2a. 自然意味論

z:=x; x:=y; y:=z の実行

$$\frac{\dfrac{\langle \mathrm{z{:=}x},\ s_0\rangle \to s_1 \qquad \langle \mathrm{x{:=}y},\ s_1\rangle \to s_2}{\langle \mathrm{z{:=}x;\ x{:=}y},\ s_0\rangle \to s_2} \qquad \langle \mathrm{y{:=}z},\ s_2\rangle \to s_3}{\langle \mathrm{z{:=}x;\ x{:=}y;\ y{:=}z},\ s_0\rangle \to s_3}$$

- ただし,

$$\begin{array}{lcl} s_0 & = & [\mathrm{x}\mapsto\mathbf{5},\ \mathrm{y}\mapsto\mathbf{7},\ \mathrm{z}\mapsto\mathbf{0}] \\ s_1 & = & [\mathrm{x}\mapsto\mathbf{5},\ \mathrm{y}\mapsto\mathbf{7},\ \mathrm{z}\mapsto\mathbf{5}] \\ s_2 & = & [\mathrm{x}\mapsto\mathbf{7},\ \mathrm{y}\mapsto\mathbf{7},\ \mathrm{z}\mapsto\mathbf{5}] \\ s_3 & = & [\mathrm{x}\mapsto\mathbf{7},\ \mathrm{y}\mapsto\mathbf{5},\ \mathrm{z}\mapsto\mathbf{5}] \end{array}$$

文の自然意味論を定める

- 自然意味論(Natural semantics/big-step semantics)：与えられた文と状態に対して(存在するとしたら)最終状態は何か？
- よってS∈Stmとs, s'∈Stateに対して遷移関係は <S,s> → s' と表される.
  - <S,s>　文Sは状態sにおいて実行される
  - s'　　　最終状態を表す
- 例：
  <**if x ≤ 1 then x** := **2 else skip**, [**x**↦0]> → [**x**↦2]

$[\mathrm{ass}_{\mathrm{ns}}]$	$\langle x := a,\ s\rangle \to s[x \mapsto \mathcal{A}[\![a]\!]s]$	
$[\mathrm{skip}_{\mathrm{ns}}]$	$\langle \mathtt{skip},\ s\rangle \to s$	
$[\mathrm{comp}_{\mathrm{ns}}]$	$\dfrac{\langle S_1,\ s\rangle \to s',\ \langle S_2,\ s'\rangle \to s''}{\langle S_1;S_2,\ s\rangle \to s''}$	
$[\mathrm{if}^{\mathrm{tt}}_{\mathrm{ns}}]$	$\dfrac{\langle S_1,\ s\rangle \to s'}{\langle \mathtt{if}\ b\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2,\ s\rangle \to s'}$	if $\mathcal{B}[\![b]\!]s = \mathbf{tt}$
$[\mathrm{if}^{\mathrm{ff}}_{\mathrm{ns}}]$	$\dfrac{\langle S_2,\ s\rangle \to s'}{\langle \mathtt{if}\ b\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2,\ s\rangle \to s'}$	if $\mathcal{B}[\![b]\!]s = \mathbf{ff}$
$[\mathrm{while}^{\mathrm{tt}}_{\mathrm{ns}}]$	$\dfrac{\langle S,\ s\rangle \to s',\ \langle \mathtt{while}\ b\ \mathtt{do}\ S,\ s'\rangle \to s''}{\langle \mathtt{while}\ b\ \mathtt{do}\ S,\ s\rangle \to s''}$	if $\mathcal{B}[\![b]\!]s = \mathbf{tt}$
$[\mathrm{while}^{\mathrm{ff}}_{\mathrm{ns}}]$	$\langle \mathtt{while}\ b\ \mathtt{do}\ S,\ s\rangle \to s$ if $\mathcal{B}[\![b]\!]s = \mathbf{ff}$	

