Lezione 9

Come ragionare sulla correttezza
correttezza positiva, copertura, modi e tipi

A. Richiamo: correttezza positiva

Correttezza positiva. Riguarda il legame fra semantica interna e semantica esterna ("user semantics"):

- *per ogni **risposta calcolata** σ (tale che KB |– $Q\sigma$), $Q\sigma$ è (universalmente) vera in ogni interpretazione esterna*

Teorema. Se le chiusure universali delle causole di KB sono vere nelle interpretazioni esterne, vale la correttezza.

Dim. Segue dal teorema di validità: KB |– $Q\sigma$ implica vera in tutti i modelli, compresi quelli che rappresentano le interpretazioni esterne

Nota critica. La corrispondenza interpretazione esterna/interna non è trattabile formalmente: test empirico ineliminabile

Esempio

Programma per leq(X,Y)

leq(0,X).
leq(s(X),s(Y)) :- leq(X,Y).

Interpretazione esterna ***N***

- 0, s hanno la solita interpretazione
  - si ricordi: ***numerali*** = {0, s(0), s(s(0)), ...}
- leq(X,Y) significa $X \leq Y$ sui naturali

- Correttezza. Dimostriamo
  - Verità in ***N*** di leq(0,X):
    $0 \leq X$ vera per ogni naturale X
  - Verità in ***N*** di leq(s(X),s(Y)) :- leq(X,Y):
    $(X+1) \leq (Y+1) \leftarrow X \leq Y$ vera per ogni X, Y:naturali

Richiamo: condizione di copertura

- Copertura: *per ogni query Q e risposta ground $Q\alpha$ vera nella semantica esterna esiste una risposta calcolata σ tale che α è un caso particolare di σ (cioè $\alpha = \sigma\delta$)*
  - la copertura garantisce che anche le risposte negative sono corrette; ricordiamo che se Q ha risposta NO
    - vale la negazione di Q se vale l'ipotesi di mondo chiuso rispetto alla semantica esterna
    - NO significa ignoranza in caso contrario

Generalmente la copertura riguarda un sottoinsieme di *query di interesse*: per ogni query Q di interesse ....

- **Esempio:** interessati a query della forma leq(X,*numerale*)
  - una query ?leq(X, s(s(0)))
  - ci aspettiamo le risposte X = 0, X = s(0), X = s(s(0))

COME VERIFICARE LA COPERTURA.

a) Per ogni query ?Q di interesse deve valere la terminazione

b) per quelle vere si deve avere risposta positiva

- La condizione b) sarà riesaminata introducendo il completamento di Clark. Qui:
  - consideriamo le *prove di terminazione basate s sui modi di input e di output e sull'individuazione di relazioni ben fondate*
    - relazioni binarie prive di catene infinite
  - assumendo la terminazione, mostriamo la copertura ragionando *per induzione sulla profondità degli alberi di prova.*

Modi e tipi

- Consideriamo i modi di input e output tipati:
  - in:T ingresso ground di tipo T
  - out:T uscita con risposta ground di tipo T
  - any:T uscita eventualmente non ground
- dove un tipo T è un insieme di termini

Esempio

- Tipo Nat:
  - termini generati da variabili, costante 0 e costruttore unario s; ad esempio:
  - es: s(0) ground, s(s(X)) aperto

Esempio di specifica con modi e tipi

Specifica: **modi** sum(in:Nat,in:Nat,out!:Nat);
sum(out:Nat,out:Nat,in:Nat);
sum(any:Nat,in:Nat,any:Nat);

interpretazione:
Nat rappresenta i naturali
sum(x,y,z) significa z = x+y;

Implementazione:

```
sum(X,0,X).
sum(X,s(Y),s(Z)) :- sum(X,Y,Z).
```

NOTA. Con out! Indichiamo che esiste una e una sola risposta.

A) *I modi indicano le query di interesse*:
sum(in:Nat,in:Nat,out!:Nat) rappresenta l'insieme di query sum(T_1,T_2,T_3) dove T_1:Nat, T_2:Nat sono ground e T_3 è un termine arbitrario di Nat, ad esempio:
sum(s(0),s(s(s(0))),s(Y)).

B) *I modi indicano il tipo delle risposte attese:*
sum(in:Nat,in:Nat,out!:Nat) indica che per la query sum(s(0),s(s(s(0))),s(Y)) ci attendiamo un'unica risposta:
Y = g, con g:Nat ground

Esercizio: quali sono le query e le risposte attese per
sum(out:Nat,out:Nat,in:Nat),
sum(any:Nat,in:Nat,any:Nat)?

