Hoare 論理

– プログラム証明と構築のための手法と論理 –

胡 振江

東京大学 計数工学科

2008 年夏学期

Outline

参考文献

- 1969 年、C.A.R Hoare は、プログラムが仕様に関して部分正当であることを証明するための公理的手法を導入した。

  C.A. R. Hoare, An Axomatic Basis for Computer Programming, CACM 12 (10), 1969, 576-580.

  計算機科学においてもっとも広く引用されている文献の一つ。

部分的正当性の表明

- Hoare Triplet

$$\{P\}\, S\, \{Q\}$$

事前条件 P を満足する時に、
プログラム S を実行すると、
その実行後には, 事後条件 Q を満足する。

プログラム S が終了すれば, プログラム実行の効果として、
事前条件と事後条件との対によって表現した意図通りの結果
が得られる。

簡単な言語

$$
\begin{array}{lcll}
S & ::= & x := e & \{\ \text{代入文}\ \} \\
 & | & S_1 ; S_2 & \{\ \text{複合文}\ \} \\
 & | & \textbf{if}\ B\ \textbf{then}\ S_1\ \textbf{else}\ S_2\ \textbf{end} & \{\ \text{if 文}\ \} \\
 & | & \textbf{while}\ B\ \textbf{do}\ S\ \textbf{end} & \{\ \text{while 文}\ \}
\end{array}
$$

プログラムの例: 階乗の計算

$\{N \geq 0\}$
i := 1;
f := 1;
while i ≤ N **do**
  f := f*i;
  i := i+1
end
$\{f=N!\}$

Hoare 論理

- 第一階述語論理の拡張
- プログラムにかかわる公理と推論規則
  - 代入文の公理
  - 複合文の規則
  - if 文の規則
  - while 文の規則
  - 帰結の規則

代入文の公理

- 公理

$$\{Q[e/x]\}\, x := e\, \{Q\}$$

代入文 x := e の実行後に事後条件 Q が成り立つには事前条件として Q[e/x] が成り立つ必要がある。

- 例

$$\{x > 9\}\, x := x + 1\, \{x > 10\}$$

複合文の推論規則

- 規則

$$\frac{\{P\}\, S_1\, \{R\} \quad \{R\}\, S_2\, \{Q\}}{\{P\}\, S_1; S_2\, \{Q\}}$$

帰結としての部分的正当性 $\{P\}\, S_1; S_2\, \{Q\}$ が導かれるためには、前提として各文について部分的正当性 $\{P\}\, S_1\, \{R\}$ と $\{R\}\, S_2\, \{Q\}$ が成り立つ必要

- 例

$$\frac{\{x > 7\}\, x := x + 2\, \{x > 9\} \quad \{x > 9\}\, x := x + 1\, \{x > 10\}}{\{x > 7\}\, x := x + 2; x := x + 1\, \{x > 10\}}$$

if 文の規則

- 規則

$$\frac{\{P \wedge B\}\, S_1\, \{Q\} \quad \{P \wedge \neg B\}\, S_2\, \{Q\}}{\{P\}\ \textbf{if}\ B\ \textbf{then}\ S_1\ \textbf{else}\ S_2\ \textbf{end}\ \{Q\}}$$

- 例

$$\frac{\{\mathit{True} \wedge x < 0\}\, x := -x\, \{x > 0\} \quad \{\mathit{True} \wedge x \geq 0\}\, x := x\, \{x \geq 0\}}{\{\mathit{True}\}\ \textbf{if}\ x < 0\ \textbf{then}\ x := -x\ \textbf{else}\ x := x\ \textbf{end}\ \{x \geq 0\}}$$

while 文の規則

- 規則

$$\frac{\{P \wedge B\}\ S\ \{P\}}{\{P\}\ \textbf{while}\ B\ \textbf{do}\ S\ \textbf{end}\ \{P \wedge \neg B\}}$$

- P: ループ不変条件 (loop invariant)

  while 文の規則では適切なループ不変条件を見つけることが重要 (一般にはそれほど容易なことではない。)

