Sviluppo di Software Sicuro - S^3
Analisi dei flussi informativi

Corso di Laurea Magistrale in
Sicurezza Informatica: Infrastrutture e Applicazioni
Università di Pisa – Polo di La Spezia
C. Montangero
Anno accademico 2009/10

Sommario

- Concetti introduttivi
- Comandi elementari
- Comandi composti
- Stabilità e terminazione
- Conclusioni

S^3 2009/10 – Flussi informativi

INTRODUZIONE

Contesto

- Dato un blocco di comandi S
  - *contesto* = elenco alfabetico delle variabili usate
- I flussi informativi vengono *calcolati*
  - grazie a composizione di matrici booleane
    - 1 per vero, 0 per falso
  - indicizzate dalle variabili nel contesto
- A partire dai comandi elementari
- Convenzioni
  - Variabili di programma: maiuscole: A, B, ... X, W,...
  - Meta variabili: minuscole: v, t, ...

Esempio

S: `if W>0 then X:=Y+Z else Z:=2 end if;`

- Ci interessa sapere:
  - cosa *può* essere modificato
  - cosa *può* essere conservato
  - cosa *può* influenzare che cosa (derives)
- Per esempio, in S:
  - X può essere modificato
  - X può essere conservato
  - X può dipendere da Y

Matrice delle *definizioni* D^S

- Matrice diagonale
- $D^S_{v,v}$ = 1 sse S *può* definire v
- Esempio

S': `if W>0 then X:=Y+Z else Z:=2 end if;`

$D^{S'}$ =

	X	Y	Z	W
X	1	0	0	0
Y	0	0	0	0
Z	0	0	0	0
W	0	0	0	1

Matrice delle var. *preservate* P^S

- Matrice diagonale
- $P^S_{v,v}$ = 1 sse s *può* preservare il *valore iniziale* di v
- Esempio

```
S': if W>0 then X:=Y+Z else Z:=2 end if;
```

$P^{S'}$ =

	X	Y	Z	W
X	1	0	0	0
Y	0	1	0	0
Z	0	0	1	0
W	0	0	0	1

S3: SPARK - C.Montangero - Copyright 2010 7

Matrice delle *ridefinizioni* R^S

- Non diagonale, in generale
- $R^S_{v,t}$ = 1 sse il valore *finale* di t *può* dipendere dal *valore iniziale* di v

Esempio

```
S': if W>0 then X:=Y+Z else Z:=2 end if;
```

$R^{S'}$ =

	X	Y	Z	W
X	1	0	0	0
Y	1	1	0	0
Z	1	0	1	0
W	1	0	1	1

S3: SPARK - C.Montangero - Copyright 2010 8

Preliminari

- Siano:
  - I la matrice identità (1 nella diagonale)
  - O la matrice nulla (tutti 0)
- Le operazioni logiche or, and vengono estese puntualmente alle matrici
- Con MN indichiamo il prodotto righe x colonne
  - or = somma, and = prodotto
  - $(MN)_{ij} = or_k (M_{ik}$ and $M_{kj})$
  - $(MN)_{ij}$ = 1 sse esiste almeno una coppia di 1 con lo stesso indice nella riga e nella colonna

S3: SPARK - C.Montangero - Copyright 2010 9

Approccio

- Comandi elementari
  - null, assegnamento, chiamata procedura
  - matrici definite in base alle semantica dei comandi
    - In HIS, R è definita in funzione di matrici ausiliarie, ottenibili per analisi sintattica delle espressioni
- Comandi composti
  - approccio composizionale
  - matrici definite in funzione delle matrici delle parti

S³ 2009/10 – Flussi informativi

COMANDI ELEMENTARI

Comandi elementari: null

- Possiamo porre:

- $D^{null} = \mathbb{O}$: null non modifica nulla
- $P^{null} = \mathbb{I}$: null non modifica nulla
- $R^{null} = \mathbb{I}$: null non modifica nulla
  - il valore finale di ogni variabile (può) dipende(re) dal suo valore iniziale

