Hoare　論理 (1)

- プログラム証明のための手法と論理 -

東京大学計数工学科
胡　振江
hu@mist.i.u-tokyo.ac.jp

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（定義）(Partial Correctness Assertion)

{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

S ::=	x := e	代入文
	| S1 ; S2	複合文
	| If B Then S1 Else S2 End	if 文
	| While B do S End	while文

部分正当性の証明

部分的正当性の表明例

```
{N>=0}
i := 1;
f := 1;
While i<=N Do
 f := f*i;
 i := i+1
End
{f=N!}
```

部分的正当性の証明

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

第一階述語論理の拡張

代入文の公理

{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

{P} S1 {R}　{R} S2 {Q}

{P} S1; S2 {Q}

帰結としての部分的正当性 {P} S1; S2 {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} S1 {R} と {R} S2 {Q} が成り立つ必要

if 文の規則

{P and B} S1 {Q}　{P and not B} S2 {Q}

{P} If B Then S1 Else S2 End {Q}

while 文の規則

{P and B} S {P}

{P} While B Do S End {P and not 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

P ➔ P1　{P1} S {Q1}　Q1 ➔ Q

{P} S {Q}

例：二変数の値の交換

{ x=x0 and y=y0 }
t := x;
x := y;
y := t
{ x=y0 and y=x0 }

⇐

{ x=x0 and y=y0 }
t := x;
x := y;
{ x=y0 and t=x0 }
y := t
{ x=y0 and y=x0 }

例：二変数の値の交換

{ x=x0 and y=y0 }
t := x;
x := y;
{ x=y0 and t=x0 }
y := t
{ x=y0 and y=x0 }

⇐

{ x=x0 and y=y0 }
t := x;
{ y=y0 and t=x0 }
x := y;
{ x=y0 and t=x0 }
y := t
{ x=y0 and y=x0 }

証明手法のまとめ

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?
- また、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：階乗

{N>=0}
i := 1;
f := 1;
While i<=N Do
 f := f*i;
 i := i+1
End
{f=N!}

f = N!
←
P and not (i<=N)

{N>=0}
i := 1;
f := 1;
{P}
While i<=N Do
 f := f*i;
 i := i+1
End
{P and not (i<=N)}

例：階乗

{N>=0}
i := 1;
f := 1;
{1<=i<=N+1 and f=(i-1)!}
While i<=N Do
 f := f*i;
 i := i+1
End
{(1<=i<=N+1 and f=(i-1)!) and not (i<=N)}

{N>=0}
i := 1;
f := 1;
{1<=i<=N+1 and f=(i-1)!}
While i<=N Do
{1<=i<=N+1 and f=(i-1)! and (i<=N)}
 f := f*i;
 i := i+1
{1<=i<=N+1 and f=(i-1)!}
End
{(1<=i<=N+1 and f=(i-1)!) and not (i<=N)}

例：階乗

{N>=0}
i := 1;
f := 1;
{1<=i<=N+1 and f=(i-1)!}
While i<=N Do
{1<=i<=N+1 and f=(i-1)! and (i<=N)}
 f := f*i;
 i := i+1
{1<=i<=N+1 and f=(i-1)!}
End
{(1<=i<=N+1 and f=(i-1)!) and not (i<=N)}

{N>=0}
i := 1;
{1<=i<=N+1 and 1=(i-1)!}
f := 1;
{1<=i<=N+1 and f=(i-1)!}
While i<=N Do
{1<=i<=N+1 and f=(i-1)! and (i<=N)}
 f := f*i;
 i := i+1
{1<=i<=N+1 and f=(i-1)!}
End
{(1<=i<=N+1 and f=(i-1)!) and not (i<=N)}

例：階乗

{N>=0}
i := 1;
{1<=i<=N+1 and 1=(i-1)!}
f := 1;
{1<=i<=N+1 and f=(i-1)!}
While i<=N Do
{1<=i<=N+1 and f=(i-1)! and (i<=N)}
 f := f*i;
 i := i+1
{1<=i<=N+1 and f=(i-1)!}
End
{(1<=i<=N+1 and f=(i-1)!) and not (i<=N)}

