ダイクストラの検証法

ダイクストラ　Edger Wybe Dijkstra (1930.5.11-2011.8.6)

構造化プログラミング，goto有害論，分散プログラミング，

形式的検証，プログラム導出　etc.

ダイクストラの方法

最弱前条件twp(P, B)を用いて検証を行う.

非決定的プログラム言語を用いる.

Bを手がかりにPを発見する方法につながる手法

最弱前条件と表明付きプログラム

twp(P, B) = {ρ | ∃ρ'Exec$_I$(P, ρ, ρ')∧ρ∈pwp(P, B) }

これは，状態の集合として定義されている.

(ただし pwp$_I$(P, B) = {ρ | Exec$_I$(P, ρ, ρ')なるどんなρ'に対しても，I, ρ'I=B })

これは，「前条件として最も弱い命題」を表す，すなわち

I I=<A>P<B> ⇔ ∀ρ(I, ρ I= A ⇒ ρ∈twp(P, B))

である.

(φがψより**弱い(weaker)**とはψからφが導出できる，つまりψ⊃φが成り立つことである．)

twp(P, B)を表明として使うことができるのであれば

(すなわち，言語が充分な記述能力を持つとすれば)

I I=<A>P<B> ⇔ I I= (A ⊃ twp(P, B))

つまり，<A>P<B>のかわりに A ⊃ twp(P, B)を用いることができる.

最弱前条件の一般法則

$$\frac{B1 \supset B2}{twp(P, B1) \supset twp(P, B2)}$$

すなわち

$$\models (B1 \supset B2) \quad \Rightarrow \quad \models (twp(P, B1) \supset twp(P, B2))$$

$$twp(P, A \wedge B) = twp(P, A) \wedge twp(P, B)$$

$$twp(P, A \vee B) = twp(P, A) \vee twp(P, B)$$

$$twp(P, False) = False$$

最弱前条件の特徴付けと検証

ホーア論理では公理と推論規則を使うところを,

ダイクストラの方法ではtwp(P, B)を特徴つける等式を用いる.

代入文の特徴付けは

$$twp(a:=t, B) = B[t/a]$$

である.

ここから

$$B[t/a] \supset twp(a:=t, B)$$

が成り立つ.

これは代入文の公理

$$< B[t/a] > a:=t < B >$$

に対応する.

同様に,

twp(skip, B) = B

twp(if C then P else Q fi, B) = $(C \supset twp(P, B)) \wedge (\neg C \supset twp(Q, B))$

twp(begin P_1; ... ;P_n, B) = $twp(P_1, \ldots twp(P_{n-1}, twp(P_n, B))\ldots)$

twp(while C do P od, B) = $\exists n H_n(C, P, B)$

ただし, H_nは

$H_0(C, P, B) = \neg C \wedge B$

$H_{n+1}(C, P, B) = C \wedge twp(P, H_n(C, P, B))$

で定義される.

条件文の規則をダイクストラ流に表現

$$\frac{(C \wedge A) \supset twp(P, B) \qquad (\neg C \wedge A) \supset twp(Q, B)}{A \supset twp(\text{if } C \text{ then } P \text{ else } Q \text{ fi}, B)}$$

これを証明する.

この推論の前提は

$$((C \wedge A) \supset twp(P, B)) \wedge ((\neg C \wedge A) \supset twp(Q, B))$$

同値変形すると

$$A \supset ((C \supset twp(P, B)) \wedge (\neg C \supset twp(Q, B)))$$

条件文の特徴付けの等式

$$twp(\text{if } C \text{ then } P \text{ else } Q \text{ fi}, B) = (C \supset twp(P, B)) \wedge (\neg C \supset twp(Q, B))$$

を用いると

$$A \supset twp(\text{if } C \text{ then } P \text{ else } Q \text{ fi}, B)$$

これは条件文の推論の結論である.

