§2　程序与系统模型

以下是最大公约数算法的流程。

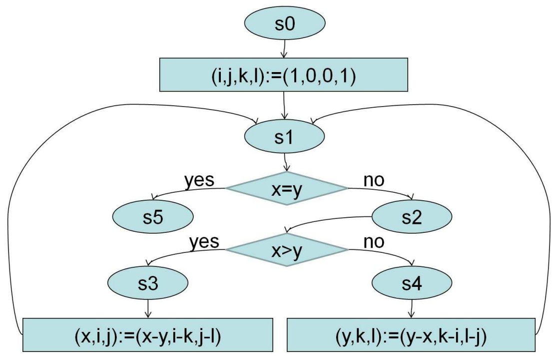

图 1　最大公约数算法的流程

我们有常数 $0, 1$ ，函数 $-$ ，谓词 $=, >$ ，变量 x, y, i, j, k, l 。

§2.1　流程图程序

设 $B = (F, P)$ 和 V 如下：

$$\boxed{\begin{array}{lcl} F & = & \{0, 1, -\} \\ P & = & \{=, >\} \\ V & = & \{x, y, i, j, k, l\} \end{array}}$$

$I = (Int, I_0)$

$$\boxed{\begin{array}{lcl} I_0(0) & = & 0 \\ I_0(1) & = & 1 \\ I_0(-) & = & - \\ I_0(=) & = & = \\ I_0(>) & = & > \end{array}}$$

最大公约数算法可表示为 (B, V) 上的流程图程序如下：

beg:	$(i, j, k, l) := (1, 0, 0, 1)$ goto l_1
l_1:	if $\neg(x = y)$ goto l_2 else goto end
l_2:	if $(x > y)$ goto l_3 else goto l_4
l_3:	$(x, i, j) := (x - y, i - k, j - l)$ goto l_1
l_4:	$(y, k, l) := (y - x, k - i, l - j)$ goto l_1

状态

相关的标号：

$$\begin{array}{l} beg \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ end \end{array}$$

变量状态

$\{(x, y, i, j, k, l) \mid x, y, i, j, k, l \in INT\}$

运行

$$\begin{array}{l} [beg, (4, 6, 1, 2, 3, 4)] \\ [l_1, (4, 6, 1, 0, 0, 1)] \\ [l_2, (4, 6, 1, 0, 0, 1)] \\ [l_4, (4, 6, 1, 0, 0, 1)] \\ [l_1, (4, 2, 1, 0, -1, 1)] \\ [l_2, (4, 2, 1, 0, -1, 1)] \\ [l_3, (4, 2, 1, 0, -1, 1)] \\ [l_1, (2, 2, 2, -1, -1, 1)] \\ [end, (2, 2, 2, -1, -1, 1)] \end{array}$$

程序性质

设 $\varphi(\sigma) \equiv \sigma(x) > 0 \wedge \sigma(y) > 0$ 为程序 T 的初始状态满足的条件， $\psi(\sigma) \equiv \sigma(x) = \sigma(y)$ 为对程序 T 的终止状态的要求。

T 对于 φ 和 ψ 的部分正确性表示为

$$\models \{\varphi\}T\{\psi\}$$

即

$$\varphi(\sigma) \to ((beg, \sigma) \overset{*}{\Rightarrow} (end, \sigma') \to \psi(\sigma'))$$

T 对于 φ 和 ψ 的完全正确性表示为

$$\models [\varphi]T[\psi]$$

即

$$\varphi(\sigma) \to ((beg, \sigma) \overset{*}{\Rightarrow} (end, \sigma') \wedge \psi(\sigma'))$$

§2.2 结构化循环语句程序

设 $B = (F, P)$ 和 V 如下：

$$\boxed{\begin{array}{lcl} F & = & \{0, 1, -\} \\ P & = & \{=, >\} \\ V & = & \{x, y, i, j, k, l\} \end{array}}$$

$I = (Int, I_0)$

$$\boxed{\begin{array}{lcl} I_0(0) & = & 0 \\ I_0(1) & = & 1 \\ I_0(-) & = & - \\ I_0(=) & = & = \\ I_0(>) & = & > \end{array}}$$

最大公约数算法可表示为 (B, V) 上的结构化循环语句程序如下：

```
i:=1; j:=0; k:=0; l:=1;
while ¬(x = y) do
    if x > y then x:=x-y; i:=i-k; j:=j-1;
              else y:=y-x; k:=k-i; l:=l-j;
    fi
od
```