{N>=0}
i := 1;
{1<=i<=N+1 and i=(i-1)!}
f := 1;
{1<=i<=N+1 and f=(i-1)!}
While i<=N Do
{1<=i<=N+1 and f=(i-1)! and (i<=N)}
 f := f*i;
{1<=i+1<=N+1 and f=i!}
 i := i+1
{1<=i<=N+1 and f=(i-1)!}
End
{(1<=i<=N+1 and f=(i-1)!) and not (i<=N)}

例：階乗

{N>=0}
i := 1;
f := 1;
{1<=i<=N+1 and f=(i-1)!}
While i<=N Do
{1<=i<=N+1 and f=(i-1)! and (i<=N)}
 f := f*i;
 i := i+1
{1<=i<=N+1 and f=(i-1)!}
End
{(1<=i<=N+1 and f=(i-1)!) and not (i<=N)}

{N>=0}
i := 1;
f := 1;
{1<=i<=N+1 and f=(i-1)!}
While i<=N Do
{1<=i<=N+1 and f=(i-1)! and (i<=N)}
 f := f*i;
{0<=i<=N and f=i!}
 i := i+1
{1<=i<=N+1 and f=(i-1)!}
End
{(1<=i<=N+1 and f=(i-1)!) and not (i<=N)}

掛算

```
{ x>0 }
z := y;
u := x - 1;
While u<>0 Do
  z := z + y;
  u := u - 1
End
{ z = x * y }
```

```
{ x>0 }
z := y;
u := x - 1;
While u<>0 Do
  z := z + y;
  u := u - 1
End
{ x>0 and z = x * y }
```

掛算

```
{ x>0 }
z := y;
u := x - 1;
While u<>0 Do
  z := z + y;
  u := u - 1
End
{ x>0 and z = x * y }
```

```
{ x>0 }
z := y;
u := x - 1;
{ x>0 and 0<=u<=x and x*y=z+u*y }
While u<>0 Do
{ x>0 and 0<=u<=x and x*y=z+u*y and u<>0}
  z := z + y;
  u := u – 1
{ x>0 and 0<=u<=x and x*y=z+u*y }
End
{ x>0 and z = x * y }
```

掛算

```
{ x>0 }
z := y;
u := x - 1;
{ x>0 and 0<=u<=x and x*y=z+u*y }
While u<>0 Do
{ x>0 and 0<=u<=x and x*y=z+u*y and u<>0}
  z := z + y;
  u := u – 1
{ x>0 and 0<=u<=x and x*y=z+u*y }
End
{ x>0 and z = x * y }
```

```
{ x>0 }
z := y;
{ x>0 and 0<=x-1<=x and x*y=z+(x-1)*y }
u := x - 1;
{ x>0 and 0<=u<=x and x*y=z+u*y }
While u<>0 Do
{ x>0 and 0<=u<=x and x*y=z+u*y and u<>0}
  z := z + y;
{ x>0 and 0<=u-1<=x and x*y=z+(u-1)*y }
  u := u – 1
{ x>0 and 0<=u<=x and x*y=z+u*y }
End
{ x>0 and z = x * y }
```

掛算

```
{ x>0 }
z := y;
{ x>0 and 0<=x-1<=x and x*y=z+(x-1)*y }
u := x - 1;
{ x>0 and 0<=u<=x and x*y=z+u*y }
While u<>0 Do
{ x>0 and 0<=u<=x and x*y=z+u*y and u<>0}
  z := z + y;
{ x>0 and 0<=u-1<=x and x*y=z+(u-1)*y }
  u := u – 1
{ x>0 and 0<=u<=x and x*y=z+u*y }
End
{ x>0 and z = x * y }
```

演習問題１：これを証明せよ．

演習問題２：最大公約数

```
{ x>0 and y>0 }
t1 := x;
t2 := y;
While t1<>t2 Do
  If t1>t2 Then
      t1 := t1-t2
  Else
      t2 := t2-t1
  End
End;
z := t1
{ z=gcd(x,y) }
```

この表明を証明せよ．

ヒント：ループ不変条件は (計算の途中結果と最大公約数の性質とを考慮)はx>0 and y>0 and t1>0 and t2>0 and gcd(x,y)=gcd(t1,t2)である．