帰結の規則

- 規則

$$\frac{P \Rightarrow P_1 \quad \{P_1\}\,S\,\{Q_1\} \quad Q_1 \Rightarrow Q}{\{P\}\,S\,\{Q\}}$$

- 例

$$\frac{\{x < 0\}\,x := -x\,\{x > 0\} \quad x > 0 \Rightarrow x \geq 0}{\{x < 0\}\,x := -x\,\{x \geq 0\}}$$

部分正当性の証明の構成法

- 所望の部分的正当性の表明を目標として、topdown 向きに証明を構成
  - 目標となる部分的正当性の表明を導出するため、どの推論規則を用いる?
  - その場合の前提として成り立つ必要がある論理式はどのようなもの?

例題：要素の交換

$\{x = x_0 \wedge y = y_0\}$
t := x;

x := y;

y := t;
$\{x = y_0 \wedge y = x_0\}$

例題：要素の交換

$\{x = x_0 \land y = y_0\}$
t := x;
$\{y = y_0 \land t = x_0\}$
x := y;
$\{x = y_0 \land t = x_0\}$
y := t;
$\{x = y_0 \land y = x_0\}$

例題：階乗

$\{N > 0\}$
i := 1;

f := 1;

while i $\leq$ N **do**

  f := f * i;

  i := i+1;

end
$\{f = N!\}$

例題：階乗

$\{N > 0\}$
i := 1;

f := 1;
$\{1 \leq i \leq N + 1 \wedge f = (i - 1)!\}$
while i $\leq$ N **do**

    f := f * i;

    i := i+1;

end
$\{f = N!\}$

例題：階乗

$\{N > 0\}$
i := 1;
$\{1 \leq i \leq N + 1 \wedge 1 = (i - 1)!\}$
f := 1;
$\{1 \leq i \leq N + 1 \wedge f = (i - 1)!\}$
while i $\leq$ N **do**
  $\{1 \leq i \leq N + 1 \wedge f = (i - 1)! \wedge i \leq N\}$
  f := f * i;
  $\{1 \leq i + 1 \leq N + 1 \wedge f = (i + 1 - 1)!\}$
  i := i+1;
  $\{1 \leq i \leq N + 1 \wedge f = (i - 1)!\}$
end
$\{f = N!\}$

例題：最大公約数

$\{x > 0 \land y > 0\}$
t_1 := x; t_2 := y;

while $t_1 \neq t_2$ **do**
  if $t_1 > t_2$ **then**

    $t_1 := t_1 - t_2;$

  else

    $t_2 := t_2 - t_1;$

  end
end
$\{t_1 = \mathrm{gcd(x,y)}\}$

例題：最大公約数

$\{x > 0 \wedge y > 0\}$

t_1 := x; t_2 := y;

$\{t_1 > 0 \wedge t_2 > 0 \wedge gcd(x, y) = gcd(t_1, t_2)\}$

while $t_1 \neq t_2$ **do**

  if $t_1 > t_2$ **then**

    $t_1 := t_1 - t_2;$

  else

    $t_2 := t_2 - t_1;$

  end

end

$\{t_1 = \mathrm{gcd(x,y)}\}$

例題：最大公約数

$\{x > 0 \wedge y > 0\}$
t_1 := x; t_2 := y;
$\{t_1 > 0 \wedge t_2 > 0 \wedge gcd(x, y) = gcd(t_1, t_2)\}$
while $t_1 \neq t_2$ **do**
  if $t_1 > t_2$ **then**
    $\{t_1 > t_2 \wedge t_1 > 0 \wedge t_2 > 0 \wedge gcd(x, y) = gcd(t_1, t_2)\}$
    $t_1 := t_1 - t_2$;
    $\{t_1 > 0 \wedge t_2 > 0 \wedge gcd(x, y) = gcd(t_1, t_2)\}$
  else
    $\{t_1 < t_2 \wedge t_1 > 0 \wedge t_2 > 0 \wedge gcd(x, y) = gcd(t_1, t_2)\}$
    $t_2 := t_2 - t_1$;
    $\{t_1 > 0 \wedge t_2 > 0 \wedge gcd(x, y) = gcd(t_1, t_2)\}$
  end
end
$\{t_1 = \mathrm{gcd(x,y)}\}$

