Fondamenti dei linguaggi di programmazione

Aniello Murano
Università degli Studi di Napoli
“Federico II”

Riassunto delle lezioni precedenti

- Prima Lezione: Introduzione e motivazioni del corso; Sintassi e semantica di ARITHM
- Seconda lezione: Sintassi e semantica operazionale del linguaggio imperativo IMP
- Terza lezione: Tecniche di prova per induzione:
  - Induzione matematica (ipotesi induttiva su numeri precedenti)
  - Induzione strutturale (ipotesi induttiva su sottotermini)
  - Induzione ben fondata (include i precedenti)
  - Induzione su derivazione (ipotesi induttiva su sottoderivazioni)

Definizioni Induttive di Domini

Introduzione

- In questo capitolo studieremo la teoria degli ***insiemi definiti induttivamente***
- Un esempio di tale insieme è il linguaggio IMP definito induttivamente dalle regole sintattiche e semantiche introdotte nelle precedenti lezioni
- Questa teoria permette di provare proprietà degli insiemi in modo efficiente, utilizzando le regole che definiscono l'insieme (***induzione sulle regole***)
- Per esempio, è possibile dimostrare che un insieme definito induttivamente tramite regole è il **minimo insieme chiuso** rispetto alle regole date
- Dunque, l'induzione sulle regole è largamente utilizzata nella costruzione dei linguaggi (come vedremo per IMP) e se applicata alle regole semantiche del linguaggio permette di ragionare sulla sua semantica operazionale.

Definizione induttiva di insiemi

- Nelle lezioni precedenti abbiamo già visto insiemi definiti induttivamente(Aexp, $\rightarrow_{Com}$, ecc.). Queste definizioni includono regole sintattiche (grammatiche) per la definizione sintattica e regole semantiche per la definizione degli alberi di derivazione.
- Una **definizione induttiva** di un insieme è una collezione di regole che include assiomi e regole di inferenza. Un assioma x indica che x è un elemento dell'insieme per default. Una regola di inferenza $\{x_1,\ldots,x_n/\}/x$ mostra che x è un elemento dell'insieme se $x_1,\ldots x_n$ lo sono
- Per esempio, l'insieme dei numeri naturali $\mathbb{N}$ è un insieme definito induttivamente, la cui definizione consiste nelle seguenti due regole

$$\frac{}{0} \qquad\qquad \frac{n}{succ(n)}$$

Principio di induzione sulle regole descrizione informale

- Il principio di induzione sulle regole si basa sulla seguente idea

"se una **proprietà è preservata** nel passaggio **dalle premesse alle conclusioni** di tutte le regole utilizzate in una derivazione allora la proprietà vale anche per la conclusione della derivazione"

- Se questo è vero per tutte le regole, allora la proprietà è **vera per tutti gli elementi** dell'insieme definito dalle regole.

Principio di induzione sulle regole

- Sia $\mathcal{R}$ un insieme di regole (insieme di assiomi e regole di inferenza) e $\mathcal{I}$ l'insieme delle istanze delle regole di $\mathcal{R}$
- Con $\mathcal{I}_\mathcal{R}$ indichiamo l'insieme di tutti gli elementi x ottenuti da $\mathcal{I}$ per cui esiste una derivazione consistente con $\mathcal{R}$.
- Formalmente $\mathcal{I}_\mathcal{R} = \{x \mid \Vdash_R x\}$
- Per esempio, si consideri la seguente regola e una sua istanza

$$\frac{\langle a_0,\sigma\rangle \to n_0 \qquad \langle a_1,\sigma\rangle \to n_1}{\langle a_0 \times a_1\ ,\ \sigma\rangle \to n = n_0 \times n_1} \qquad \frac{\langle 2,\sigma\rangle \to 3 \qquad \langle 3,\sigma\rangle \to 4}{\langle 2 \times 3\ ,\ \sigma\rangle \to 12 = 3 \times 4}$$