Table 2.1 Natural semantics for **While**

代入の公理$[ass_{ns}]$

- 公理$[ass_{ns}]$：状態sで**x**:=aを実行した結果は**x**の値がaになるように更新されたsである．
  - 状態sでの値aは意味関数A〚a〛で計算される．
- $[ass_{ns}]$において**x**, a, sは特定の値，算術式，状態にインスタンス化されるメタ変数である．

- 例．**x**がy, aが**z+1**, s_0が$[\mathbf{y} \mapsto 1, \mathbf{z} \mapsto 2]$と仮定する．
  $[ass_{ns}]$のインスタンスは,
  <**x**:=a,s> → s[**x**:=A〚a〛s]

代入の公理[ass_{ns}]

- 状態sで**x**:=aを実行した結果は**x**の値がaになるように更新されたsである．
  - 状態sでの値aは意味関数A〚a〛で計算される．
- [ass_{ns}]において**x**, a, sは特定の値，算術式，状態にインスタンス化されるメタ変数である．

- 例．**x**がy, aが**z+1**, s_0が[$\mathbf{y} \mapsto 1, \mathbf{z} \mapsto 2$]と仮定する．[$ass_{ns}$]のインスタンスは，
  $\langle \mathbf{y} := \mathbf{z+1}, s_0 \rangle \rightarrow s_0[\mathbf{y} \mapsto \mathcal{A}[\![\mathbf{z+1}]\!]\, s_0]$

Skipの公理[skip_{ns}]

- 状態sでの**skip**の実行結果は単に状態s （変化なし）

- 例． sがs_0=[**y** $\mapsto$ 1,**z** $\mapsto$ 2] と仮定する．
  [skip_{ns}]のインスタンスは
  <**skip**,s> → s

Skipの公理[skip_{ns}]

- 状態sでの**skip**の実行結果は単に状態s （変化なし）
- 例．sがs_0=[$\mathbf{y} \mapsto 1, \mathbf{z} \mapsto 2$] と仮定する．
  [skip_{ns}]のインスタンスは
  <**skip**,s_0> → s_0

結合の規則

$$\frac{\langle S_1,s\rangle \rightarrow s' \qquad \langle S_2,s'\rangle \rightarrow s''}{\langle S_1;S_2,s\rangle \rightarrow s''}$$

- 規則[cmp_{ns}]：
  状態sで$S_1;S_2$を実行した結果は以下のように得られる.
  - sでS_1を実行して結果s'を得られる.
  - s'でS_2を実行して全体の結果s''を得られる.

文の自然意味論を定める

- →の定義は以下の形の規則で与えられる：

前提

$$\frac{\langle S_1,\, s_1\rangle \to s'_1,\, \cdots,\, \langle S_\mathrm{n},\, s_\mathrm{n}\rangle \to s'_\mathrm{n}}{\langle S,\, s\rangle \to s'} \quad \text{if } \cdots$$

結論

条件

- SはS_1,…,S_nから構成される
- 前提が満たされ（条件が満たされると）結論が成り立つ
- 前提がない規則を公理という
  - 線を省略する

導出木の構築

- 状態s_0=[**y**↦,**z**↦2]で**skip**; **y**:=**z+1**の実行結果を得たい
- **skip**;**y**:=**z+1**はS_1;S_2の形であるので[comp_{ns}]規則を使う
- インスタンス化する：

$$\frac{\langle S_1, s\rangle \rightarrow s', \ \langle S_2, s'\rangle \rightarrow s''}{\langle S_1; S_2, s\rangle \rightarrow s''}$$

導出木の構築

- 状態s_0=[**y**↦1,**z**↦2]で**skip**; **y**:=**z+1**の実行結果を得たい
- **skip**;**y**:=**z+1**はS_1;S_2の形であるので[comp_{ns}]規則を使う
- インスタンス化する：S_1= **skip**

$$\frac{\langle \mathbf{skip}, s\rangle \rightarrow s',\ \langle S_2, s'\rangle \rightarrow s''}{\langle \mathbf{skip}; S_2, s\rangle \rightarrow s''}$$