Outline

Dijkstra's Weakest Precondition (WP)

- $wp(S, Q)$: the set of initial states that guarantee termination of S in a state satisfying Q:

$$\frac{P \Rightarrow wp(S, Q)}{\{P\}\, S\, \{Q\}}$$

WP の定義

$$
\begin{array}{lcl}
wp(x := e, Q) & = & \{Q[e/x]\} \\
wp(S_1; S_2, Q) & = & wp(S_1, wp(S_2, Q)) \\
wp(\textbf{if}\ B\ \textbf{then}\ S_1\ \textbf{else}\ S_2\ \textbf{end}, Q) & = & B \Rightarrow wp(S_1, Q) \wedge \\
 & & \neg B \Rightarrow wp(S_2, Q) \\
wp(\textbf{while}\ B\ \textbf{do}\ S\ \textbf{end}, Q) & = & \exists k : k \geq 0.\ P_k \\
 & & \textbf{where} \\
 & & \quad P_0 = \neg B \wedge Q \\
 & & \quad P_k = B \wedge wp(S, P_{k-1})
\end{array}
$$

WP's Healthiness Conditions

$$
\begin{aligned}
wp(S, Q \land R) &= wp(S, Q) \land wp(S, R) \\
wp(S, Q \lor R) &= wp(S, Q) \lor wp(S, R) \\
wp(S, \neg Q) &= \neg wp(S, Q) \\
\\
wp(S, \mathit{false}) &= \mathit{false} \\
wp(S, \mathit{true}) &= \text{condition for } S \text{ to be terminate}
\end{aligned}
$$

例題 1

$$
\begin{aligned}
& wp(x := x + 1; y := y + 1, x = y) \\
= \quad & wp(x := x + 1, wp(y := y + 1, x = y)) \\
= \quad & wp(x := x + 1, x = y + 1) \\
= \quad & x + 1 = y + 1 \\
= \quad & x = y
\end{aligned}
$$

例題 2

$$
\begin{aligned}
& wp(\textbf{if}\ i = j\ \textbf{then}\ m := k\ \textbf{else}\ j := k\ \textbf{end}, k = j = m) \\
= \ & (i = j \Rightarrow wp(m := k, k = j = m)) \wedge \\
& (i \neq j \Rightarrow wp(j := k, k = j = m)) \\
= \ & (i = j \Rightarrow k = j = k) \wedge (i \neq j \Rightarrow k = k = m) \\
= \ & (i = j \Rightarrow k = j) \wedge (i \neq j \Rightarrow k = m)
\end{aligned}
$$

例題 3

Let

$$\begin{aligned} W &\equiv \textbf{while } n \neq m \textbf{ do } S \textbf{ end} \\ S &\equiv j := j * i; k := k + j; n := n + 1 \\ Q &\equiv k = \frac{i^{m+1}-1}{i-1} \ \wedge \ j = i^m \end{aligned}$$

where $i \neq 0$ and $i \neq 1$.

$$\begin{aligned} P_0 &= \neg(n \neq m) \wedge Q \\ &= m = n \ \wedge \ k = \frac{i^{m+1}-1}{i-1} \ \wedge \ j = i^m \end{aligned}$$

$$\begin{aligned} P_1 &= n \neq m \wedge wp(S, P_0) \\ &= n = m - 1 \wedge k = \frac{i^m-1}{i-1} \ \wedge \ j = i^n \end{aligned}$$

$$\begin{aligned} P_r &= n \neq m \wedge wp(S, P_{r-1}) \\ &= n = m - r \wedge k = \frac{i^m-1}{i-1} \ \wedge \ j = i^n \end{aligned}$$