- $\langle 2 \times 3\ ,\ \sigma\rangle \to 12$ non è un elemento di $\mathcal{I}_\mathcal{R}$ perché l'istanza è corretta, ma non è derivabile con le regole di $\mathcal{R}$
- **Principio di induzione:** Sia P una proprietà, allora $\forall x \in \mathcal{I}_\mathcal{R}.\ P(x)$ sse per tutte le istanze di regole (X/y) in $\mathcal{R}$ per le quali $X \subseteq \mathcal{I}_\mathcal{R}$

$$(\forall x \in X.\ P(x)) \Rightarrow P(y)$$

Definizione di insieme (R-)chiuso

- Un insieme Q è $\mathcal{R}$-chiuso rispetto alle istanze delle regole $\mathcal{R}$ sse ogni volta che le premesse di una istanza di una regola appartengono a Q allora anche le conclusioni appartengono a Q.
- Formalmente: Q è $\mathcal{R}$-chiuso rispetto alle istanze delle regole R sse per ogni istanza di regola (X/y), si ha che $X \subseteq Q \Rightarrow y \in Q$

◆ Rules

$$n \in \mathrm{Aexp} \qquad X \in \mathrm{Aexp} \qquad \frac{a_0 \in \mathrm{Aexp} \quad a_1 \in \mathrm{Aexp}}{a_0 + a_1 \in \mathrm{Aexp}}$$

◆ Rule instances (X/y): set of premises/conclusion

$\varnothing/1 \quad \varnothing/2 \quad \varnothing/3 \quad \dots$

$\varnothing/x_1 \quad \varnothing/x_2 \quad \varnothing/x_3 \quad \dots$

$\{1\}/(1+1) \quad \{1, 2\}/(1+2) \quad \{1, 2\}/(2+1) \quad \dots$

$\{1, x_1\}/(1+x_1) \quad \{1, x_1\}/(x_1+1) \quad \{x_1, 2\}/(2+x_1) \quad \dots$

$\{x_1\}/(x_1+x_1) \quad \{x_1, x_2\}/(x_1+x_2) \quad \dots$

$\dots$

Minimo insieme chiuso

- $\mathcal{I}_\mathcal{R}$ è il minimo insieme chiuso rispetto alle regole di $\mathcal{R}$
  - $\mathcal{I}_\mathcal{R}$ è chiuso
  - Se Q è un insieme $\mathcal{R}$-chiuso allora $\mathcal{I}_\mathcal{R} \subseteq$ Q

Dimostrazione

- $\mathcal{I}_\mathcal{R}$ è $\mathcal{R}$-chiuso

  Sia $(X/y) \in \mathcal{R}$. Se $X \subseteq \mathcal{I}_\mathcal{R}$ allora per definizione di $\mathcal{I}_\mathcal{R}$ tutti gli elementi di X sono derivabili e possiamo costruire con (X/y) una deivazione per y. Quindi y è in $\mathcal{I}_\mathcal{R}$.

- $\mathcal{I}_\mathcal{R}$ è il più piccolo: se Q è $\mathcal{R}$-chiuso allora $\mathcal{I}_\mathcal{R} \subseteq Q$.

  Sia Q un insieme $\mathcal{R}$-chiuso. Se $y \in \mathcal{I}_\mathcal{R}$ allora deve esistere una derivazione $d \Vdash_\mathcal{R} y$. Dimostriamo, per induzione sulla derivazione d, che se $d \Vdash y$ e $y \in \mathcal{I}_\mathcal{R}$ allora $y \in Q$.

  - I casi di base sono immediati perchè Q è $\mathcal{R}$-chiuso.

  - Caso induttivo: siano $d_i \Vdash_\mathcal{R} x_i$, $(i = 1, .., n)$ le derivazioni delle premesse di d. Per ipotesi induttiva $x_i \in Q$, $(i = 1, .., n)$ e poichè Q è $\mathcal{R}$-chiuso deve essere $y \in Q$.

Induzione sulle regole

Dalla induzione sulle derivazioni possiamo derivare la seguente induzione sulle regole

$$\frac{\forall (X/y) \in \mathcal{R}. \, (X \subseteq \mathcal{I}_\mathcal{R} \wedge \forall x \in X. P(x)) \implies P(y)}{\forall x \in \mathcal{I}_\mathcal{R}. \, P(x)}$$

Prova. Sia $Q = \{x \in \mathcal{I}_\mathcal{R} | P(x)\}$. Per dimostrare che la proprietà P è vera per tutti gli elementi di $\mathcal{I}_\mathcal{R}$ è sufficiente dimostrare che $\mathcal{I}_\mathcal{R} \subseteq Q$.