導出木の構築

- 状態s_0=[**y**↦1,**z**↦2]で**skip**; **y**:=**z+1**の実行結果を得たい
- **skip**;**y**:=**z+1**はS_1;S_2の形であるので[comp_{ns}]規則を使う
- インスタンス化する：S_1= **skip** かつ S_2=**y:=z+1**

$$\frac{\langle \textbf{skip}, s\rangle \rightarrow s',\ \langle \textbf{y:=z+1}, s'\rangle \rightarrow s''}{\langle \textbf{skip;y:=z+1}, s\rangle \rightarrow s''}$$

導出木の構築

- 状態s_0=[**y**↦1,**z**↦2]で**skip**; **y**:=**z+1**の実行結果を得たい
- **skip;y**:=**z+1**はS_1;S_2の形であるので[comp$_{ns}$]規則を使う
- インスタンス化する：S_1= **skip** かつ S_2=**y:=z+1**かつs=s_0
- 前提を導出する：

$$\frac{\langle \texttt{skip}, s_0\rangle \to s',\ \langle \texttt{y:=z+1}, s'\rangle \to s''}{\langle \texttt{skip;y:=z+1}, s_0\rangle \to s''}$$

導出木の構築

- 状態s_0=[**y**↦1,**z**↦2]で**skip**; **y**:=**z+1**の実行結果を得たい
- **skip**;**y**:=**z+1**はS_1;S_2の形であるので[comp$_{ns}$]規則を使う
- インスタンス化する：S_1= **skip** かつ S_2=**y**:=**z+1**かつs=s_0
- 前提を導出する：

$$\frac{\langle \texttt{skip}, s_0\rangle \rightarrow s_0, \quad \langle \texttt{y:=z+1}, s_0\rangle \rightarrow s''}{\langle \texttt{skip;y:=z+1}, s_0\rangle \rightarrow s''}$$

導出木の構築

- 状態s_0=[**y**↦1,**z**↦2]で**skip**; **y**:=**z+1**の実行結果を得たい
- **skip**;**y**:=**z+1**はS_1;S_2の形であるので[$comp_{ns}$]規則を使う
- インスタンス化する：S_1= **skip** かつ S_2=**y**:=**z+1**かつs=s_0
- 前提を導出する：
  - <**skip**,s_0> → s_0 は[$skip_{ns}$]のインスタンス

$$\frac{\langle \mathbf{skip}, s_0\rangle \rightarrow s_0, \quad \langle \mathbf{y}:=\mathbf{z+1}, s_0\rangle \rightarrow s''}{\langle \mathbf{skip};\mathbf{y}:=\mathbf{z+1}, s_0\rangle \rightarrow s''}$$

導出木の構築

- 状態$s_0=[\mathbf{y}\mapsto 1, \mathbf{z}\mapsto 2]$で**skip**; **y**:=**z+1**の実行結果を得たい
- **skip**;**y**:=**z+1**は$S_1;S_2$の形であるので$[\mathrm{comp}_{ns}]$規則を使う
- インスタンス化する：$S_1=$ **skip** かつ $S_2=$**y**:=**z+1**かつ$s=s_0$
- 前提を導出する：
  - <**skip**,s_0> → s_0 は$[\mathrm{skip}_{ns}]$のインスタンス
  - <**y**:=**z+1**,s_0> → $s_0[\mathbf{y}\mapsto\mathbf{3}]$は$[\mathrm{ass}_{ns}]$のインスタンス

$$\frac{\langle \mathbf{skip}, s_0\rangle \to s_0,\quad \langle \mathbf{y}:=\mathbf{z+1}, s_0\rangle \to s_0[\mathbf{y}\mapsto 3]}{\langle \mathbf{skip};\mathbf{y}:=\mathbf{z+1}, s_0\rangle \to s_0[\mathbf{y}\mapsto 3]}$$