例題 3(続)

Let

$$
\begin{array}{lcl}
W & \equiv & \textbf{while } n \neq m \textbf{ do } S \textbf{ end} \\
S & \equiv & j := j * i; k := k + j; n := n + 1 \\
Q & \equiv & k = \frac{i^{m+1}-1}{i-1} \ \wedge \ j = i^m
\end{array}
$$

where $i \neq 0$ and $i \neq 1$.

$$
\begin{array}{lcl}
wp(W, Q) & = & \exists r : r \geq 0.\, P_r \\
 & = & \exists r : r \geq 0.\, n = m - r \wedge k = \frac{i^m-1}{i-1} \ \wedge \ j = i^n
\end{array}
$$

Outline

参考資料

Program Construction : Calculating Implementations from Specifications, by Roland Backhouse, ISBN: 0-470-84882-0, 352 pages, May 2003, US $45.00
９章, １０章, １３章

(正しい) プログラムの構成

Given $\{P\}$, $\{Q\}$, construct a program S such that

$$\{P\}\,S\,\{Q\}$$

- 仕様：$\{P\}\,S\,\{Q\}$ (S は未知)
- 実現：S の導出

仕様の例

- $\{true\}\, S\, \{i = j\}$
- $\{i < 5\}\, S\, \{i < 10\}$
- $\{i + j = C\}\, i := i + 1; S\, \{i + j = C\}$
- $\{s = n^2\}\, S; n := n + 1\, \{s = n^2\}$
- $\{true\}\, S\, \{z = max(x, y)\}$
- $\{0 < m = M\}\, S\, \{m = M \div 2\}$
- $\{0 < N\}\, S\, \{s = \Sigma i \mid 0 \leq i < N : a[i]\}$
- $\{0 < N\}\, S\, \{s = \Sigma i \mid 0 \leq i < N : a[i] * X^i\}$

(C,M,N,X は定数)

代入文の導出

- 問題：仕様 $\{P\}\, x := e\, \{Q\}$ を満たす e を導出せよ。
- 方法：$P \Rightarrow wp(x := e, Q)$ を満たす e を導出する。

例 1

- 仕様：$\{true\}\ i := e\ \{i = j\}$
- 導出：
  1. $wp(i := e, i = j) \equiv e = j$
  2. $true \Rightarrow e = j$ を満たす e を求める。

$$e = j$$

例 1

- 仕様：$\{i+j=C\}\ i := i+1; j := e\ \{i+j=C\}$
- 導出：
  1. $wp(i := i+1; j := e, i+j=C) \equiv i+1+e=C$
  2. $i+j=C \Rightarrow i+1+e=C$ を満たす e を求める。

$$e = j - 1$$

条件文の導出

- 問題：仕様 $\{P\}$ **if** B **then** S_1 **else** S_2 **end** $\{Q\}$ を満たす B, S_1, S_2 を求めよ。
- 導出方法：
  1. B を決める。（人間の知恵が必要である）
  2. $\{P \land B\}\ S_1\ \{Q\}$ を満たす S_1 を求める。
  3. $\{P \land \neg B\}\ S_2\ \{Q\}$ を満たす S_2 を求める。

例

- 仕様：$\{true\}$ **if** B **then** S_1 **else** S_2 **end** $\{z = max(x, y)\}$
- 導出：
  1. $B \equiv x \geq y$
  2. $\{x \geq y\}\ S_1\ \{z = max(x, y)\}$ を満たす S_1 を求める。
     S_1 を $z := e$ のような代入文で実現したい。

$$
\begin{array}{ll}
 & \{x \geq y\}\ z := e\ \{z = max(x, y)\} \\
\Leftarrow & \quad \{\text{ 代入文の実現法 }\} \\
 & x \geq y \Rightarrow wp(z := e, z = max(x, y)) \\
\equiv & \quad \{\text{ def. of wp }\} \\
 & x \geq y \Rightarrow e = max(x, y) \\
\equiv & \quad \{\ x \geq y \equiv x = max(x, y)\ \} \\
 & x = max(x, y) \Rightarrow e = max(x, y) \\
\Leftarrow & \quad \{\text{ simple }\} \\
 & e = x
\end{array}
$$

