Y}}
  r(X,Y) -> {{X=a,Y=a},{X=b,Y=b}}
- astrazione della semantica concreta
  p(X,Y) -> true
  q(X,Y) -> true
  r(X,Y) -> X & Y
- Programma astratto
  p(X,Y) :- $lub_{\mathcal{G}}$ (X,Y).
  q(X,Y) :- true.
  r(X,Y) :- $glb_{\mathcal{G}}$ (p(X,Y),q(X,Y)).
- Semantica astratta
  p(X,Y) -> true
  q(X,Y) -> true
  r(X,Y) -> true

Il dominio $\mathcal{D}ef$

Analisi di groundness



$\gamma_{\mathcal{D}ef}$ (v) =

- {e ∈ $\mathcal{E}qns$ | X = Y ∈ e},
  se v= X ↔ Y X is ground if and only if Y is ground
- {e ∈ $\mathcal{E}qns$ | X = t ∈ e e Y occorre in t},
  se v= X → Y if X is ground then Y is ground
- .....

- programma
  p(X,Y) :- X=a.
  p(X,Y) :- Y=b.
  q(X,Y) :- X=Y.
  r(X,Y) :- p(X,Y),q(X,Y).

- semantica concreta
  p(X,Y) -> {{X=a},{Y=b}}
  q(X,Y) -> {{X=Y}}
  r(X,Y) -> {{X=a,Y=a},{X=b,Y=b}}

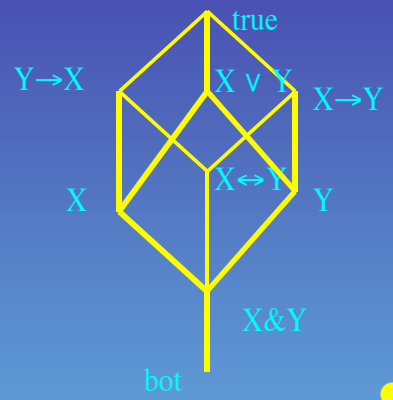
- astrazione della semantica concreta
  p(X,Y) -> true
  q(X,Y) -> X ↔ Y
  r(X,Y) -> X & Y

- programma astratto
  p(X,Y) :- $lub_{\mathcal{D}ef}$ (X,Y).
  q(X,Y) :- X ↔ Y.
  r(X,Y) :- $glb_{\mathcal{D}ef}$ (p(X,Y),q(X,Y)).

- semantica astratta
  p(X,Y) -> true
  q(X,Y) -> X ↔Y
  r(X,Y) -> X ↔Y

Il dominio *Pos*

Analisi di groundness



γ_{pos} (v) =

- {e $\in$ *Eqns* | X oppure Y è legato ad un termine ground in e }, if v= X v Y either X or Y is ground
- ....

- programma
  p(X,Y) :- X=a.
  p(X,Y) :- Y=b.
  q(X,Y) :- X=Y.
  r(X,Y) :- p(X,Y),q(X,Y).

- semantica concreta
  p(X,Y) -> {{X=a},{Y=b}}
  q(X,Y) -> {{X=Y}}
  r(X,Y) -> {{X=a,Y=a},{X=b,Y=b}}

- astrazione della semantica concreta
  p(X,Y) -> X v Y
  q(X,Y) -> X ↔ Y
  r(X,Y) -> X & Y

- programma astratto
  p(X,Y) :- lub_{pos} (X,Y).
  q(X,Y) :- X ↔ Y.
  r(X,Y) :- glb_{pos} (p(X,Y),q(X,Y)).