導出木

- 遷移<S,s>→s’を導出するとき，次の導出木を構築する：
  - 根は<S,s>→s’
  - 葉は公理
  - 中間の節（根で無い節）はインスタンス化された規則の結論
- 導出木の構築方法
  - 根から開始する
  - どの規則が適用できるかを探す．文の形や条件が適切な規則を探す．
  - 葉（公理）に達するまで再帰的に規則の各前提を根とする導出木を構築する

if文の規則

$$\frac{\langle S_1,\ s\rangle \to s'}{\langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2,\ s\rangle \to s'}\quad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{tt}$$

$$\frac{\langle S_2,\ s\rangle \to s'}{\langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2,\ s\rangle \to s'}\quad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{ff}$$

- 2つのどちらが適用されるかは条件による
- B〚b〛s = ttなら**then**節であるS_1が実行される
- B〚b〛s = ffなら**else**節であるS_2が実行される

while文の規則

$$[\text{while}^{\text{tt}}_{\text{ns}}] \qquad \frac{\langle S,\ s\rangle \to s',\ \langle \texttt{while}\ b\ \texttt{do}\ S,\ s'\rangle \to s''}{\langle \texttt{while}\ b\ \texttt{do}\ S,\ s\rangle \to s''} \ \text{if}\ \mathcal{B}[\![b]\!]s = \mathbf{tt}$$

$$[\text{while}^{\text{ff}}_{\text{ns}}] \qquad \langle \texttt{while}\ b\ \texttt{do}\ S,\ s\rangle \to s\ \text{if}\ \mathcal{B}[\![b]\!]s = \mathbf{ff}$$

- **`if`**同様に2つの規則
- B［b］ s = tt なら$\text{while}^{\text{tt}}_{\text{ns}}$を適用できる
  - ループの本体を1回実行する
  - 得られた状態でループの実行を継続する
- B［b］ s = ff なら実行を停止させる$\text{while}^{\text{ff}}_{\text{ns}}$を適用できる.
- 注意： $\text{while}^{\text{tt}}_{\text{ns}}$の前提には結論と同一のものがあるので、結合的な定義では無い.

例題1

- 記法：s_{ij} **x** = i かつ s_{ij} **y** = j, 例えば $s_{30}=[\mathbf{x} \mapsto 3, \mathbf{y} \mapsto 0]$
- 次の導出木を構築する：$\langle w, s_{30}\rangle \to s$
  ただし w=**y**:=**1**;$\underline{w}$, $\underline{w}$=**while** ¬(**x**=**1**) **do** S, S= **y**:=**y*x**; **x**:=**x-1**

$$\frac{}{\langle w, s_{30}\rangle \to}$$

例題1

- 記法：s_{ij} $\mathbf{x} = i$ かつ s_{ij} $\mathbf{y} = j$, 例えば $s_{30}=[\mathbf{x} \mapsto 3, \mathbf{y} \mapsto 0]$
- 次の導出木を構築する：$\langle w, s_{30}\rangle \to s$
  ただし $w=\mathbf{y}{:=}\mathbf{1};\underline{w}$, $\underline{w}=\mathbf{while}\ \neg(\mathbf{x}=\mathbf{1})\ \mathbf{do}\ S$, $S=\mathbf{y}{:=}\mathbf{y*x};\ \mathbf{x}{:=}\mathbf{x-1}$

$$
\dfrac{\langle \mathbf{y}{:=}\mathbf{1}, s_{30}\rangle \to s_{31} \quad \langle \underline{w}, s_{31}\rangle \to s_{16}}{\langle w, s_{30}\rangle \to s_{16}}
$$

$$
\dfrac{\dfrac{\langle \mathbf{y}{:=}\mathbf{y*x}, s_{31}\rangle \to s_{33} \quad \langle \mathbf{x}{:=}\mathbf{x-1}, s_{33}\rangle \to s_{23}}{\langle S, s_{31}\rangle \to s_{23}} \qquad \dfrac{\dfrac{\langle \mathbf{y}{:=}\mathbf{y*x}, s_{23}\rangle \to s_{26} \quad \langle \mathbf{x}{:=}\mathbf{x-1}, s_{26}\rangle \to s_{16}}{\langle S, s_{23}\rangle \to s_{26} \qquad \langle \underline{w}, s_{26}\rangle \to s_{16}}}{\langle \underline{w}, s_{23}\rangle \to s_{16}}}{}
$$

例題2

S=**n**:=**2**;**sum**:=**0**;**i**:=**1**;**while i** ≤ **n do** (**sum**:=**sum+i**;**i**:=**i+1**)

<S,[**n** ↦ **-1**,**sum** ↦ **15**,**i** ↦ **26**]> → ?