  3. 同様に、$\{x > y\}\ S_2\ \{z = max(x, y)\}$ を満たす S_2 を求める。

複合文の導出：問題の分割（1／2）

- 例題：$\{0 <= m = M\}\ S\ \{m = M \div 2\}$ を満たす S を導出せよ。ただし、$\div$ はそのまま使てはいけない。m が偶数の時に $m \div 2 = rot(m)$ が成立する。

- 問題の分割：

$$\begin{array}{l} \{0 \leq m = M\} \\ S_1; \\ \{0 \leq m = M \wedge even(m) \wedge P\} \\ \mathrm{m\ :=\ rot(m)} \\ \{m = M \div 2\} \end{array}$$

上を満たす S_1 と P を求める。

複合文の導出：問題の分割（2／2）

P と S_1 の導出：

1. P の導出

$$
\begin{array}{ll}
 & 0 \leq m = M \wedge even(m) \wedge {\color{red}P} \Rightarrow wp(m := rot(m), {\color{blue}m = M \div 2}) \\
\equiv & \quad \{\ \text{def. of wp}\ \} \\
 & 0 \leq m = M \wedge even(m) \wedge {\color{red}P} \Rightarrow rot(m) = M \div 2 \\
\Leftarrow & \quad \{\ \text{rot の性質}\ \} \\
 & P \equiv m \div 2 = M \div 2
\end{array}
$$

2. S_1 の導出：条件文の導出法を利用する。

$$
\begin{array}{l}
{\color{blue}\{0 \leq m = M\}} \\
\textbf{if } even(m) \textbf{ then } S_1' \textbf{ else } S_2' \textbf{ end}; \\
{\color{blue}\{0 \leq m = M \wedge even(m) \wedge m \div 2 = M \div 2\}}
\end{array}
$$

練習問題：S_1' と S_2' を導出せよ。

While 文の導出

- 問題：仕様 $\{P\}\ S; \textbf{while}\ B\ \textbf{do}\ T\ \textbf{end}\ \{Q\}$ を満たす S, B, T を求めよ。
- 問題の細分化：

$$
\begin{array}{l}
\{P\} \\
\mathrm{S}; \\
\{inv \land bf \geq 0\}; \\
\textbf{while}\ \mathrm{B}\ \ \textbf{do} \\
\qquad \{inv \land bf = C > 0 \land B\} \\
\qquad \mathrm{T}; \\
\qquad \{inv \land bf < C\} \\
\textbf{end} \\
\{Q\}
\end{array}
$$

inv: ループ不変性質；bf: bound 関数

While 文の導出手順

1. 次の性質を用いて、Q から inv, B と bf を求める。

$$\neg B \wedge inv \wedge fb \geq 0 \Rightarrow Q$$

2. 次の仕様を満たす S を導出する。

$\{P\}$
S;
$\{inv \wedge bf \geq 0\}$;

3. 次の仕様を満たす T を導出する。

$\{inv \wedge bf = C > 0 \wedge B\}$
T;
$\{inv \wedge bf < C\}$

例：配列要素和を求める問題

- 仕様：

$\{0 < N\}$
S;
while B **do**
    T;
end
$\{s = \Sigma i \mid 0 \le i < N : a[i]\}$

1. Q から inv, B と bf を導出

$$
\begin{array}{ll}
 & s = \Sigma i \,|\, 0 \leq i < N : a[i] \\
\equiv & \quad \{ \text{ introduce } k : 0 \leq k \leq N \; \} \\
 & 0 \leq k \leq N \wedge s = \Sigma i \,|\, 0 \leq i < k : a[i] \wedge k \geq N
\end{array}
$$

