# Dominio non relazionale per l'analisi delle congruenze

Arianna Cipolla

Università degli studi di Parma

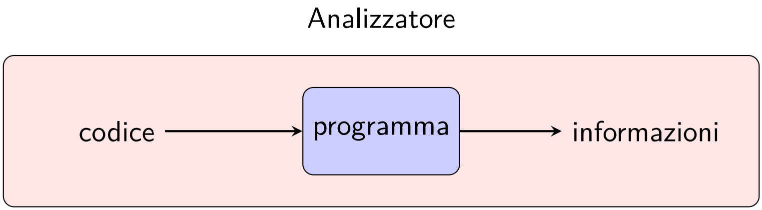
*arianna.cipolla@studenti.unipr.it*

Seminario Linguaggi, Interpreti e Compilatori
Dicembre 13, 2024

# Indice


1. Analisi Statica
   - Definizione
   - Esempio
   - Galois
2. Dominio delle congruenze
   - Definizione
   - Connessioni di Galois
   - Operazioni
   - Esempio Concreto
   - Utilità
3. Bibliografia

# Analisi Statica - Definizione



# Esempio

```
int mod(int A, int B) {
    int Q = 0;
    int R = A;
    while (R >= B) {
        R = R - B;
        Q = Q + 1;
    }
    return R;
}
```

/* Stato concreto */

A = 10; B = 3;

Q = 3; R = 1;

/* Dominio dei segni */

$A \geq 0$; $B \geq 0$;

$Q \geq 0$; $R = \top$;

/* Dominio dei segni relazionale */

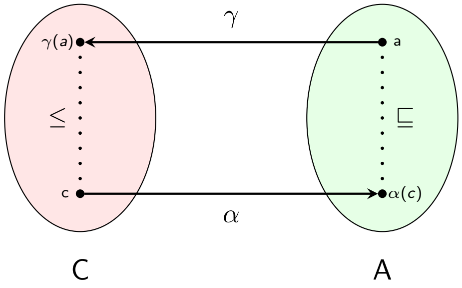
$A \geq 0$; $B \geq 0$;

$Q \geq 0$; $0 \leq R < B$;

# Connessioni di Galois

Una connessione di Galois è una relazione tra due domini definita da due funzioni:

- $\gamma_b$: mappa concretizzazione
- $\alpha_b$: mappa astrazione



# Dominio delle congruenze - Definizone

L'insieme delle congruenze viene definito come:

$$X \in a\mathbb{Z} + b$$

dove:

- $a$ è un numero intero che rappresenta il modulo
- $\mathbb{Z}$ è l'insieme dei numeri interi
- $b$ è un numero intero che rappresenta il resto

dunque $X$ è l'insieme di tutti i numeri interi che divisi per $a$ danno resto $b$.

# Astrazione

L'insieme dei valori astratti viene definito come:

$$\mathcal{B}^\sharp = \{(a\mathbb{Z} + b) \mid a \in \mathbb{N},\ b \in \mathbb{Z}\} \cup \{\perp_b^\sharp\}$$

- $1\mathbb{Z} + 0$ insieme più grande
- $0\mathbb{Z} + c$ singolo intero c
- $\perp_b^\sharp$ insieme vuoto

# Reticolo

## Definition

Un reticolo è una struttura matematica che organizza gli oggetti in base a un criterio di ordine.

Il reticolo completo viene formato come $(\mathbb{N}, |, \vee, \wedge, 1, 0)$ dove:

- $\mathbb{N}$ è l'insieme dei numeri interi positivi
- $|$ è la relazione di ordine parziale "divide"
- $\vee$ l'operazione di join espande cercando il minimo che contiene entrambi (mcm)
- $\wedge$ l'operazione di meet restringe cercando il massimo che è contenuto in entrambi (MCD)
- 1 è l'infimo, il divisore comune più piccolo di tutti i numeri
- 0 è il supremo, è il multiplo comune più grande di tutti i numeri

# Reticolo

Nella forma astratta il reticolo viene definito come:

$$(\mathcal{B}^\sharp, \sqsubseteq_b^\sharp, \sqcup_b^\sharp, \sqcap_b^\sharp, \perp_b^\sharp, (1\mathbb{Z}+0))$$

dove:

- $\mathcal{B}^\sharp = \{(a\mathbb{Z}+b) \mid a \in \mathbb{N},\ b \in \mathbb{Z}\} \cup \{\perp_b^\sharp\}$
- $(a\mathbb{Z}+b) \sqsubseteq_b^\sharp (a'\mathbb{Z}+b') \iff a'|a \text{ e } b \equiv b'[a']$
- $(a\mathbb{Z}+b) \sqcup_b^\sharp (a'\mathbb{Z}+b') = (a \wedge a' \wedge |b-b'|)\mathbb{Z}+b$
- $(a\mathbb{Z}+b) \sqcap_b^\sharp (a'\mathbb{Z}+b') = \begin{cases} (a \vee a')\mathbb{Z}+b'' & \text{if } b \equiv b'[a \vee a'] \\ \perp_b^\sharp & \text{altrimenti} \end{cases}$

  dove $b''$ è un valore congruente sia a $b$ che a $b'$ mod$(a \vee a')$

# Connessioni di Galois

Possiamo costruire una connessione di Galois come segue:

$$\gamma_b(X_b^\sharp) = \begin{cases} \{ak + b \mid k \in \mathbb{Z}\} & \text{se } X_b^\sharp = (a\mathbb{Z} + b) \\ \emptyset & \text{se } X_b^\sharp = \perp_b^\sharp \end{cases}$$

Per garantire che ogni insieme concreto abbia una sola rappresentazione astratta in $a\mathbb{Z} + b$ assumiamo che:

- $a = 0$
  *oppure*
- $0 \leq b < a$

$$\alpha_b(C) = \sqcup^\sharp_{c \in C}(0\mathbb{Z} + c)$$

Il join combina i numeri di C per ottenere un rappresentante astratto unico che include tutti i numeri di C in modo compatto.

# Operazioni Astratte

- Intersezione astratta coincide con il $\sqcap_b^\sharp$ (meet) ed è esatta non perdendo precisione
- Unione astratta coincide con il $\sqcup_b^\sharp$ (join) ed è ottimale, il risultato è il più preciso possibile
- Le operazioni aritmetiche $(+, -, *)$ astratte corrispondono alle loro controparti concrete e sono ottimali
- L'operazione di divisione $(\div)$ nel caso del singleton è precisa, mentre in tutti gli altri casi restituisce ua rappresentazione meno precisa
- L'operatore $\overleftarrow{\leq 0_b^\sharp}$ identifica $[-\infty, 0]_b^\sharp = 1\mathbb{Z} + 0$ e nel caso in cui $X^\sharp = 0\mathbb{Z} + c$ (con $c > 0$) ritorna $\perp_b^\sharp$

# Gestione del dominio

Il dominio non relazionale delle congruenze ha un'altezza infinita, di conseguenza quando si vanno a formare delle catene di insiemi si può crescere o decrescere infinitamente.

- Le **catene strettamente crescenti** sono limitate dalla natura decrescente del modulo $a$ limitato ad 1
  - come widening (ampliamento) viene utilizzato il meet ($\triangledown_b = \sqcup_b^\sharp$)
- Le **catene strettamente decrescenti** sono potenzialmente infinite ($\mathbb{Z}, 2\mathbb{Z}, 4\mathbb{Z}, \dots$) quindi è utile definire un operatore di restringimento ($\triangle_b$)
  - $(a\mathbb{Z} + b) \ \triangle_b \ (a'\mathbb{Z} + b') = \begin{cases} a'\mathbb{Z} + b' & \text{se } a = 1 \\ a\mathbb{Z} + b & \text{altrimenti} \end{cases}$

    Esempi:
    $\mathbb{Z} \triangle_b 2\mathbb{Z}$ prendiamo $2\mathbb{Z}$
    $2\mathbb{Z} \triangle_b 4\mathbb{Z}$ prendiamo $2\mathbb{Z}$

# Esempio di codice

```
int x = 0, y = 2;
while (x < 40) {
    x = x + 2;
    if (x < 5)
        //y = 18k + 2
        y = y + 18;
    else if (x > 8)
        //y = -30k + 2
        y = y - 30;
}
```

$x \in 0\mathbb{Z} + 0$
$y \in 0\mathbb{Z} + 2$

```
/* Iterazioni */
1:  x = 0; y = 2;
2:  x = 2; y = 20;
3:  x = 4; y = 38;
4:  x = 6; y = 8;
5:  x = 8; y = -22;
6:  x = 10; y = -52;
```

$x \in 2\mathbb{Z} + 0$
$y \in 6\mathbb{Z} + 2 =$
$= \{\dots, -22, \dots, 2, 8, 14, 20, \dots\}$

# Utilizzo

**Indirizzi di memoria**: molti seguono una periodicità o una struttura regolare, un insieme di essi è **congruente** se segue la relazione

$$\text{Indirizzo} \equiv b(mod\ a)$$

Esempio: $\text{Indirizzi} = 32k + 4$

- gli indirizzi accedono sempre al $4^\circ$ byte di una riga di cache da 32 byte
- sono congruenti modulo $a = 32$, con offset $b = 4$.

# Bibliografia

- Antoine Miné (2017)
  Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation
- Samuel Larsen, Emmett Witchel and Saman Amarasinghe (2002)
  Increasing and Detecting Memory Address Congruence