ソフトウェア基礎論配布資料 (3)
変数の宣言と定義

五十嵐 淳
京都大学 大学院情報学研究科知能情報学専攻
e-mail: igarashi@kuis.kyoto-u.ac.jp

平成 17 年 10 月 18 日

1 変数，宣言，有効範囲と変数参照

1.1 変数

数学・論理において**変数**(*variable*) とは，何かの議論の中で何らかの対象を表すための記法である．

x を自然数とする

など**宣言**(*declare*) することによって，それ以降 x をあたかも自然数であるかの如く扱って，「$x+1$」などを意味のあるものとして使うことが可能になる．この「$x+1$」中の「x」を，(宣言と区別して) **変数参照**(*variable reference*) という．また，変数宣言には，例えば，定理の文言内，教科書のある章の間で，といった**有効範囲**(*scope*) が常に (通常暗黙のうちに) 定められていて，その範囲内でのみ参照することができる．別の言い方をすると，宣言もしない (つまり有効範囲にない) 変数のを参照することはできない．

一方，**定義**(*definition*) とは，ある対象と変数を同一視する (対象に名前をつける) という行為と考えられる．例えば

y を $x+1$ と定義する

などと言えば，変数 y(ここで宣言もされている) は $x+1$ と同じ意味をなすものとして，使うことが可能になる．

「$x+1$」が算術式かどうか,「$y+1$」と「$x+1+1$」と「同じもの」であるか，といった事柄を考えるためには，そもそも「x」の宣言の有効範囲にいるのかどうか,「y の定義は何か」といった事を考える必要がある．そのような「議論を行ううえでの仮定」を「議論を取り囲むもの」という意味で**環境**(*environment*) と呼ぶ．

以下では，プログラミング言語に現れる変数の宣言・定義の概念を，上の数学的・論理的直感に基づいて，算術式の言語に導入する．

ちなみに，プログラミング言語 (特に C 言語など) における変数の概念は，このような数学的・論理的な変数とは，一見異なっているように見える．つまり，数学での変数 x は任意の自然数を表しているが，$x*20+x$ といった時に， ひとつ目の x とふたつ目の x が表す自然数は同じである，という意味で有効範囲内では固定された対象を指している．(そのため，英語では “Let x be an arbitrary but fixed natural number.” などと言ったりする．) これに対し，`x = x + 1;` といった代入文では変数の値が変化しうるように見える．しかし，これも，変数はその対象として「値」ではなく「値の入っている箱」(もしくは，メモリアドレス—これはその有効範囲内で一定の値である) を表していると考えられる．ちなみに，Lisp, ML, Haskell といった関数型言語と呼ばれる言語や，Prolog といった論理型言語での変数は，数学・論理での変数に近い．

2 算術式の言語 (最終版)

2.1 変数・宣言・環境・環境のもとでの算術式の定義

まず，変数の集合として可算無限集合 **Var** を考える．要素としては `x`, `y`, `z`, ... などを仮定する．また，以下の議論では変数を表すメタ変数として，x,y,z,... などを使用する．

宣言としては，単なる変数宣言 (“Let x be ...” の代わりに単に x と記述する．) と，定義つきの宣言 ($x = a$ と記述する) を考える[1].

環境は宣言の列として取り扱う．例えば，`x,y=S(Z)` や `x,y=x+S(Z)` は環境である．このように，定義には，以前に宣言された変数を含んだ式が現れることがある．また，ひとつの環境の中で宣言される変数に重複があっても構わなく，`x,x=S(Z)` も環境である．これは，プログラミング言語で大域変数と同じ名前の変数を局所変数として宣言することができることと同様に考えられる．

以下では，環境の集合と，ある表現が算術式であるという判断を規則を使って定義するが，今や算術式は環境がひとつ定まった下で議論されるものである．このため $a \in \mathbf{Aexp}$ という判断を「環境 Γ のもとで a は算術式である」という意味の

$$\Gamma \vdash a \in \mathbf{Aexp}$$

という判断に拡張する．例えば `x,y=S(Z)` $\vdash$ `x + y` $\in$ **Aexp** や `x,y=x+S(Z)` $\vdash$ `y * S(x)` $\in$ **Aexp** といった判断が導出できる．

