

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 18 年 10 月 24 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、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, \dots 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, \dots 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、定義付きの宣言 ($x = a$ 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 $a \in \text{Aexp}$ 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

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

という判断に拡張する。例えば $x, y = S(Z) \vdash x + y \in \text{Aexp}$ や $x, y = x + S(Z) \vdash y * S(x) \in \text{Aexp}$ 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「(\sim という仮定の下で) \sim 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。(ここでは、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される定義を考えている。

$\frac{}{\bullet \in \mathbf{Env}} \quad (\text{ENV-EMPTY})$	$\frac{\Gamma \vdash \#^i x \in \mathbf{Aexp} \quad (x \neq y)}{\Gamma, y \vdash \#^i x \in \mathbf{Aexp}} \quad (\text{A-VAR4})$
$\frac{\Gamma \in \mathbf{Env}}{\Gamma, x \in \mathbf{Env}} \quad (\text{ENV-DECL})$	$\frac{\Gamma \vdash \#^i x \in \mathbf{Aexp} \quad \Gamma \vdash a \in \mathbf{Aexp} \quad (x \neq y)}{\Gamma, y = a \vdash \#^i x \in \mathbf{Aexp}} \quad (\text{A-VAR5})$
$\frac{\Gamma \in \mathbf{Env} \quad \Gamma \vdash a \in \mathbf{Aexp}}{\Gamma, x = a \in \mathbf{Env}} \quad (\text{ENV-DEFN})$	$\frac{\Gamma \in \mathbf{Env}}{\Gamma \vdash Z \in \mathbf{Aexp}} \quad (\text{A-ZERO})$
$\frac{\Gamma \in \mathbf{Env}}{\Gamma, x \vdash \#^0 x \in \mathbf{Aexp}} \quad (\text{A-VAR0})$	$\frac{\Gamma \vdash a \in \mathbf{Aexp}}{\Gamma \vdash S(a) \in \mathbf{Aexp}} \quad (\text{A-SUCC})$
$\frac{\Gamma \in \mathbf{Env} \quad \Gamma \vdash a \in \mathbf{Aexp}}{\Gamma, x = a \vdash \#^0 x \in \mathbf{Aexp}} \quad (\text{A-VAR1})$	$\frac{\Gamma \vdash a_1 \in \mathbf{Aexp} \quad \Gamma \vdash a_2 \in \mathbf{Aexp}}{\Gamma \vdash a_1 + a_2 \in \mathbf{Aexp}} \quad (\text{A-PLUS})$
$\frac{\Gamma \vdash \#^i x \in \mathbf{Aexp}}{\Gamma, x \vdash \#^{i+1} x \in \mathbf{Aexp}} \quad (\text{A-VAR2})$	$\frac{\Gamma \vdash a_1 \in \mathbf{Aexp} \quad \Gamma \vdash a_2 \in \mathbf{Aexp}}{\Gamma \vdash a_1 * a_2 \in \mathbf{Aexp}} \quad (\text{A-MULT})$
$\frac{\Gamma \vdash \#^i x \in \mathbf{Aexp} \quad \Gamma \vdash a \in \mathbf{Aexp}}{\Gamma, x = a \vdash \#^{i+1} x \in \mathbf{Aexp}} \quad (\text{A-VAR3})$	

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 $x, x=S(Z) \vdash x + 0 \in \mathbf{Aexp}$ 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はない，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 \mathbf{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．

$$\begin{array}{c}
\frac{\Delta \vdash a \Downarrow n}{\Delta, x = a \vdash x \Downarrow n} \quad (\text{E-DEF}) \\
\frac{\Delta \vdash \#^i x \Downarrow n \quad \Delta \vdash a' \in \mathbf{Aexp}}{\Delta, y = a' \vdash \#^i x \uparrow_y \longrightarrow n} \quad (\text{E-SHIFT}) \\
\frac{}{\Delta \vdash Z \Downarrow Z} \quad (\text{E-ZERO}) \\
\frac{\Delta \vdash a \Downarrow n}{\Delta \vdash S(a) \Downarrow S(n)} \quad (\text{E-SUCC}) \\
\frac{\Delta \vdash a_1 \Downarrow S^n(Z) \quad \Delta \vdash a_2 \Downarrow S^m(Z)}{\Delta \vdash a_1 + a_2 \Downarrow S^{n+m}(Z)} \quad (\text{E-PLUS}) \\
\frac{\Delta \vdash a_1 \Downarrow S^n(Z) \quad \Delta \vdash a_2 \Downarrow S^m(Z)}{\Delta \vdash a_1 * a_2 \Downarrow S^{n \cdot m}(Z)} \quad (\text{E-MULT})
\end{array}$$