状态

相关的程序段标号：

```
S0 i:=1; S1 j:=0; S2 k:=0; S3 l:=1;
S4 while ¬(x = y) do
    S5 if x > y then S6 x:=x-y; S7 i:=i-k; S8 j:=j-1;
              else S9 y:=y-x; S10 k:=k-i; S11 l:=l-j;
    fi
od S12
```

变量状态

$\{(x, y, i, j, k, l) \mid x, y, i, j, k, l \in INT\}$

运行

$$\begin{array}{ll}
[S_0, (4, 6, 1, 2, 3, 4)] & i := 1 \\
[S_1, (4, 6, 1, 2, 3, 4)] & j := 0 \\
[S_2, (4, 6, 1, 0, 3, 4)] & k := 0 \\
[S_3, (4, 6, 1, 0, 0, 4)] & l := 1 \\
[S_4, (4, 6, 1, 0, 0, 1)] & \text{while } \neg(x = y) \\
[S_5, (4, 6, 1, 0, 0, 1)] & \text{if } x > y \\
[S_6, (4, 6, 1, 0, 0, 1)] & y := y - x \\
[S_7, (4, 2, 1, 0, 0, 1)] & k := k - i \\
[S_8, (4, 2, 1, 0, -1, 1)] & l = l - j \\
[S_4, (4, 2, 1, 0, -1, 1)] & \text{while } \neg(x = y) \\
[S_5, (4, 2, 1, 0, -1, 1)] & \text{if } x > y \\
[S_9, (4, 2, 1, 0, -1, 1)] & x := x - y \\
[S_{10}, (2, 2, 1, 0, -1, 1)] & i := i - k \\
[S_{11}, (2, 2, 2, 0, -1, 1)] & j := j - l \\
[S_{11}, (2, 2, 2, -1, -1, 1)] & \text{while } \neg(x = y) \\
[S_{12}, (2, 2, 2, -1, -1, 1)] & \epsilon
\end{array}$$

程序性质

设 $\varphi(\sigma) \equiv \sigma(x) > 0 \wedge \sigma(y) > 0$ 为程序 T 的前断言，$\psi(\sigma) \equiv \sigma(x) = \sigma(y)$ 为对程序 T 的终后断言。

T 对于 φ 和 ψ 的部分正确性表示为

$$\models \{\varphi\}T\{\psi\}$$

即

$$\varphi(\sigma) \to ((T,\sigma)\overset{*}{\Rightarrow}(\epsilon,\sigma') \to \psi(\sigma'))$$

T 对于 φ 和 ψ 的完全正确性表示为

$$\models [\varphi]T[\psi]$$

即

$$\varphi(\sigma) \to ((T,\sigma)\overset{*}{\Rightarrow}(\epsilon,\sigma') \wedge \psi(\sigma'))$$

.

设 $\varphi(\sigma) \equiv \sigma(x) \geq 0 \wedge \sigma(y) \geq 0$ 为程序 T 的前断言，$\psi(\sigma) \equiv \sigma(x) = \sigma(y)$ 为对程序 T 的终后断言。

则 T 不满足对于 φ 和 ψ 的完全正确性。

.

其它前断言和后断言：

（1）前断言：

$\varphi(\sigma) \equiv \sigma(x) \geq 0 \wedge \sigma(y) \geq 0 \wedge \sigma(x) = \sigma(a) \wedge \sigma(y) = \sigma(b)$

后断言：

$\psi(\sigma) \equiv \sigma(x) = gcd(\sigma(a), \sigma(b))$

.

（2）前断言：

$\varphi(\sigma) \equiv \sigma(x) \geq 0 \wedge \sigma(y) \geq 0 \wedge \sigma(x) = \sigma(a) \wedge \sigma(y) = \sigma(b)$

后断言：

$\psi(\sigma) \equiv \sigma(x) = gcd(\sigma(a), \sigma(b)) \wedge \sigma(x) = \sigma(i) * \sigma(a) + \sigma(j) * \sigma(b)$

§2.3 一阶迁移系统 (FTS)

设 $B = (F, P)$ 和 V 如下：

$$\boxed{\begin{array}{lcl} F & = & \{0, 1, s_0, s_1, s_2, s_3, s_4, s_5, -\} \\ P & = & \{=, >, <\} \\ V & = & \{a, x, y, i, j, k, l\} \end{array}}$$

$I = (Int, I_0)$

$I_0(s_0) = I_0(0)$	$=$	0
$I_0(s_1) = I_0(1)$	$=$	1
$I_0(s_2)$	$=$	2
$I_0(s_3)$	$=$	3
$I_0(s_4)$	$=$	4
$I_0(s_5)$	$=$	5
$I_0(-)$	$=$	$-$
$I_0(=)$	$=$	$=$
$I_0(>)$	$=$	$>$
$I_0(<)$	$=$	$<$