同様に，while文の規則をダイクストラ流に表現

$$\frac{(C \wedge A) \supset twp(P, A)}{A \wedge twp(\text{while } C \text{ do } P \text{ od}, True) \supset twp(\text{while } C \text{ do } P \text{ od}, \neg C \wedge A)}$$

これをwhile文の特徴付け

$twp(\text{while } C \text{ do } P \text{ od}, B) = \exists n H_n(C, P, B)$

ただし，H_nは

$H_0(C, P, B) = \neg C \wedge B$

$H_{n+1}(C, P, B) = C \wedge twp(P, H_n(C, P, B))$

を用いて証明する.

twp(P, True)は，「現在の状態でPを実行すると停止する」を意味する.

すなわち，この推論規則の結論はプログラムが停止することを仮定している.

したがって，この結論は{A} while C do P od{¬C∧A}と同値.

一方，A⊃twp(while C do P od, True)を考えれば，Aのもとで停止するための条件を含んだ完全正当性の推論規則であるとみなすことができる.

非決定的プログラムとその検証

ガード付きコマンド(guarded command)

条件文の代わりに以下の文を用いる

IF ≡ if $C_1 \rightarrow P_1$ □ $C_2 \rightarrow P_2$ □ … □ $C_n \rightarrow P_n$ fi

C_i : **ガード**

$C_i \rightarrow P_i$: **ガード付きコマンド**

条件C_iが成り立つときにP_iを実行する

ガード付きコマンドの実行は非決定的

C_iが成り立つのならどれを実行してもよい.

成り立つガードがないときは停止しない.

$Exec(IF, \rho, \rho') \equiv C_1 \wedge Exec(P_1, \rho, \rho') \vee C_2 \wedge Exec(P_2, \rho, \rho') \vee ... \vee C_n \wedge Exec(P_n, \rho, \rho')$

whileの代わりに以下のDOを用いる.

$DO \equiv \text{do } C_1 \rightarrow P_1 \quad \square \quad C_2 \rightarrow P_2 \quad \square \quad \ldots \quad \square \quad C_n \rightarrow P_n \text{ od}$

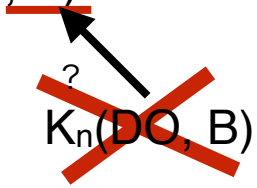
ガードC_iが成立すればどのガード付きコマンド $C_i \rightarrow P_i$ を実行してもよい.
成り立つガードがある限り繰り返す.
成り立つガードがなくなれば正常終了する.

$twp(IF, B) = (C_1 \vee \ldots \vee C_n) \wedge (C_1 \supset twp(P_1, B)) \wedge \ldots \wedge (C_n \supset twp(P_n, B))$

$twp(DO, B) = \exists n K_n(DO, B)$

$K_0(DO, B) = B \wedge \neg(C_1 \vee \ldots \vee C_n)$

$K_{n+1}(DO, B) = twp(IF, K_n(DO, B)) \vee K_0(DO, B)$



ここで,

$IF \equiv$ if $C_1 \rightarrow P_1 \quad \square \quad C_2 \rightarrow P_2 \quad \square \quad \ldots \quad \square \quad C_n \rightarrow P_n$ fi

$DO \equiv$ do $C_1 \rightarrow P_1 \quad \square \quad C_2 \rightarrow P_2 \quad \square \quad \ldots \quad \square \quad C_n \rightarrow P_n$ od

のことである.

while文の規則に類似したDO文の規則

$$\frac{((C1 \vee \ldots \vee Cn) \wedge B) \supset twp(IF, B) \qquad ((C1 \vee \ldots \vee Cn) \wedge B \wedge bound=b) \supset twp(IF, bound<b)}{B \supset twp(DO, B \wedge \neg(C1 \vee \ldots \vee Cn))}$$

twpの計算

twpの特徴付けを用いて，twp(P, B)が計算できる.

例えば，代入文の特徴付けはtwp(a:=t, B) =B[t/a]であったので,

具体的にBを与えることにより,

$$\mathrm{twp}(a:=a*3,\ a=3) = (a=3)[a*3/a] = a*3=3$$

となる.