La premessa della regola dice che Q è $\mathcal{R}$-chiuso e, dato che $\mathcal{I}_\mathcal{R}$ è il più piccolo insieme $\mathcal{R}$-chiuso, questo implica proprio $\mathcal{I}_\mathcal{R} \subseteq Q$.

- Notiamo che la premessa della regola è necessaria in quanto $\mathcal{I}_\mathcal{R}$ è $\mathcal{R}$-chiuso.

Break time

Punto fisso

Set Construction Operator

- Define an operator based on rule instances

  $R(B) = \{ y \mid \exists X \subseteq B.\ (X/y) \in \mathbf{R} \}$

  – create set of conclusions where all the premises are in B

- Series

$A_0 = \varnothing = R^0(\varnothing)$	empty set
$A_1 = R(A_0) = R^1(\varnothing)$	axioms
$A_2 = R(A_1) = R^2(\varnothing)$	derived from axioms in 1 step
...	
$A_n = R(A_{n-1}) = R^n(\varnothing)$	

Example

- Operator

  $R(B) = N \cup Loc \cup \{ a_0 + a_1 \mid a_0, a_1 \in B \}$

- Sets

  $A_0 = \varnothing$

  $A_1 = R(A_0) = N \cup Loc$

  – numbers and locations

  $A_2 = R(A_1) = N \cup Loc \cup \{ a_0 + a_1 \mid a_0, a_1 \in A_1 \}$

  – numbers, locations and sum of any number or location

  ...

Explicit Construction

- Inclusion

  $A_0 \subseteq A_1 \subseteq A_2 \subseteq \cdots \subseteq A_n \subseteq \cdots$

- Define limit

  $$A = \cup_{n \in \omega} A_n$$

  - A is R-closed
  - $R(A) = A$
  - A is the least R-closed set

- Fixed-point of an operator
  - $fix(R) = \cup_{n \in \omega} R^n(\varnothing)$

Summary

- Definitions are equivalent
  - R-Closure
  - Limit of repeated application of operator R
  - Fixed-point of operator R

- Lets use them...

Inductive Syntax Definition

- Formal explanation of Aexp, Bexp, and Com

$$
\begin{array}{lllll}
a & \in & Aexp & ::= & n \mid X \mid a_0 + a_1 \mid a_0 - a_1 \mid a_0 \times a_1 \\
b & \in & Bexp & ::= & \mathbf{true} \mid \mathbf{false} \mid a_0 = a_1 \mid a_0 \leq a_1 \mid \\
 & & & & \neg b \mid b_0 \wedge b_1 \mid b_0 \vee b_1 \\
c & \in & Com & ::= & \mathbf{skip} \mid X := a \mid c_0; c_1 \mid \\
 & & & & \mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1 \mid \mathbf{while}\ b\ \mathbf{do}\ c
\end{array}
$$

- Interpret grammar as inference rules which generate the set
  - Basis for *structural induction*

Inductive Definition of Relations

- We have already seen one: $\leadsto$

$$\langle n, \sigma\rangle \leadsto n \quad [\text{Const}]$$

$$\langle X, \sigma\rangle \leadsto \sigma(X) \quad [\text{Loc}]$$

$$\frac{\langle a_1, \sigma\rangle \leadsto n_1 \qquad \langle a_2, \sigma\rangle \leadsto n_2}{\langle a_1 - a_2, \sigma\rangle \leadsto n} \quad [\text{Sub}]$$

where n *is the result of substracting* n_2 *from* n_1

$$\frac{\langle a_1, \sigma\rangle \leadsto n_1 \qquad \langle a_2, \sigma\rangle \leadsto n_2}{\langle a_1 + a_2, \sigma\rangle \leadsto n} \quad [\text{Sum}]$$

where n *is the sum of* n_1 *and* n_2

$$\frac{\langle a_1, \sigma\rangle \leadsto n_1 \qquad \langle a_2, \sigma\rangle \leadsto n_2}{\langle a_1 \times a_2, \sigma\rangle \leadsto n} \quad [\text{Prod}]$$

where n *is the product of* n_1 *and* n_2

CS 386L Fall 2004

27

Inductive Semantics of While

$$\frac{\langle b, \sigma\rangle \leadsto \text{false}}{\langle \textbf{while}\ b\ \textbf{do}\ c, \sigma\rangle \leadsto \sigma} \qquad \frac{\langle b, \sigma\rangle \leadsto \text{true} \quad \langle c, \sigma\rangle \leadsto \sigma' \quad \langle \textbf{while}\ b\ \textbf{do}\ c, \sigma'\rangle \leadsto \sigma''}{\langle \textbf{while}\ b\ \textbf{do}\ c, \sigma\rangle \leadsto \sigma''}$$