$\Rightarrow$

$$
\begin{array}{lll}
inv & \equiv & 0 \leq k \leq N \wedge s = \Sigma i \,|\, 0 \leq i < k : a[i] \\
B & \equiv & \neg(k \geq N) \\
bf & \equiv & N - k
\end{array}
$$

2. S の導出

$\{0 < N\}$
S;
$\{0 \le k \le N \wedge s = \Sigma i \,|\, 0 \le i < k : a[i] \wedge bf \ge 0\};$

$\Rightarrow$

$$S \equiv k := e_1;\ s := e_2$$

$\Rightarrow$

$$S \equiv k := 0;\ s := 0$$

3. T の導出

$\{0 \le k \le N \wedge s = \Sigma i \,|\, 0 \le i < k : a[i] \wedge bf = C > 0 \wedge B\}$
T;
$\{0 \le k \le N \wedge s = \Sigma i \,|\, 0 \le i < k : a[i] \wedge bf < C\}$

$\Rightarrow$

$\{0 \le k \le N \wedge s = \Sigma i \,|\, 0 \le i < k : a[i] \wedge bf = C > 0 \wedge B\}$
T'; k:=k+1;
$\{0 \le k \le N \wedge s = \Sigma i \,|\, 0 \le i < k : a[i] \wedge bf < C\}$

$\Rightarrow$

$$T' \equiv s := s + a[k]$$

配列要素和を求める問題を解くプログラム

$\{0 < N\}$

k:=0; s:=0

while $(k < N)$ **do**

  s := s+a[k]; k := k+1;

end

$\{s = \Sigma i \mid 0 \le i < N : a[i]\}$

Outline

定理証明系

対象を論理的に形式化し、その性質を証明・検証

- システム：Coq, PVS, Isabelle, HOL, ACL2
- 応用例
    - 数学の証明：ラムダ計算
    - ハードウェアの検証
    - プログラミング言語：Java の型システムの健全性
    - 暗号プロトコルの安全性

Coq

- Coq は厳密な証明を扱うことのできるのコンピュータのツールである。

  http://coq.inria.fr/

- 人間は Coq と協力して証明を作ったりチェックしたりすることができる。
  - 関数または命題の定義する。
  - ソフトウェア仕様を示す。
  - 数学的な命題を証明する。
  - 補題を組合せて、大きな定理を証明していく。
- 参考書：

  Yves Bertot, Pierre Casteran, "Innteractive Theorem Proving and Program Development", 2004

Coq の例：Scottish Puzzle

A private club has the following rules:

- Every non-scottish member wears red socks
- Every member wears a kilt or doesn't wear red socks
- The married members don't go out on Sunday
- A member goes out on Sunday if and only if he is Scottish
- Every member who wears a kilt is Scottish and married
- Every scottish member wears a kilt

Now, we show with Coq that these rules are so strict that no one can be accepted.

Coq の例：Scottish Puzzle

A private club has the following rules:

- Every non-scottish member wears red socks
- Every member wears a kilt or doesn't wear red socks
- The married members don't go out on Sunday
- A member goes out on Sunday if and only if he is Scottish
- Every member who wears a kilt is Scottish and married
- Every scottish member wears a kilt

Now, we show with Coq that these rules are so strict that no one can be accepted.

Coq の例：Scottish Puzzle

Coq < Section club.

Coq < Variables Scottish RedSocks WearKilt Married GoOutSunday : Pr
Scottish is assumed
RedSocks is assumed
WearKilt is assumed
Married is assumed
GoOutSunday is assumed

Coq < Hypothesis rule1 : $\neg$ Scottish $\rightarrow$ RedSocks.
rule1 is assumed

Coq < Hypothesis rule2 : WearKilt $\vee$ $\neg$ RedSocks.
rule2 is assumed

Coq の例：Scottish Puzzle

Coq < Hypothesis rule3 : Married $\rightarrow \neg$ GoOutSunday.
rule3 is assumed

Coq < Hypothesis rule4 : GoOutSunday $\leftrightarrow$ Scottish.
rule4 is assumed

