プログラミング言語の基礎概念

導出システム (derivation system) … プログラミング言語の意味論 (semantics) や型システムを記述するための枠組み

自然演繹(←数理論理学で形式的証明を記述するための)やシーケント計算といった体系を一般化した汎用的な枠組み

議論の対象(論理式、プログラム、型など)に対する様々な判断を推論規則に従って導くための記述体系

判断の形式とそれに対する推論規則群を与えることによって定められる

例) 自然数の大小比較を対象とした導出システム
判断の例: 3は5より小さい
推論規則の例: 任意の自然数 n_1, n_2, n_3 について、n_1 が n_2 より小さく、かつ n_2 が n_3 より小さいならば、n_1 は n_3 より小さい

1.1 自然数の加算・乗算: 導出システム Nat

定義 1.1 導出システム Nat で我々が扱う判断の一般的な形式は
n_1, n_2, n_3 をペアノ自然数
- n_1 plus n_2 is n_3
- n_1 times n_2 is n_3

のいずれかとする。

※ 形式上は正しいが内容的には正しくない判断も考えてよい
例) $S(S(z))$ plus $S(S(S(z)))$ is $S(z)$
今後、単に「判断」といったら、形式的に正しい判断を指す

定義 1.2 判断 "n_1 plus n_2 is n_3" のための推論規則は次のふたつである

規則 P-ZERO: 任意のペアノ自然数 n に対して、
n plus n_2 is n_3 を導いてよい.

規則 P-SUCC: 任意のペアノ自然数 n_1, n_2, n_3 に対して、
n_1 plus n_2 is n_3 ならば
$S(n_1)$ plus n_2 is $S(n_3)$ を導いてよい.

規則 P-SUCC のような、「J_1 ならば J_2」という形の、前提 (premise) のある推論規則は、前提となる判断 J_1 が既に得られている場合に限り J_2 の判断を導いてよい

定義 1.3 判断 "n_1 times n_2 is n_3" のための推論規則は以下のふたつ

規則 T-ZERO: z times (n) is z （任意のペアノ自然数）

規則 T-SUCC: n_1 times n_2 is n_3 かつ n_2 plus n_3 is n_4 ならば、$\mathsf{S}(n_1)$ times n_2 is n_4

- $n_1 \times n_2 = n_3$
- $n_2 + n_3 = n_4$

$\boxed{\mathsf{S}(n_1) \times n_2 = n_4}$

$n_2 + (n_1 \times n_2) = n_4$

$n_2(n_1 + 1) = n_4$ （$n_1 + 1 = \mathsf{S}(n_1)$）

$\mathsf{S}(\mathsf{S}(\mathsf{z}))$ times $\mathsf{S}(\mathsf{S}(\mathsf{z}))$ is $\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{z}))))$

規則 T-ZERO より z times $\mathsf{S}(\mathsf{S}(\mathsf{z}))$ is z … ①

〃 z plus z is z

P-SUCC → $\mathsf{S}(\mathsf{z})$ plus z is $\mathsf{S}(\mathsf{z})$

P-SUCC → $\mathsf{S}(\mathsf{S}(\mathsf{z}))$ plus z is $\mathsf{S}(\mathsf{S}(\mathsf{z}))$ … ②

規則 T-SUCC、①、② より $\mathsf{S}(\mathsf{z})$ times $\mathsf{S}(\mathsf{S}(\mathsf{z}))$ is $\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{z})))$ … ③

規則 P-ZERO より z plus $\mathsf{S}(\mathsf{S}(\mathsf{z}))$ is $\mathsf{S}(\mathsf{S}(\mathsf{z}))$

P-SUCC → $\mathsf{S}(\mathsf{z})$ plus $\mathsf{S}(\mathsf{S}(\mathsf{z}))$ is $\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{z})))$

P-SUCC → $\mathsf{S}(\mathsf{S}(\mathsf{z}))$ plus $\mathsf{S}(\mathsf{S}(\mathsf{z}))$ is $\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{z}))))$ … ④

規則 T-SUCC、③、④ より $\mathsf{S}(\mathsf{S}(\mathsf{z}))$ times $\mathsf{S}(\mathsf{S}(\mathsf{z}))$ is $\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{z}))))$ 終

規則の適用によって判断を導く過程を導出 (derivation) と呼ぶ。

練習 1.2 (1) $\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{z})))$ plus $\mathsf{S}(\mathsf{z})$ is $\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{z}))))$

$n = \mathsf{S}(\mathsf{z})$ とすると P-ZERO より z plus $\mathsf{S}(\mathsf{z})$ is $\mathsf{S}(\mathsf{z})$

P-SUCC より $\mathsf{S}(\mathsf{z})$ plus $\mathsf{S}(\mathsf{z})$ is $\mathsf{S}(\mathsf{S}(\mathsf{z}))$

P-SUCC より $\mathsf{S}(\mathsf{S}(\mathsf{z}))$ plus $\mathsf{S}(\mathsf{z})$ is $\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{z})))$

P-SUCC より $\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{z})))$ plus $\mathsf{S}(\mathsf{z})$ is $\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{z}))))$

プログラミング言語の基礎概念

(2) $S(Z)$ plus $S(S(S(Z)))$ is $S(S(S(S(Z))))$

$n = S(S(S(Z)))$ と P-ZERO より Z plus $S(S(S(Z)))$ is $S(S(S(Z)))$

P-SUCC より $S(Z)$ plus $S(S(S(Z)))$ is $S(S(S(S(Z))))$

(3) メモ
- $S(\underset{n_1}{S(S(Z))})$ times $\underset{n_2}{Z}$ is $\underset{n_4}{Z}$
- $\underset{n_1}{S(S(Z))}$ times $\underset{n_2}{Z}$ is $\underset{n_3}{Z}$
- $\underset{n_2}{Z}$ plus $\underset{n_3}{Z}$ is $\underset{n_4}{Z}$

$n = Z$ とすると P-ZERO より Z plus Z is Z …①

メモ
- $S(Z)$ times Z is Z
- Z times Z is Z $(n = Z)$

$n = Z$ とすると T-ZERO より Z times Z is Z …②

規則 T-SUCC, ①, ② より $S(Z)$ times Z is Z …③

規則 T-SUCC, ①, ③ より $S(S(Z))$ times Z is Z …④

規則 T-SUCC, ①, ④ より $S(S(S(Z)))$ times Z is Z 終

練習 1.3 判断 "n_1 plus n_2 is n_3" が導出できる時、その導出には何回の規則の適用が必要か？

1回目の適用
P-ZERO　Z plus (n) (n_2) is (n) (n_2)　$(n_1 + 1)$ 回

"n_1 times n_2 is n_3" についてはどうか.

$(n_1 + 1)$ 回、T-SUCC（最後の1回だけはT-ZERO）を適用する
T-SUCC を適用するごとに、$(n_2 + 1)$ 回の適用が必要になる

T-SUCC & T-ZERO
$(n_1 + 1) + (n_2 + 1) \times (n_1 + 1)$

1.2 推論規則と導出の記法

規則 X : J_1 かつ … かつ J_n ならば J_0。 $\dfrac{J_1 \quad \cdots \quad J_n}{J_0}$