Comandi elementari: procedure

- Possiamo sfruttare le annotazioni

```
procedure P(X : out Float; I: in Integer )
--# global in out Z;
--# derives X from I & Z from *,I;
```

- La dichiarazione definisce le matrici per P:
- Il contesto è dato dai parametri e dalle globali
- NB: le globali stanno anche nel contesto del chiamante

```
D  I X Z     P  I X Z     R  I X Z
I  0         I  1         I  1 1 1
X     1      X     0      X
Z        1   Z        1   Z        1
```


Comandi elementari: procedure (2)

- La chiamata associa argomenti e parametri
- Assumiamo {X,Y,Z,W} come contesto del chiamante

```
P(X,Y)
```

- Nelle tabelle, le colonne relative agli argomenti out vengono sostituite da quelle dei corrispondenti parametri
- Per le altre: solo gli out sono definiti, e quindi
  - in D, tutti 0
  - in P, tutti 1
  - in R, 1 sulla diagonale (valore finale = valore iniziale)
- non le formalizzeremo...

Esempio

```
D  I X Z     P  I X Z     R  I X Z
I  0         I  1         I  1 1 1
X     1      X     0      X
Z        1   Z        1   Z        1
```

matrici procedura

```
     X I          X I          X I
D    X Y Z W   P  X Y Z W   R  X Y Z W
X X  1         X  0         X
I Y     0      Y     1      Y  1 1 1
  Z        1   Z        1   Z        1
  W           0 W           1 W           1
```

matrici chiamata

Assegnamento

v := e(Σ) , dove Σ è l'insieme delle variabili che appaiono in e

- $D^{v := e(\Sigma)}_{i,i} = 1$ sse i=v
- $P^{v := e(\Sigma)}_{i,i} = 1$ sse i≠v
- $R^{v := e(\Sigma)}_{i,j} = 1$ sse i=v e j in Σ

- Σ comprende gli argomenti di eventuali chiamate di funzioni
  - no effetti collaterali

Assegnamento: esempio

S1: X:=Y+Z Σ = {Y, Z}

- $D^{X:=Y+Z}_{i,i} = 1$ sse i=X
- $P^{X:=Y+Z}_{i,i} = 1$ sse i≠X
- $R^{X:=Y+Z}_{i,j} = 1$ sse j=X e i in {Y, Z} o $P_{i,j} = 1$

D	X	Y	Z	W
X	1			
Y				
Z				
W				

P	X	Y	Z	W
X				
Y		1		
Z			1	
W				1

R	X	Y	Z	W
X				
Y	1	1		
Z	1		1	
W				1

Assegnamento: esempio

S2: W:=2*X Σ = {X}

- $D^{W:=2*X}_{i,i} = 1$ sse i=W
- $P^{W:=2*X}_{i,i} = 1$ sse i≠W
- $R^{W:=2*X}_{i,j} = 1$ sse j=W e i=X

D	X	Y	Z	W
X				
Y				
Z				
W		1		

P	X	Y	Z	W
X	1			
Y		1		
Z			1	
W				

R	X	Y	Z	W
X	1			1
Y		1		
Z			1	
W				

S3 2009/10 – Flussi informativi

COMANDI COMPOSTI

Sequenza

S' S''

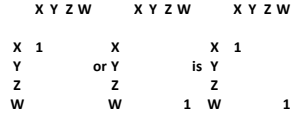
- Si compongono le matrici dei singoli blocchi

- $D^{S' S''} = D^{S'}$ or $D^{S''}$
- $P^{S' S''} = P^{S'}$ and $P^{S''}$
- $R^{S' S''} = R^{S'} R^{S''}$, infatti
  - $R^{S' S''}{}_{i,j} = \text{or}_k\ (R^{S'}{}_{i,k}$ and $R^{S''}{}_{k,j})$, infatti
  - $R^{S' S''}{}_{i,j} = 1$ sse esiste k tale che
    - j può dipendere da k, che può dipendere da i
  - ergo, sse j può dipendere da i