Coq < Hypothesis rule5 : WearKilt $\rightarrow$ Scottish $\wedge$ Married.
rule5 is assumed

Coq < Hypothesis rule6 : Scottish $\rightarrow$ WearKilt.
rule6 is assumed

Coq の例：Scottish Puzzle

Coq < Lemma NoMember : False.
1 subgoal

Scottish : Prop
RedSocks : Prop
WearKilt : Prop
Married : Prop
GoOutSunday : Prop
rule1 : $\neg$ Scottish $\rightarrow$ RedSocks
rule2 : WearKilt $\vee$ $\neg$ RedSocks
rule3 : Married $\rightarrow$ $\neg$ GoOutSunday
rule4 : GoOutSunday $\leftrightarrow$ Scottish
rule5 : WearKilt $\rightarrow$ Scottish $\wedge$ Married
rule6 : Scottish $\rightarrow$ WearKilt
============================
False

Coq の例：Scottish Puzzle

Coq < tauto.
Proof completed.

Coq < Qed.
tauto.
NoMember is defined

Caduceus

- C プログラムの証明・検証ツールである。

  http://caduceus.lri.fr/

- C プログラムの表明から Coq で検証する命題を自動的に生成する。

Caduceus の記述例

```
/*@ predicate is_min(int t[],int n,int min) {
  @ (\forall int i; 0 <= i < n => min <= t[i]) &&
  @ (\exists int i; 0 <= i < n && min == t[i])
  @ }
  @*/

/*@ requires n > 0 && \valid_range(t,0,n)
  @ ensures is_min(t,n,\result)
  @*/

int min(int t[],int n) {
  int i;
  int tmp = t[0];
  /*@ invariant 1 <= i <= n && is_min(t,i,tmp)
    @ variant n-i
    @*/
  for (i=1 ; i < n; i++) {
     if (t[i] < tmp) tmp = t[i];
  }
  return tmp;
}
```

Caduceus の生成例

```
Require Export max_spec_why.

(* Why obligation from file "max.c", line 1, characters 16-97: *)
(*Why goal*) Lemma max_impl_po_1 :
  forall (x: Z),
  forall (y: Z),
  forall (HW_1: x > y),
  (* File "max.c", line 1, characters 15-96 *) ((x >= x /\ x >= y) /\
  (forall (z:Z), (z >= x /\ z >= y -> z >= x))).
Proof.
intuition.
(* FILL PROOF HERE *)
Save.

(* Why obligation from file "max.c", line 1, characters 16-97: *)
(*Why goal*) Lemma max_impl_po_2 :
  forall (x: Z),
  forall (y: Z),
  forall (HW_2: x <= y),
  (* File "max.c", line 1, characters 15-96 *) ((y >= x /\ y >= y) /\
  (forall (z:Z), (z >= x /\ z >= y -> z >= y))).
Proof.
intuition.
(* FILL PROOF HERE *)
Save.
```

Outline

問題1：プログラム正当性の証明

次の表明を証明せよ。

$\{x > 0\}$
z := y;
u := x-1;
while u $\neq$ 0 **do**
    z := z+y;
    u := u-1
end
$\{x > 0 \wedge z = x * y\}$

ヒント：次のループ不変性質を利用する。

$$x > 0 \;\wedge\; 0 \leq u \leq x \;\wedge\; x * y = z + u * y$$

問題２：プログラムの導出

次の仕様を満たすプログラムを導出せよ。

1. $\{i < 5\}\ i := e\ \{i < 10\}$
2. $\{s = n^2\}\ s := e; n := n + 1\ \{s = n^2\}$
3. $\{0 < N\}\ S;\ \textbf{while}\ B\ \textbf{do}\ T\ \textbf{end}\ \{s = \Sigma i \mid 0 \leq i < N : a[i] * X^i\}$

期末試験について

- 試験日：2008/7/28 8:30-10:00
- スライド、資料、ノートを持ち込んでも OK

ご協力ありがとうございました。

アンケートのご協力もお願いします。