例題2

b=**sum**:=**sum+i**;**i**:=**i+1**, w=**while i** ≤ **n do** b, S=**n**:=**2**;**sum**:=**0**;**i**:=**1**;w

$$\dfrac{\langle n:=2,[n \mapsto -1,sum \mapsto 15,i \mapsto 26]\rangle \to ? \qquad \dfrac{\langle sum:=0,?\rangle \to ? \qquad \dfrac{\langle i:=1,?\rangle \to ? \qquad \dfrac{\Pi 1}{\langle w,?\rangle \to ?}}{\langle i:=1;w,?\rangle \to ?}}{\langle sum:=0;i:=1;w,?\rangle \to ?}}{\langle S,[n \mapsto -1,sum \mapsto 15,i \mapsto 26]\rangle \to ?}$$

例題2

b=**sum**:=**sum+i**;**i**:=**i+1**, w=**while i** ≤ **n do** b, S=**n**:=**2**;**sum**:=**0**;**i**:=**1**;w

$$\dfrac{\langle n:=2,[n\mapsto -1,sum\mapsto 15,i\mapsto 26]\rangle\to[n\mapsto 2,sum\mapsto 15,i\mapsto 26] \quad \dfrac{\langle sum:=0,[n\mapsto 2,sum\mapsto 15,i\mapsto 26]\rangle\to[n\mapsto 2,sum\mapsto 0,i\mapsto 26] \quad \dfrac{\langle i:=1,[n\mapsto 2,sum\mapsto 15,i\mapsto 26]\rangle\to[n\mapsto 2,sum\mapsto 0,i\mapsto 1] \quad \dfrac{\Pi 1}{\langle w,[n\mapsto 2,sum\mapsto 0,i\mapsto 1]\rangle\to[n\mapsto 2,sum\mapsto 3,i\mapsto 3]}}{\langle i:=1;w,[n\mapsto 2,sum\mapsto 0,i\mapsto 26]\rangle\to[n\mapsto 2,sum\mapsto 3,i\mapsto 3]}}{\langle sum:=0;i:=1;w,[n\mapsto 2,sum\mapsto 15,i\mapsto 26]\rangle\to[n\mapsto 2,sum\mapsto 3,i\mapsto 3]}}{\langle S,[n\mapsto -1,sum\mapsto 15,i\mapsto 26]\rangle \to [n\mapsto 2,sum\mapsto 3,i\mapsto 3]}$$

Π1=

$$\dfrac{\dfrac{\langle sum:=sum+i,[n\mapsto 2,sum\mapsto 0,i\mapsto 1]\rangle\to[n\mapsto 2,sum\mapsto 1,i\mapsto 1] \quad \langle i:=i+1,[n\mapsto 2,sum\mapsto 1,i\mapsto 1]\rangle\to[n\mapsto 2,sum\mapsto 1,i\mapsto 2]}{\langle b,[n\mapsto 2,sum\mapsto 0,i\mapsto 1]\rangle\to[n\mapsto 2,sum\mapsto 1,i\mapsto 2]} \quad \dfrac{\Pi 2}{\langle w,[n\mapsto 2,sum\mapsto 1,i\mapsto 2]\rangle\to[n\mapsto 2,sum\mapsto 3,i\mapsto 3]}}{\langle w,[n\mapsto 2,sum\mapsto 0,i\mapsto 1]\rangle\to[n\mapsto 2,sum\mapsto 3,i\mapsto 3]}$$