最弱前条件の特徴付け(再掲)

$twp(a:=t, B) = B[t/a]$

$twp(skip, B) = B$

$twp(\text{if } C \text{ then } P \text{ else } Q \text{ fi}, B) = (C \supset twp(P, B)) \wedge (\neg C \supset twp(Q, B))$

$twp(\text{begin } P_1; \ldots ; P_n, B) = twp(P_1, \ldots twp(P_{n-1}, twp(P_n, B))\ldots)$

$twp(\text{while } C \text{ do } P \text{ od}, B) = \exists n H_n(C, P, B)$

ただし,

$H_0(C, P, B) = \neg C \wedge B$

$H_{n+1}(C, P, B) = C \wedge twp(P, H_n(C, P, B))$

で定義される.

$twp(IF, B) = (C_1 \vee \ldots \vee C_n) \wedge (C_1 \supset twp(P_1, B)) \wedge \ldots \wedge (C_n \supset twp(P_n, B))$

ただし, $IF \equiv \text{if } C_1 \rightarrow P_1 \ \square \ C_2 \rightarrow P_2 \ \square \ \ldots \ \square \ C_n \rightarrow P_n \text{ fi}$

$twp(DO, B) = \exists n K_n(DO, B)$

ただし,

$DO \equiv \text{do } C_1 \rightarrow P_1 \ \square \ C_2 \rightarrow P_2 \ \square \ \ldots \ \square \ C_n \rightarrow P_n \text{ od}$

$K_0(DO, B) = B \wedge \neg(C_1 \vee \ldots \vee C_n)$

$K_{n+1}(DO, B) = twp(IF, K_n(DO, B)) \vee K_0(DO, B)$

例題

DO: do　　x>2 → begin x:=x-1; i:=i+1 end
　　□　　x>3 → begin x:=x-2; i:=i+1 end od
B: i≤2
を考える．ただしx, iは自然数とする.

twp(DO, B)を計算する.

$twp(DO, B) = \exists n K_n(DO, B)$

$K_0(DO, B) = i\leq 2 \wedge \neg(x>2 \vee x>3) = x\leq 2 \wedge i\leq 2$

$K_1(DO, B) = twp(IF, K_0(DO, B)) \vee K_0(DO, B) = twp(IF, i\leq 2 \wedge x\leq 2) \vee x\leq 2 \wedge i\leq 2$

$twp(IF, i\leq 2 \wedge x\leq 2) = (x>2 \vee x>3) \wedge (x>2 \supset twp(\text{begin } x:=x-1; i:=i+1 \text{ end}, x\leq 2 \wedge i\leq 2)) \wedge (x>3 \supset twp(\text{begin } x:=x-2; i:=i+1 \text{ end}, x\leq 2 \wedge i\leq 2))$

$twp(\text{begin } x:=x-1; i:=i+1 \text{ end}, x\leq 2 \wedge i\leq 2) = twp(x:=x-1, twp(i:=i+1, x\leq 2 \wedge i\leq 2)) = twp(x:=x-1, i+1\leq 2 \wedge x\leq 2) = i\leq 1 \wedge x\leq 3$

$twp(\text{begin } x:=x-2; i:=i+1 \text{ end}, x\leq 2 \wedge i\leq 2) = i\leq 1 \wedge x\leq 4$

$\therefore twp(IF, i\leq 2 \wedge x\leq 2) = (x>2 \vee x>3) \wedge (x>2 \supset i\leq 1 \wedge x\leq 3) \wedge (x>3 \supset i\leq 1 \wedge x\leq 4) = (x=3 \vee x=4) \wedge i\leq 1$

$\therefore K_1(DO, B) = (x=3 \vee x=4) \wedge i\leq 1 \vee x\leq 2 \wedge i\leq 2$