- semantica astratta
  p(X,Y) -> X v Y
  q(X,Y) -> X ↔Y
  r(X,Y) -> X & Y

Verifica di programmi con Interpretazione Astratta

- $\mathcal{F}$ = funzione di valutazione semantica concreta
  - abbastanza concreta per poter osservare la proprietà
  - la proprietà è modellata da un dominio astratto $(\mathcal{A}, \leq)$ e da una inserzione di Galois α, γ
- $\mathcal{F}^\alpha$ = funzione di valutazione semantica astratta
- $\mathcal{S}^\alpha$ = specifica della proprietà, cioè astrazione della semantica concreta intesa
- correttezza parziale: $\alpha(\text{lfp } \mathcal{F}) \leq \mathcal{S}^\alpha$
- condizione sufficiente per la correttezza parziale:

$$\mathcal{F}^\alpha(\mathcal{S}^\alpha) \leq \mathcal{S}^\alpha$$

- (Comini, Levi, Meo & Vitiello, JLP 99)
  - se $\mathcal{F}^\alpha(\mathcal{S}^\alpha) \leq \mathcal{S}^\alpha$
  - allora $\mathcal{S}^\alpha$ è un pre-puntofisso di $\mathcal{F}^\alpha$
  - quindi

$$\alpha(\text{lfp } \mathcal{F}) \leq \text{lfp } \mathcal{F}^\alpha \leq \mathcal{S}^\alpha$$

Analisi e Verifica

- $\mathcal{F}$ = funzione di valutazione semantica concreta
- $\mathcal{F}^\alpha$ = funzione di valutazione semantica astratta

- analisi: calcola lfp $\mathcal{F}^\alpha$
  - dobbiamo calcolare un punto fissont
  - dominio noetheriano oppure widening

- $\mathcal{S}^\alpha$ = specifica della proprietà
- verifica: dimostra $\mathcal{F}^\alpha(\mathcal{S}^\alpha) \leq \mathcal{S}^\alpha$
  - niente calcolo di punto fisso, nessuna necessità di domini noetheriani
  - rappresentazione finita della specifica
  - decidibilità di $\leq$

Progetto sistematico di domini astratti

- una volta che abbiamo il dominio astratto, il progetto della semantica astratta è sistematico
- l'interpretazione astratta fornisce anche risultati che possono essere sfruttati per rendere (più) sistematico il progetto di domini astratti
  - per confrontare e combinare domini
  - per raffinare domini con il fine di migliorarne la precisione
- prodotto ridotto (dei domini $\mathcal{A}$ e $\mathcal{B}$)
  - permette di analizzare (insieme) le proprietà modellate da $\mathcal{A}$ e $\mathcal{B}$
  - spesso fornisce risultati migliori delle due analisi separate
    - per l'interazione fra domini

lifting al powerset (e completamento disgiuntivo)

- più o meno, trasforma $\mathcal{A}$ in $\mathcal{P}(\mathcal{A})$
- maggiore precisione
  - nessuna perdita di informazione nel calcolo dei lub's

Operazioni sui Domini Astratti

Progetto di domini

- molti utili operatori su domini astratti (raffinamenti)
  - un survey in (File', Giacobazzi & Ranzato, ACM Comput. Surv. 96)
- Completamento lineare
  (Giacobazzi, Ranzato & Scozzari, SAS 98)
  - dipendenze funzionali modellate dall'implicazione di logica lineare
- ricostruzione di tutti I domini noti per l'analisi di groundness (Scozzari, SAS 97)
  - $\mathcal{DEF} = \mathcal{G} \rightarrow \mathcal{G}$
  - $\mathcal{POS} = \mathcal{DEF} \rightarrow \mathcal{DEF}$
  - $\mathcal{POS} = \mathcal{POS} \rightarrow \mathcal{POS}$
    - ottimalità di $\mathcal{POS}$

poi applicata con successo ad altri domini in programmazione logica

- tipi (Levi & Spoto, PLILP 98)
- sharing e freeness
  (Levi & Spoto, PEPM 00)

problemi aperti

- gli stessi raffinamenti si applicano anche ad altri paradigmi o ai sistemi di tipo?

Interpretazione astratta

- una teoria matematica semplice e solida su cui basare
  - semantica comparativa
  - analisi statica
  - verifica
  - una metodologia per la derivazione sistematica di
  - domini astratti a partire dalla proprietà
  - semantica astratta a partire dalla semantica concreta e dal dominio astratto