図 2: 評価関係: $\Delta \vdash a \Downarrow n$

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

以下ではしばしば環境の先頭の「 \bullet, \cdot 」を省略する。

2.2 評価関係

変数を含む式の値は、その式で参照されている変数全ての定義内容が明らかにならないと決定できない。つまり、 $\mathbf{ENV-DECL}$ を使わずに構成された環境の元での式に対してのみ、値を与えることができる。以下、このような、定義 $x = a$ の列からなる環境を定義環境と呼び $\Delta \in \mathbf{DEnv}$ で表記する。(厳密には、そのような環境を構成するための規則を与えるべきだが、省略している。) 定義環境の導入に応じて評価関係の判断は、 $\Delta \vdash a \Downarrow n$ となる。関係を導出する規則は図 2 で与えられる。

さて、この評価関係を使った文脈同値関係を定義してみよう。文脈同値関係はふたつの式を任意の文脈においた時に両者の評価結果が等しくなることと同値なのであった。この時、文脈に対しては、プログラム、つまり評価の対象になるような文脈であることが要請される。そのために、まず、文脈の環境は定義環境である必要がある。文脈を構成するための判断として $\Delta \vdash C \in \mathbf{Ctx}$ を定義する(規則は省略)。さらに、埋め込む式の環境が、文脈の環境に合ったものでなくてはならない。これを表現するために、環境と定義環境間の関係 $\Gamma \leq \Delta$ を定義する。

$$\frac{\Delta \in \mathbf{DEnv}}{\bullet \leq \Delta}$$

$$\frac{\Gamma \leq \Delta \quad \Delta \vdash a \in \mathbf{Aexp}}{\Gamma, x \leq \Delta, x = a}$$

$$\frac{\Gamma \leq \Delta \quad \Gamma \vdash a \in \mathbf{Aexp}}{\Gamma, x = a \leq \Delta, x = a}$$

定義より $\Gamma \leq \Delta$ が成立する時、 Γ は Δ の内側を切り取って、定義をいくつか落としたものだということがわかる。すると、まず、式を \leq の成り立つ定義文脈に埋め込んでも式になることを示すことができる。

2.2.1 補題: もし $\Gamma \vdash a \in \mathbf{Aexp}$ かつ $\Delta \vdash C \in \mathbf{Aexp}$ かつ $\Gamma \leq \Delta$ ならば、 $\Delta \vdash C[a] \in \mathbf{Aexp}$ である。

以上をふまえて、文脈同値は以下のように定義される。

2.2.2 定義: Γ の下での式 a_1 と a_2 が文脈同値であること $\Gamma \vdash a_1 \cong a_2$ は以下の条件が成立することと同値である。

$$\Gamma \leq \Delta, n \in \mathbf{Nv}, \Delta \vdash C \in \mathbf{Ctx} \text{ が成立する任意の } \Delta, n, C \text{ に対し, } \Delta \vdash C[a_1] \Downarrow n \iff \Delta \vdash C[a_2] \Downarrow n \text{ である。}$$

2.3 簡約関係

簡約関係も、評価関係と同様に、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

$$\begin{aligned} x=Z, y=S(x) \vdash y + x &\longrightarrow S(x) + x \\ x=Z, y=S(x) \vdash y + x &\longrightarrow y + Z \end{aligned}$$

といった判断が考えられる。

さて、 $x=Z, x=S(x)$ という判断の下で変数参照 x (正確には $\#^0 x$ は何に簡約されるだろうか?これを素朴に $S(x)$ 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、 $x=Z, x=S(x)$ の下での $S(x)$ 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、

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

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、 $x=Z$ の下での算術式 $S(x)$ 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。