同様に

$K_2(DO, B) = twp(IF, K_1(DO, B)) \vee K_0(DO, B)$

ただし $K_1(DO, B) = (x=3 \vee x=4) \wedge i \leq 1 \vee x \leq 2 \wedge i \leq 2$

$twp(IF, K_1(DO, B))$

$= (x>2 \vee x>3) \wedge (x>2 \supset twp(\text{begin } x:=x-1;\ i:=i+1 \text{ end}, K_1(DO, B))) \wedge (x>3 \supset twp(\text{begin } x:=x-2;\ i:=i+1 \text{ end}, K_1(DO, B)))$

$= (x>2 \vee x>3) \wedge (x>2 \supset (x=4 \vee x=5) \wedge i \leq 0 \vee x \leq 3 \wedge i \leq 1) \wedge (x>3 \supset (x=5 \vee x=6) \wedge i \leq 0 \vee x \leq 4 \wedge i \leq 1)$

$= x=3 \wedge i \leq 1 \vee x=4 \wedge i \leq 1 \vee x=5 \wedge i \leq 0 \vee x=6 \wedge i \leq 0$

$\therefore K_2(DO, B) = x \leq 2 \wedge i \leq 2 \vee x=3 \wedge i \leq 1 \vee x=4 \wedge i \leq 1 \vee x=5 \wedge i=0 \vee x \leq 6 \wedge i=0$

$K_3(DO, B) = twp(IF, K_2(DO, B)) \vee K_0(DO, B)$

$twp(IF, K_2(DO, B))$

$= (x>2 \vee x>3) \wedge (x>2 \supset twp(\text{begin } x:=x-1;\ i:=i+1 \text{ end}, K_2(DO, B))) \wedge (x>3 \supset twp(\text{begin } x:=x-2;\ i:=i+1 \text{ end}, K_2(DO, B)))$

$= (x>2 \vee x>3) \wedge (x>2 \supset x \leq 3 \wedge i \leq 1 \vee x=4 \wedge i \leq 0 \vee x=5 \wedge x \leq 0 \vee x=6 \wedge i=-1 \vee x \leq 7 \wedge i=-1) \wedge (x>3 \supset x \leq 4 \wedge i \leq 1 \vee x=5 \wedge i \leq 0 \vee x=6 \wedge x \leq 0 \vee x=7 \wedge i=-1 \vee x \leq 6 \wedge i=-1)$

$\therefore K_3(DO, B) = x \leq 2 \wedge i \leq 2 \vee x=3 \wedge i \leq 1 \vee x=4 \vee i \leq 1 \vee x=5 \wedge i=0 \vee x \leq 6 \wedge i=0$

$n \geq 2$のとき$K_n(DO, B) = K_2(DO, B)$となるので,

$twp(DO, B) = K_2(DO, B) = x \leq 2 \wedge i \leq 2 \vee x=3 \wedge i \leq 1 \vee x=4 \wedge i \leq 1 \vee x=5 \wedge i=0 \vee x \leq 6 \wedge i=0$

演習

DO: do $z \neq x \rightarrow z := z+1;\ y := y*z$ od
B: $y = x!$
に対してtwp(DO, B)を計算せよ.

解

$twp(DO, B) = \exists n\ K_n(DO, B)$

$K_0(DO, B) = \ y=x!\wedge z=x$

$K_1(DO, B) = \ twp(IF, K_0(DO, B))\vee K_0(DO, B)$

$twp(IF, K_0(DO, B)) \ = \ z\neq x\wedge(z\neq x\supset twp(z:=z+1;\ y:=y^*z,\ y=x!\wedge z=x))$

$= \ y(z+1)=x!\wedge z+1=x$

$\therefore K_1(DO, B) = y(z+1)=x!\wedge z+1=x \vee y=x!\wedge z=x$

$K_2(DO, B) = \ twp(IF, K_1(DO, B))\vee K_0(DO, B)$

$twp(IF, K_1(DO, B)) \ = \ z\neq x\wedge(z\neq x\supset twp(z:=z+1;\ y:=y^*z,\ y(z+1)=x!\wedge z+1=x \vee y=x!\wedge z=x)$