> $\vdash$ という記号は，数理論理学などでは「(〜という仮定の下で)〜が証明 (導出) できる」読まれる記号であるが，もともと Gottlob Frege という 19 世紀後半〜20 世紀始めの数学者・論理学者・哲学者が導入した記号である．(そこでは，縦棒と横棒は別々の意味を持つ記号である．)

[1] 本来，宣言と定義は別の行為であるが，ここでは上の例のような宣言と同時に行われる定義を考えている．

$$\frac{}{\bullet \in \mathbf{Env}} \quad (\textsc{Env-Empty})$$

$$\frac{\Gamma \in \mathbf{Env}}{\Gamma, x \in \mathbf{Env}} \quad (\textsc{Env-Decl})$$

$$\frac{\Gamma \in \mathbf{Env} \qquad \Gamma \vdash a \in \mathbf{Exp}}{\Gamma, x = a \in \mathbf{Env}} \quad (\textsc{Env-Defn})$$

$$\frac{\Gamma \in \mathbf{Env}}{\Gamma, x \vdash \#^0 x \in \mathbf{Aexp}} \quad (\text{A-}\textsc{Var0})$$

$$\frac{\Gamma \in \mathbf{Env} \qquad \Gamma \vdash a \in \mathbf{Aexp}}{\Gamma, x = a \vdash \#^0 x \in \mathbf{Aexp}} \quad (\text{A-}\textsc{Var1})$$

$$\frac{\Gamma \vdash \#^i x \in \mathbf{Aexp}}{\Gamma, x \vdash \#^{i+1} x \in \mathbf{Aexp}} \quad (\text{A-}\textsc{Var2})$$

$$\frac{\Gamma \vdash \#^i x \in \mathbf{Aexp} \qquad \Gamma \vdash a \in \mathbf{Aexp}}{\Gamma, x = a \vdash \#^{i+1} x \in \mathbf{Aexp}} \quad (\text{A-}\textsc{Var3})$$

$$\frac{\Gamma \vdash \#^i x \in \mathbf{Aexp} \qquad (x \not\equiv y)}{\Gamma, y \vdash \#^i x \in \mathbf{Aexp}} \quad (\text{A-}\textsc{Var4})$$

$$\frac{\Gamma \vdash \#^i x \in \mathbf{Aexp} \qquad \Gamma \vdash a \in \mathbf{Aexp} \qquad (x \not\equiv y)}{\Gamma, y = a \vdash \#^i x \in \mathbf{Aexp}} \quad (\text{A-}\textsc{Var5})$$

$$\frac{\Gamma \in \mathbf{Env}}{\Gamma \vdash \mathtt{Z} \in \mathbf{Aexp}} \quad (\text{A-}\textsc{Zero})$$

$$\frac{\Gamma \vdash a \in \mathbf{Aexp}}{\Gamma \vdash \mathtt{S}(a) \in \mathbf{Aexp}} \quad (\text{A-}\textsc{Succ})$$

$$\frac{\Gamma \vdash a_1 \in \mathbf{Aexp} \qquad \Gamma \vdash a_2 \in \mathbf{Aexp}}{\Gamma \vdash a_1 \ \mathtt{+} \ a_2 \in \mathbf{Aexp}} \quad (\text{A-}\textsc{Plus})$$

$$\frac{\Gamma \vdash a_1 \in \mathbf{Aexp} \qquad \Gamma \vdash a_2 \in \mathbf{Aexp}}{\Gamma \vdash a_1 \ \mathtt{*} \ a_2 \in \mathbf{Aexp}} \quad (\text{A-}\textsc{Mult})$$

図 1: 環境・環境の下での算術式

変数参照の記法　上の例と同様 `x,x=S(Z)` $\vdash$ `x + 0` $\in$ **Aexp** は正しい判断であるが，この変数参照 `x` は，どちらの宣言を参照しているのだろうか？また，常に片方の宣言を参照しているものとして解釈が固定されてしまうのであろうか？

多くのプログラミング言語では，変数参照は「プログラムの文面上の一番近くでの宣言」を常に参照することになっている．このような参照先の決め方を lexical scoping という，別の言い方をすると，`x,x=S(Z)` という環境下は最初の `x` の宣言の有効範囲内にない，とも言える．