$\frac{\Gamma \vdash a \in \mathbf{Aexp}}{\Gamma, x = a \vdash x \longrightarrow a \uparrow_x} \quad (\text{E-DEF})$	$\frac{\Gamma \vdash a_1 \in \mathbf{Aexp} \quad \Gamma \vdash a_2 \in \mathbf{Aexp}}{\Gamma \vdash a_1 * \mathbf{S}(a_2) \longrightarrow a_1 * a_2 + a_1} \quad (\text{E-MULTSUCC})$
$\frac{\Gamma \vdash \#^i x \longrightarrow a}{\Gamma, y \vdash \#^i x \uparrow_y \longrightarrow a \uparrow_y} \quad (\text{E-SHIFT1})$	$\frac{\Gamma \vdash a \longrightarrow a'}{\Gamma \vdash \mathbf{S}(a) \longrightarrow \mathbf{S}(a')} \quad (\text{E-SUCC})$
$\frac{\Gamma \vdash \#^i x \longrightarrow a \quad \Gamma \vdash a' \in \mathbf{Aexp}}{\Gamma, y = a' \vdash \#^i x \uparrow_y \longrightarrow a \uparrow_y} \quad (\text{E-SHIFT2})$	$\frac{\Gamma \vdash a_1 \longrightarrow a'_1}{\Gamma \vdash a_1 + a_2 \longrightarrow a'_1 + a_2} \quad (\text{E-PLUSL})$
$\frac{\Gamma \vdash a_0 \in \mathbf{Aexp}}{\Gamma \vdash a_0 + \mathbf{Z} \longrightarrow a_0} \quad (\text{E-PLUSZERO})$	$\frac{\Gamma \vdash a_2 \longrightarrow a'_2}{\Gamma \vdash a_1 + a_2 \longrightarrow a_1 + a'_2} \quad (\text{E-PLUSR})$
$\frac{\Gamma \vdash a_1 \in \mathbf{Aexp} \quad \Gamma \vdash a_2 \in \mathbf{Aexp}}{\Gamma \vdash a_1 + \mathbf{S}(a_2) \longrightarrow \mathbf{S}(a_1 + a_2)} \quad (\text{E-PLUSSUCC})$	$\frac{\Gamma \vdash a_1 \longrightarrow a'_1}{\Gamma \vdash a_1 * a_2 \longrightarrow a'_1 * a_2} \quad (\text{E-MULTL})$
$\frac{\Gamma \vdash a_0 \in \mathbf{Aexp}}{\Gamma \vdash a_0 * \mathbf{Z} \longrightarrow \mathbf{Z}} \quad (\text{E-MULTZERO})$	$\frac{\Gamma \vdash a_2 \longrightarrow a'_2}{\Gamma \vdash a_1 * a_2 \longrightarrow a_1 * a'_2} \quad (\text{E-MULTR})$

図 3: 簡約関係: $\Gamma \vdash a \longrightarrow a'$

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。

$$\begin{aligned}
\#^i x \uparrow_x &= \#^{i+1} x \\
\#^i x \uparrow_y &= \#^i x \quad (y \neq x) \\
0 \uparrow_x &= 0 \\
\mathbf{S}(a_0) \uparrow_x &= \mathbf{S}(a_0 \uparrow_x) \\
(a_1 + a_2) \uparrow_x &= (a_1 \uparrow_x) + (a_2 \uparrow_x) \\
(a_1 * a_2) \uparrow_x &= (a_1 \uparrow_x) * (a_2 \uparrow_x)
\end{aligned}$$

$a \uparrow_x$ 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。

2.3.1 定義: 簡約関係 $\Gamma \vdash a \longrightarrow a'$ は図 3 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。)

2.3.2 例: 以下の判断が導出できる。

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

2.4 簡約関係の性質

停止性, 合流性が成立する.

3 ふたつの意味の関係

簡約関係から導かれる同値な式は文脈同値である.

3.1 定理: 任意の $\Gamma \in \mathbf{Env}$, $\Gamma \vdash a_1 \in \mathbf{Aexp}$, $\Gamma \vdash a_2 \in \mathbf{Aexp}$ に対し, $\Gamma \vdash a_1 \leftrightarrow a_2 \Rightarrow \Gamma \vdash a_1 \cong a_2$.

が, 逆は成立しない. 例えば \bullet, x の下で $x + S(Z)$ と $S(Z) + x$ は文脈同値であるが, \leftrightarrow の関係にはない.