最大公约数算法表示为 (B, V) 上的迁移系统 (T, Θ) 如下：

T	$(a = s_0)$	$\longrightarrow$	$(i, j, k, l, a) := (1, 0, 0, 1, s_1)$
	$(a = s_1 \wedge x = y)$	$\longrightarrow$	$(a) := (s_5)$
	$(a = s_1 \wedge \neg(x = y))$	$\longrightarrow$	$(a) := (s_2)$
	$(a = s_2 \wedge x > y)$	$\longrightarrow$	$(a) := (s_3)$
	$(a = s_2 \wedge \neg(x > y))$	$\longrightarrow$	$(a) := (s_4)$
	$(a = s_3)$	$\longrightarrow$	$(x, i, j, a) := (x - y, i - k, j - l, s_1)$
	$(a = s_4)$	$\longrightarrow$	$(y, k, l, a) := (y - x, k - i, l - j, s_1)$
Θ	$(x > 0 \wedge y > 0 \wedge a = s_0)$		

状态空间

$\Sigma = \{(a, x, y, i, j, k, l)\} \mid a \in \{0, 1, ..., 5\}, x, y, i, j, k, l \in INT\}$ 。

初始状态

系统的初始状态集合为 $\{(0, x, y, i, j, k, l) | x, y > 0, i, j, k, l \in INT\}$ 。

运行

$$
\begin{array}{l}
(s_0, 4, 6, 1, 2, 3, 4) \\
(s_1, 4, 6, 1, 0, 0, 1) \\
(s_2, 4, 6, 1, 0, 0, 1) \\
(s_4, 4, 6, 1, 0, 0, 1) \\
(s_1, 4, 2, 1, 0, -1, 1) \\
(s_2, 4, 2, 1, 0, -1, 1) \\
(s_3, 4, 2, 1, 0, -1, 1) \\
(s_1, 2, 2, 2, -1, -1, 1) \\
(s_5, 2, 2, 2, -1, -1, 1)
\end{array}
$$

可达状态

以上所列状态的集合为 $\{\sigma' \mid \sigma \xrightarrow{*} \sigma', \sigma \models_I x = 4 \wedge y = 6 \wedge i = 1 \wedge j = 2 \wedge k = 3 \wedge l = 4 \wedge a = s_0\}$ 。

系统的可达状态集合 $rh(\Theta)$ 为 $\{\sigma' \mid \sigma \xrightarrow{*} \sigma', \sigma \models_I x > 0 \wedge y > 0 \wedge a = s_0\}$ 。

安全性质

例子： $\varphi \equiv a = s_5 \to x = y$

统满足安全性质 φ ，即 $\forall \sigma \in rh(\Theta).\sigma \models_I \varphi$ 。

表示方式 2

最大公约数算法表示为 (B, V) 上的迁移系统 (T, Θ) 如下：

T	$(a = s_0) \longrightarrow (i, j, k, l, a) := (1, 0, 0, 1, s_1)$ $(a = s_1 \wedge x = y) \longrightarrow (a) := (s_2)$ $(a = s_1 \wedge x > y) \longrightarrow (x, i, j) := (x - y, i - k, j - l)$ $(a = s_1 \wedge x < y) \longrightarrow (y, k, l) := (y - x, k - i, l - j)$
Θ	$(x > 0 \wedge y > 0 \wedge a = s_0)$

状态空间

$\Sigma = \{(a, x, y, i, j, k, l)\} \mid a \in \{0, 1, ..., 5\}, x, y, i, j, k, l \in INT\}$ 。

初始状态

系统的初始状态集合为 $\{(0, x, y, i, j, k, l) | x, y > 0, i, j, k, l \in INT\}$ 。

表示方式 3

最大公约数算法表示为 (B, V) 上的迁移系统 (T, Θ) 如下：

T	$(x = y) \longrightarrow (a) := (s_2)$ $(x > y) \longrightarrow (x, i, j) := (x - y, i - k, j - l)$ $(x < y) \longrightarrow (y, k, l) := (y - x, k - i, l - j)$
Θ	$(x > 0 \wedge y > 0 \wedge i = 1 \wedge j = 0 \wedge k = 0 \wedge l = 1 \wedge a = s_0)$

状态空间

$\Sigma = \{(a, x, y, i, j, k, l)\} \mid a \in \{0, 1, ..., 5\}, x, y, i, j, k, l \in INT\}$ 。

初始状态

系统的初始状态集合为 $\{(0, x, y, 1, 0, 0, 1) \mid x, y > 0\}$ 。