Modi e tipi nella verifica di correttezza

Data una specifica con modi, tipi e interpretazione esterna *E*, una verifica di correttezza comprende:

- Verifica della correttezza positiva (verità delle clausole in *E*)
  - già visto
- Verifica dei modi
- Verifica di terminazione
- Verifica dei tipi e della copertura
  - *Un programma è corretto rispetto ai tipi dichiarati se per ogni query Q che rispetti i tipi, ogni (eventuale) risposta Qσ rispetta i tipi*

Verifica dei modi(*): modi ben dati

in out
Atomo(X..... T(X))

(*) in versione "rozza",
da usare come metodo
di ragionamento;
c'è una vasta letteratura

out out
Testa(X ... Y) :- Corpo(X ... Y)
in in

Testa(..) :- ... B(.. X ..), .., C(.. X ..) ..
out in

Esempio

sum(in:Nat,in:Nat,out:Nat) "con modi ben dati" nel programma:

in out
sum(X , 0, X).

out out
sum(X , s(Y), s(Z)) :- sum(X , Y , Z).
in in in in

Induzione: la chiamata ricorsiva
è consistente rispetto ai modi

Verifica di terminazione: il caso di ricorsione su argomenti con modo in

- Consideriamo il caso più semplice e comune:
  - ricorsione su $t_1,..,t_n$ con modo in
  - si può definire una dimensione $dim(t_1,..,_{tn}) >= 0$
- Accenniamo all'estensione a casi più generali, che verranno esaminati solo tramite esempi

$p(.. t_1,..,t_n ..) :- .... , p(.. r_1,..,r_n ..), ...$

$dim(r_1,..,r_n) < dim(t_1,..,t_n)$

Estensioni: si ha una relazione ben fondata $rb((r_1,..,r_n),(t_1,...,t_n))$
Estensioni: con mutua ricorsione si esamineranno alberi di prova incompleti

Esempio

Terminazione di sum(in:Nat,in:Nat,out:Nat) con ricorsione sum(..,s(t),..) :- sum(..,t,..)

sum(X,s(Y),s(Z)) :- sum(X,Y,Z).

Definiamo $dim(s^n(0)) = n$
per ogni Y:Nat ground: $dim(Y) < dim(s(Y))$

Verifica dei tipi e della copertura: ragionare per induzione sulla profondità degli alberi di prova

- Per copertura e correttezza rispetto ai tipi, si può ragionare per induzione sulla profondità degli alberi di prova:
  - **Base:** per tutte le query di interesse che falliscono immediatamente o terminano con un albero di profondità 1 vale la proprietà da dim.
  - **Passo:** se una query di interesse avvia un albero di prova di profondità > 1, *per ogni possibile* query e passo di ricerca per ogni possibile query e passo di ricerca di un albero di prova:
    - si esegue il passo di ricerca,
    - si assume la proprietà da dimostrare per le query di livello superiore
    - la si dimostra per la radice.

Esempio.

Consideriamo una generica query prevista da sum(in:Nat,in:Nat,out!:Nat);dimostriamo che rispetta i tipi.
Le query di interesse sono
sum(t,t',T), con t,t':Nat ground e T:Nat qualsiasi

Base: le sole query base sono sum(t,0,T)

T unifica con t: $\overline{\text{sum(t,0,t)}}$ con t:Nat (rispetta i tipi)
T non unifica con t, fail
I tipi sono rispettati nel caso base.

Passo: le sole query passo sono sum(t,s(t'),T):

$$\frac{\text{sum}(t, t',T)}{\text{sum}(t,s(t'),s(T))}$$

Per ipotesi induttiva, sum(t,t',T) fallisce o termina con risposta sum(t,t',t"), dove t" : Nat;
quindi si otterrà, nella radice: sum(t,s(t'),s(t"))
Siccome t":Nat, anche s(t"):Nat
--(CVD passo)--

Esempio/Esercizio

- Per dimostrare la copertura bisogna dimostrare che per ogni query di interesse sum(t,t',T):
  se vi è una risposta sum(t,t',t") vera nell'interpretazione esterna dei naturali,
  allora vi è un albero di prova non fallito con radice sum(t,t',t").
- Si procede con base e passo come prima;
  farlo per esercizio (vedremo alla lavagna)

Esercizio

Eseguire il programma sotto riportato con le query:

prod(s(0),s(s(0)),X)	pp(s(0),s(s(0)),X)	pq (s(0),s(s(0)),X)
prod(s(0),X,s(s(0)))	pp(s(0),X,s(s(0)))	pq(s(0),X,s(s(0)))
prod(X,s(s(0)),s(s(s(0))))	pp(X,s(s(0)),s(s(s(0))))	pq(X,s(s(0)),s(s(s(0))))

spiegare le differenze analizzando i modi

```
sum(X,0,X).
sum(X,s(Y),s(Z)) :- sum(X,Y,Z).
prod(_,0,0).
prod(X,s(Y),Z) :- sum(V,X,Z), prod(X,Y,V).
pp(_,0,0).
pp(X,s(Y),Z) :- sum(X,V,Z), prod(X,Y,V).
pq(_,0,0).
pq(X,s(Y),Z) :- pq(X,Y,V), sum(V,X,Z).
```

Esercizio

- Determinare tutti i cammini lunghi n in un grafo. Il grafo sia rappresentato dagli archi e la lunghezza dai naturali 0, s(0), ….;
- dividere la soluzione in due parti, una generale (definizione dei cammini con lunghezza in un grafo) e una particolare (il grafo specifico);
- provare la parte generale con diversi grafi specifici.
- NB. Pensare sempre in termini correttezza positiva e copertura, individuando i modi e i tipi.