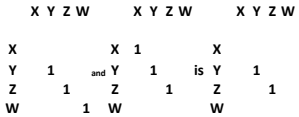
Sequenza: esempio

S = S1 S2 = X:=Y+Z; W:=2*X;

- $D^S = D^{S1}$ or D^{S2}



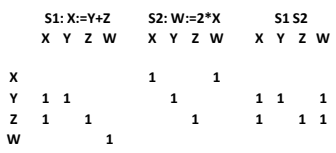
- $P^S = P^{S1}$ and P^{S2}



Sequenza: esempio

S = S1 S2 = X:=Y+Z; W:=2*X;

- $R^S = R^{S1}\ R^{S2}$



S3: SPARK - C.Montangero - Copyright 2010 22

Condizionale

C = if e(Σ) then S' else S'' end if;

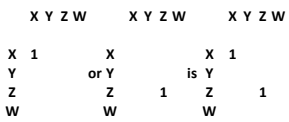
- $D^C = D^{S'}$ or $D^{S''}$
- $P^C = P^{S'}$ or $P^{S''}$
- Si deve tener conto della dipendenza da *e*
  - tutte le variabili definite in C dipendono anche dalle variabili in *e*
- $R = R^{e(\Sigma)}$ or $R^{S'}$ or $R^{S''}$, dove
  - $R^{e(\Sigma)}_{i,j} = 1$ sse j in D^C e i in Σ

S3: SPARK - C.Montangero - Copyright 2010 23

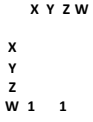
Condizionale : esempio

C': `if W>0 then X:=Y+Z else Z:=2 end if;`

- $D^{C'} = D^{X:=Y+Z}$ or $D^{Z:=2}$



- $R^{W>0}$



S3: SPARK - C.Montangero - Copyright 2010 24

Condizionale : esempio (2)

C': **if** W>0 **then** X:=Y+Z **else** Z:=2 **end if;**

- $R' = R^{X:=Y+Z}$ or $R^{Z:=2}$

- $R^{C'} = R^{e(\Sigma)}$ or R'

- Esercizio: if then

Cicli

W = while e(Σ) loop S end loop;

- $D^W = D^S$ (se si entra)
- $P^W = \mathbb{I}$ (se non si entra)

- R si ottiene per sviluppo del while con if then:

W = if e(Σ) then S W end if;

R in un ciclo

W = while e(Σ) loop S end loop;

W = if e(Σ) then S W end if;

$R^W = R^{e(\Sigma)}$ or $\mathbb{I}$ or $R^S R^W$
$= \mathbb{I}(R^{e(\Sigma)}$ or $\mathbb{I})$ or $R^S (R^{e(\Sigma)}$ or $\mathbb{I}$ or $R^S R^W)$
$= \mathbb{I}(R^{e(\Sigma)}$ or $\mathbb{I})$ or $R^S (R^{e(\Sigma)}$ or $\mathbb{I})$ or $(R^S)^2 R^W$
$= \ldots$
$= (R^{e(\Sigma)}$ or $\mathbb{I})$ $(\mathbb{I}$ or R^S or $(R^S)^2$ or ... $)$
$= (R^{e(\Sigma)}$ or $\mathbb{I})$ $(R^S)^*$

dove $(R^S)^*$ è la chiusura riflessiva e transitiva di R^S

e allora?