ここでは，過去に行った宣言は全て有効範囲にあり，それを参照する手段 (記法) を提供する．具体的には，変数参照を $\#^i x$ と記述し,「i 番目に近い x の宣言」を参照する．$i = 0$ のとき $\#^0 x$ を単に x と書く．

2.1.1 定義: 環境 (メタ変数 Γ) の集合 **Env** と，判断 $\Gamma \vdash a \in \mathbf{Aexp}$ を図 1 の規則で定義する．

2.1.2 例: 以下の判断が導出できる．

1. $\bullet, \mathtt{x}, \mathtt{y{=}S(x)} \in \mathbf{Env}$
2. $\bullet, \mathtt{x}, \mathtt{y{=}S(x)} \vdash \mathtt{x\ +\ S(y)} \in \mathbf{Aexp}$
3. $\bullet, \mathtt{x}, \mathtt{y{=}S(x)}, \mathtt{x{=}x{+}y} \vdash \#^1\mathtt{x\ *\ S(y)\ +\ x} \in \mathbf{Aexp}$

以下ではしばしば環境の先頭の 「•,」を省略する.

2.2 簡約関係

簡約関係も，単に a が a' に簡約されるのではなく,「ある環境の下で」簡約されるという $\Gamma \vdash a \longrightarrow a'$ という判断を考える．足し算・かけ算の計算過程だけではなく，定義の参照を定義内容で置き換える過程も簡約として取り扱う．つまり，例えば

$$\mathtt{x{=}Z,y{=}S(x)} \vdash \mathtt{y\ +\ x} \longrightarrow \mathtt{S(x)\ +\ x}$$
$$\mathtt{x{=}Z,y{=}S(x)} \vdash \mathtt{y\ +\ x} \longrightarrow \mathtt{y\ +\ Z}$$

といった判断が考えられる.

さて，$\mathtt{x{=}Z}$，$\mathtt{x{=}S(x)}$ という判断の下で変数参照 x (正確には $\#^0\mathtt{x}$ は何に簡約されるだろうか？これを素朴に $\mathtt{S(x)}$ とするのは間違いである．何故なら，(内側の) x を定義した時点で，その内容は「(外側の) x の次の数」であるが，$\mathtt{x{=}Z}$，$\mathtt{x{=}S(x)}$ の下での $\mathtt{S(x)}$ は「(内側の) x の次の数」を示すからである．これを考慮すると，

$$\mathtt{x{=}Z,x{=}S(x)} \vdash \mathtt{x} \longrightarrow \mathtt{S(}\#^1\mathtt{x)}$$

が導出されるべき判断である．このような判断を導出するためには，$\mathtt{x{=}Z}$ の下での算術式 $\mathtt{S(x)}$ が，より宣言の多い環境下で，どのような形で表現されるかを考える必要がある.

このために，以下で定義される $a \Uparrow x$ という関数 (shift 関数と呼ぶ) を考える.

$$\begin{array}{lcl}
\#^i x \Uparrow x & = & \#^{i+1} x \\
\#^i x \Uparrow y & = & \#^i x \qquad (y \neq x) \\
\mathtt{0} \Uparrow x & = & \mathtt{0} \\
\mathtt{S(}a_0\mathtt{)} \Uparrow x & = & \mathtt{S(}a_0 \Uparrow x\mathtt{)} \\
(a_1\mathtt{+}a_2) \Uparrow x & = & (a_1 \Uparrow x)\mathtt{+}(a_2 \Uparrow x) \\
(a_1\mathtt{*}a_2) \Uparrow x & = & (a_1 \Uparrow x)\mathtt{*}(a_2 \Uparrow x)
\end{array}$$

$a \Uparrow x$ は直感的には，ある環境 Γ のもとでの算術式 a の Γ, x のもとでの表現を表す.

これを使って簡約関係を定義する.

2.2.1 定義: 簡約関係 $\Gamma \vdash a \longrightarrow a'$ は図 2 の規則で定義される.