- Convert rules to operator $R_{\leadsto}$

$$\begin{aligned} R_{\leadsto}(S) = \{\, & \langle \textbf{while}\ b\ \textbf{do}\ c, \sigma, \sigma\rangle \mid \langle b, \sigma, \text{false}\rangle \in \leadsto_{\text{Bexp}} \,\} \\ \cup \{\, & \langle \textbf{while}\ b\ \textbf{do}\ c, \sigma, \sigma''\rangle \mid \\ & \langle b, \sigma, \text{true}\rangle \in \leadsto_{\text{Bexp}} \\ & \&\ \langle c, \sigma, \sigma'\rangle \in S \\ & \&\ \langle \textbf{while}\ b\ \textbf{do}\ c, \sigma', \sigma''\rangle \in S \,\} \\ \cup\ & \textit{...other rules for Com...} \end{aligned}$$

CS 386L Fall 2004

Fixed-Point of $R_{\leadsto}$

- Equivalence
  $\mathit{fix}(R_{\leadsto}) = \leadsto$
- We showed that evaluation is a *function*

Inductive Function Definitions

- Variables in an arithmetic expression
  - Loc(n) $= \emptyset$
  - Loc(X) $= \{ X \}$
  - Loc($a_0 + a_1$) $=$ Loc(a_0) $\cup$ Loc(a_1)
  - Loc($a_0 - a_1$) $=$ Loc(a_0) $\cup$ Loc(a_1)
  - Loc($a_0 \times a_1$) $=$ Loc(a_0) $\cup$ Loc(a_1)
- We are used to seeing programs of this form
  - But normally a "definition" gives a convenient name to something that is previously known
  - But here the "thing being defined" is used in the definition
  - In what sense is this a definition?

Inductive Function Definitions

- Let Loc be the smallest binary relation closed under the following rules

$$(n, \emptyset) \in \text{Loc}$$

$$(X, \{X\}) \in \text{Loc}$$

$$\frac{(a_0, s_0) \in \text{Loc} \qquad (a_1, s_1) \in \text{Loc}}{\begin{array}{c}(a_0 + a_1, s_0 \cup s_1) \in \text{Loc} \\ (a_0 - a_1, s_0 \cup s_1) \in \text{Loc} \\ (a_0 \times a_1, s_0 \cup s_1) \in \text{Loc}\end{array}}$$

- Loc is a relation. Is it a function?

CS 386L Fall 2004

31

Inductive Function Definitions

- Variables in an arithmetic expression
  - BadLoc(n) $= \emptyset$
  - BadLoc(X) $= \{ X \}$
  - BadLoc($a_0 + a_1$) $=$ BadLoc(a_0) $\cup$ BadLoc(a_1)
  - BadLoc($a_0 - a_1$) $=$ BadLoc(a_0) $\cup$ BadLoc(a_1)
  - BadLoc($a_0 \times a_1$) $=$ BadLoc($a_0 \times (a_1 \times 1)$)
- Why is this a problem?
  - How do you tell well-formed inductive definitions from ill-formed ones

CS 386L Fall 2004

Bad Rule

$$\frac{(a_0 \times (a_1 \times 1), s) \in \text{BadLoc}}{(a_0 \times a_1, s) \in \text{BadLoc}}$$

- Closure?
  - The set $\{ (3\times(2\times1), \emptyset), (3\times2, \emptyset) \}$ is R-closed
  - But so is $\{ (5\times(3\times1), \emptyset), (5\times3, \emptyset) \}$
  - The least set is degenerate
- Operator?
  $R(B) = \{ (a_0\times a_1, s) \mid (a_0\times(a_1\times1), s) \in B \}$
  - If this is the only rule for $\times$, then no terms involving $\times$ will be included in $R^n(\emptyset)$ for any n