- La chiusura riflessiva e transitiva di R
  - si calcola in $O(n^3)$ – Teorema di Warshall – n variabili
- Esempio: GCD

```
while D/=0 loop R:=C rem D; C:=D; D:=R; end loop;
```

- la var R può dipendere dal suo valore iniziale
  - se non si entra
- C pure, se si entra,
  - tramite D, dal secondo giro in poi
  - D = C rem D

	Re or I			Rs			Rs*		
	C	D	R	C	D	R	C	D	R
C	1			1	1		1	1	1
D	1	1	1	1	1	1	1	1	1
R			1						1

Un paio di commenti

- La chiusura riflessiva e transitiva è uguale a R^{corpo}
- Se ci sono variabili immodificate in *e*, non è così:
- siano M, N i valori iniziali di C e D:

```
while D/=M loop R:=C rem D; C:=D; D:=R; end loop;
```

- sopra, Examiner non segnala errori (serve più semantica)
- Però può individuare errori legati al mancato aggiornamento della guardia: ad esempio

```
while D/=0 loop R:=M rem N; C:=D; D:=R; end loop;
```

- Nozione di *stabilità* della guardia:
  - se si entra nel ciclo, *non si termina*

Flusso di un sottoprogramma

- Calcolabile per composizione
  - confronto con le annotazioni
  - errori se discrepanti
- Altri costrutti
  - per traduzione a quelli già visti
  - vincoli di SPARK essenziali
- Variabili composite
  - considerate globalmente
    - aggiornate se si aggiorna una parte (grana grossa)
  - vincoli SPARK essenziali
    - esempio: importazione solo al primo livello

S³ 2009/10 – Flussi informativi

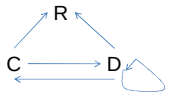
STABILITA' E TERMINAZIONE

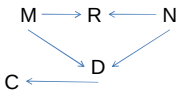
Stabilità di una guardia

- Due passi:
  - stabilità delle variabili
  - stabilità della guardia
- Rappresentazione grafica della R (del corpo)
  - un arco orientato da i a j se $R_{ij} = 1$
    - il valore di i può contribuire al valore di j
  - solo le variabili del corpo

Esempio - GCD

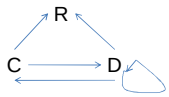
R:=C rem D; C:=D; D:=R;

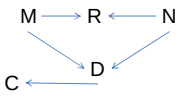




R:=M rem N; C:=D; D:=R;

Variabili stabili





- Una var è *stabile* se *non* è raggiungibile da un ciclo
  - di indice i, se i è la lunghezza del cammino per arrivarci
- A destra non ci sono variabili stabili
- A sinistra sono tutte stabili
  - M,N indice 0; R, D indice 1; C indice 2

Guardia stabile

- Una variabile che compare nella guardia ma *non* nel corpo è stabile di indice 0
- La guardia è *stabile*
  - se tutte le sue variabili sono stabili

```
while D/=0 loop R:=C rem D; C:=D; D:=R; end loop;
```

```
while D/=0 loop R:=M rem N; C:=D; D:=R; end loop;
```

- La seconda guardia è stabile, la prima no.
- Nel secondo caso, Examiner segnala che c'è un problema
  - se si entra nel ciclo, perché la guardia non cambia

Alcune osservazioni

- (si entra) e (guardia stabile) => non termina
  - vuol dire anche
- (si entra) e termina => (guardia non stabile)
  - ma non si può dedurre che
- (si entra) e (guardia non stabile) => termina
  - per questa ci vuole più semantica
  - funzione di terminazione

S³ 2009/10 – Flussi informativi

CONCLUSIONI

Valutazione

- Le verifiche sono efficienti
- Forniscono indicazioni utili sulla correttezza
  - ma non complete
- I messaggi di errore sono
  - semplici
  - utili per individuare i difetti
- Buon compromesso per software non critico

S³ 2009/10 – Flussi informativi

PROSSIMO ARGOMENTO:

CONTRATTI

Esercizio: if then

C = if e(Σ) then S' else null; end if;

- Si compongono le matrici dei singoli blocchi

- $D^C = D^{S'}$ or $D^{null} = D^{S'}$ or $\mathbb{O} = D^{S'}$
- $P^C = P^{S'}$ or $P^{null} = P^{S'}$ or $\mathbb{I} = \mathbb{I}$
- $R = R^{e(\Sigma)}$ or $R^{S'}$ or $\mathbb{I}$