同様に eager/lazy な簡約関係も定義することができる．(eager な関係で使われる値の概念は，定義内容を持たない変数宣言の参照を値と考え拡張する．)

$$\frac{\Gamma \vdash a \in \mathbf{Aexp}}{\Gamma, x = a \vdash x \longrightarrow a \Uparrow x} \text{ (E-Def0)}$$

$$\frac{\Gamma \vdash \#^i x \longrightarrow a}{\Gamma, x \vdash \#^{i+1} x \longrightarrow a \Uparrow x} \text{ (E-Def1)}$$

$$\frac{\Gamma \vdash \#^i x \longrightarrow a \qquad \Gamma \vdash a' \in \mathbf{Aexp}}{\Gamma, x = a' \vdash \#^{i+1} x \longrightarrow a \Uparrow x} \text{ (E-Def2)}$$

$$\frac{\Gamma \vdash \#^i x \longrightarrow a \qquad (y \not\equiv x)}{\Gamma, y \vdash \#^i x \longrightarrow a \Uparrow y} \text{ (E-Def3)}$$

$$\frac{\Gamma \vdash \#^i x \longrightarrow a \qquad \Gamma \vdash a' \in \mathbf{Aexp} \qquad (y \not\equiv x)}{\Gamma, y = a' \vdash \#^i x \longrightarrow a \Uparrow y} \text{ (E-Def4)}$$

$$\frac{\Gamma \vdash a_0 \in \mathbf{Aexp}}{\Gamma \vdash a_0 \texttt{ + Z} \longrightarrow a_0} \text{ (E-PlusZero)}$$

$$\frac{\Gamma \vdash a_1 \in \mathbf{Aexp} \qquad \Gamma \vdash a_2 \in \mathbf{Aexp}}{\Gamma \vdash a_1 \texttt{ + S(}a_2\texttt{)} \longrightarrow \texttt{S(}a_1 \texttt{ + } a_2\texttt{)}} \text{ (E-PlusSucc)}$$

$$\frac{\Gamma \vdash a_0 \in \mathbf{Aexp}}{\Gamma \vdash a_0 \texttt{ * Z} \longrightarrow \texttt{Z}} \text{ (E-MultZero)}$$

$$\frac{\Gamma \vdash a_1 \in \mathbf{Aexp} \qquad \Gamma \vdash a_2 \in \mathbf{Aexp}}{\Gamma \vdash a_1 \texttt{ * S(}a_2\texttt{)} \longrightarrow a_1 \texttt{ * } a_2 \texttt{ + } a_1} \text{ (E-MultSucc)}$$

$$\frac{\Gamma \vdash a \longrightarrow a'}{\Gamma \vdash \texttt{S(}a\texttt{)} \longrightarrow \texttt{S(}a'\texttt{)}} \text{ (E-Succ)}$$

$$\frac{\Gamma \vdash a_1 \longrightarrow a_1'}{\Gamma \vdash a_1 \texttt{ + } a_2 \longrightarrow a_1' \texttt{ + } a_2} \text{ (E-PlusL)}$$

$$\frac{\Gamma \vdash a_2 \longrightarrow a_2'}{\Gamma \vdash a_1 \texttt{ + } a_2 \longrightarrow a_1 \texttt{ + } a_2'} \text{ (E-PlusR)}$$

$$\frac{\Gamma \vdash a_1 \longrightarrow a_1'}{\Gamma \vdash a_1 \texttt{ * } a_2 \longrightarrow a_1' \texttt{ * } a_2} \text{ (E-MultL)}$$

$$\frac{\Gamma \vdash a_2 \longrightarrow a_2'}{\Gamma \vdash a_1 \texttt{ * } a_2 \longrightarrow a_1 \texttt{ * } a_2'} \text{ (E-MultR)}$$

図 2: 簡約関係: $\Gamma \vdash a \longrightarrow a'$

2.2.2 例: 以下の判断が導出できる.

1. $\bullet, \mathtt{x}, \mathtt{y{=}S(x)}, \mathtt{x{=}x{+}y} \vdash \mathtt{\#^1x\ +\ y} \longrightarrow \mathtt{\#^1x\ +\ S(\#^1x)}$
2. $\bullet, \mathtt{x}, \mathtt{y{=}S(x)}, \mathtt{x{=}x{+}y} \vdash \mathtt{\#^1x\ +\ S(\#^1x)} \longrightarrow \mathtt{S(\#^1x{+}\#^1x)}$