$= \ y(z+1)(z+2)=x!\wedge z+2=x \vee y(z+1)=x!\wedge z+1=x$

$\therefore K_2(DO, B) = y(z+1)(z+2)=x!\wedge z+2=x \vee y(z+1)=x!\wedge z+1=x \vee y=x!\wedge z=x$

$\therefore K_n(DO, B) = y(z+1)(z+2)...(z+n)=x!\wedge z+n=x \vee... \vee y(z+1)=x!\wedge z+1=x \vee y=x!\wedge z=x$

$\therefore twp(DO, B) =\exists n(y(z+1)(z+2)...(z+n)=x!\wedge z+n=x)$

非決定プログラムの開発検証

例

x: 自然数

仕様:

x≤nの範囲で関数f(x)が最大値となるようなxを見つける

すなわち

$twp(X0, x \leq n \wedge \forall j(j \leq n \supset f(j) \leq f(x))$

なるプログラムX0を開発し，同時に検証する.

解法

f(i)の値をi=1, ... , n-1の順にみていき,

x=0, 1, ... , iの範囲で見つかった最大のf(x)のxを変数xにおくことにする.

以下, B0を　$x \leq n \wedge \forall j(j \leq n \supset f(j) \leq f(x))$　とする.

まずi=x=0とする.

$X0 \equiv$ begin x:=0; i:=0; X1 end

これを仕様に代入すると

twp(begin x:=0; i:=0; X1 end, B0)

これは

$x=0 \wedge i=0 \supset twp(X1, B0)$

と同じである.

X1はiを1づつ増やしてn-1までf(i)の最大値を探す部分であるので,

$$X1 \equiv \text{do } i<n \rightarrow X2 \text{ od}$$

$$X2 \equiv \text{begin } X3;\ i:=i+1 \text{ end}$$

となる.

ループX1の不変表明は,

「x=0, ... , iまでで見つかった最大のf(x)のxを変数xにおく」

すなわち

$$x \leq i \leq n \wedge \forall j(j \leq i \supset f(j) \leq f(x))$$

とならなければならない.

これをB1とおく.

ちなみに初期条件 $x=0 \wedge i=0$ はB1を満たす.

$$\frac{((C1 \vee \ldots \vee Cn) \wedge B) \supset twp(IF, B)}{((C1 \vee \ldots \vee Cn) \wedge B \wedge bound=b) \supset twp(IF, bound<b)}$$
$$B \supset twp(DO, B \wedge \neg(C1 \vee \ldots \vee Cn))$$

を用いて

$$B1 \supset twp(X1, B1 \wedge \neg(i<n))$$

が証明できれば

$$x=0 \wedge i=0 \quad \supset \quad twp(X1, B0)$$

が導ける.

検証の前提は

$$(i<n \wedge B1) \supset twp(X2, B1)$$
$$(i<n \wedge B1 \wedge bound=b) \supset (X2, bound<b)$$

である. 後者は

$$X2 \quad \equiv \quad \text{begin } X3; \ i:=i+1 \ \text{end}$$

でiを1づつ増やしていることになるので自明.

前者は,

$$(i<n \wedge B1) \supset twp(X3, B1[i+1/i])$$

と同値であるので, f(x)が0, ... , iの範囲で最大であるとき, X3を実行すると0, ... , i+1の範囲で最大になればよい.

つまり,

$$X3 \quad \equiv \quad \text{if } f(i+1) \leq f(x) \rightarrow \text{skip} \ \square \ f(i+1) > f(x) \rightarrow x := x+1 \ \text{fi}$$

とすればIFの推論規則より

$$(i<n \wedge B1) \supset twp(X3, B1[i+1/i])$$

を満たす.

(ただし, $B1 \equiv x \leq i \leq n \wedge \forall j(j \leq i \supset f(j) \leq f(x))$)

したがって, 求めるプログラムは

```
begin x:=0, i:=0;
    do i<n→
        begin
            if f(i+1)≤f(x)→skip □ f(i+1)>f(x)→x:=x+1 fi;
            i:=i+1;
        end
    od
end
```

となる.