Π2=

$$\dfrac{\dfrac{\langle sum:=sum+i,[n\mapsto 2,sum\mapsto 1,i\mapsto 2]\rangle\to[n\mapsto 2,sum\mapsto 3,i\mapsto 2] \quad \langle i:=i+1,[n\mapsto 2,sum\mapsto 3,i\mapsto 2]\rangle\to[n\mapsto 2,sum\mapsto 3,i\mapsto 3]}{\langle b,[n\mapsto 2,sum\mapsto 1,i\mapsto 2]\rangle\to[n\mapsto 2,sum\mapsto 3,i\mapsto 3]} \quad \langle w,[n\mapsto 2,sum\mapsto 3,i\mapsto 3]\rangle\to[n\mapsto 2,sum\mapsto 3,i\mapsto 3]}{\langle w,[n\mapsto 2,sum\mapsto 1,i\mapsto 2]\rangle\to[n\mapsto 2,sum\mapsto 3,i\mapsto 3]}$$

停止とループ

- 状態sからのSの実行は
  - s'が存在した<S,s>→s'となるときかつそのときに限り停止するという
  - <S,s>→s'となるs'が存在しないときかつそのときに限りループするという
- 任意のsに対して実行が停止する場合,
  その文は常に停止するという
- 任意のsに対して実行がループする場合,
  その文は常にループするという

等価性

- 形式的に言語の意味を定めると文やその性質を議論できるようになる

定義
2つの文S1とS2は以下の場合に等価という.
任意の状態sとs'に対して
<S1,s>→s'のときかつそのときに限り<S2,s>→s'

等価性

補題 2.5
文

$$\texttt{\textbf{while}}\ b\ \texttt{\textbf{do}}\ S \qquad (*)$$

は

$$\texttt{\textbf{if}}\ b\ \texttt{\textbf{then}}\ (S;\ \texttt{\textbf{while}}\ b\ \texttt{\textbf{do}}\ S)\ \texttt{\textbf{else}}\ \texttt{\textbf{skip}} \qquad (**)$$

と等価である。

- 証明では両方向を示す必要がある.
  - “⇒” 2つのケース: $[\text{while}^{tt}_{ns}]$ or $[\text{while}^{ff}_{ns}]$
  - “⇐” 2つのケース: $[\text{if}^{tt}_{ns}]$ or $[\text{if}^{ff}_{ns}]$

導出木の構造に関する帰納法

- 今見た証明法は次のように一般化できる.

- 全ての公理に対して性質が成り立つことを示す.
- それ以外の全ての規則に対して性質が成り立つことを示す.
  - その前提では性質が成り立つことを仮定する.
    この仮定を帰納法の仮定という.
  - 結論で成り立つことを証明する.
    （条件があればそれらも成立していることを仮定する）

決定性

定義　自然意味論は，任意の文Sと状態s, s', s''に対して

$$\langle S,s \rangle \to s' \text{ かつ } \langle S,s \rangle \to s'' \text{ ならば } s'=s''$$

がなりたつとき決定的という．

任意の文Sと初期状態sに対して最終状態s'が（もしも停止するならば）一意に定まることを示している．

定理　While言語の自然意味論は決定的である．
証明は導出木の構造に関する帰納法で示せる．

結合的定義

- 注意：前の定理は構造帰納法では証明できない．
- 理由：意味論が結合的に定義されていない．規則$\text{while}^{tt}{}_{ns}$では結論と前提で同じものがある．

- 導出木の構造に関する帰納法は，導出木の構造帰納法のようなもの．
  - 基底段階：単純な木に関する証明をする
  - 帰納段階：部分木に関しての仮定から全体の木に関する証明をする

文の意味関数

- 文の意味は状態から状態への部分関数
  S_{ns} : Stm → (State → State)
- 部分関数：領域の全要素に対して定義されている必要は無い
- 定義：
  $$S_{ns}[\![S]\!]\ s = \begin{cases} s' & \text{if } \langle S,s\rangle \to s' \\ \underline{\text{undef}} & \text{otherwise} \end{cases}$$

- なぜ部分関数か？
  `while true do skip`　のような停止しない文があるから.

まとめ

- 文の自然意味論
- 証明法：導出木の構造に関する帰納法