(1) $p_1 \sqcap p_1 = p_1$;

(2) $p_1 \sqcap p_2 = p_2 \sqcap p_1$;

(3) $(p_1 \sqcap p_2) \sqcap p_3 = p_1 \sqcap (p_2 \sqcap p_3)$.

推论 5. 对任何 $S_1,S_2 \in 2^{ProperP}$:

(1) $Reduce(S_1) \in L_G$;

(2) $S_1 \sqsubseteq Reduce(S_1) \wedge Reduce(S_1) \sqsubseteq S_1$;

(3) $S_1 \sqsubseteq S_2 \Rightarrow Reduce(S_1) \sqsubseteq Reduce(S_2)$.

引理 1. 对任何 $L_1,L_2 \in L_G, L_1 \sqcap_G L_2 \in L_G$.

证明:结论直接来自 $\sqcap_G$ 的定义. □

引理 2. 对任何 $S_1,S_2,S_3 \in 2^{ProperP}$:

(1) $S_1 \sqcap_G S_2 \sqsubseteq S_1 \wedge S_1 \sqcap_G S_2 \sqsubseteq S_2$;

(2) $S_1 \sqsubseteq S_2 \wedge S_1 \sqsubseteq S_3 \Rightarrow S_1 \sqsubseteq S_2 \sqcap_G S_3$.

证明:根据推论 3,对于任何性质 $p_1 \in S_1, p_2 \in S_2, p_1 \sqcap p_2 \in S_2 \sqcap_G S_2$,并且 $p_1 \sqcap p_2 \preceq p_1 \wedge p_1 \sqcap p_2 \preceq p_2$.因此,$S_1 \sqcap_G S_2 \sqsubseteq S_1 \wedge S_1 \sqcap_G S_2 \sqsubseteq S_2$ 成立.$S_1 \sqsubseteq S_2 \wedge S_1 \sqsubseteq S_3$ 推出 $\forall p_1 \in S_1, \exists p_2 \in S_2, \exists p_3 \in S_3 \Rightarrow p_1 \preceq p_2 \wedge p_1 \preceq p_3$.根据推论 3,$S_1 \sqsubseteq S_2 \sqcap_G S_3$ 成立. □

引理 3. 对任何 $L_1,L_2 \in L_G, L_1 \sqsubseteq L_2 \wedge L_2 \sqsubseteq L_1 \Leftrightarrow L_1 = L_2$.

证明:$L_1 \sqsubseteq L_2 \wedge L_2 \sqsubseteq L_1$ 推出 $\forall p_1 \in L_1, \exists p_2 \in L_2, \exists p_3 \in L_1 \Rightarrow p_1 \preceq p_2 \wedge p_2 \preceq p_3$.$p_1 \preceq p_2 \wedge p_2 \preceq p_3 \wedge L_1 \in L_G \Rightarrow p_1 = p_3 \Rightarrow p_1 \preceq p_2 \wedge p_2 \preceq p_1 \Rightarrow p_1 = p_2$.因此 $\forall p_1 \in L_1, \exists p_2 \in L_2 \Rightarrow p_1 = p_2 \Rightarrow p_1 \in L_2$.同理可证,$\forall p_2 \in L_2, \exists p_1 \in L_1 \Rightarrow p_1 = p_2 \Rightarrow p_2 \in L_1$.因此,$L_1 = L_2$. □

引理 4. 对任何 $L_1,L_2 \in L_G, L_1 \sqsubseteq L_2 \Leftrightarrow L_1 = L_1 \sqcap_G L_2$.

证明:要证明 $L_1 \sqsubseteq L_2 \Leftrightarrow L_1 = L_1 \sqcap_G L_2$,只需证明:

1) $L_1 \sqsubseteq L_2 \Rightarrow L_1 = L_1 \sqcap_G L_2$;

2) $L_1 = L_1 \sqcap_G L_2 \Rightarrow L_1 \sqsubseteq L_2$.

根据引理 2,结论 1)可证.令 $S(L_1,L_2) = \{p_1 \sqcap p_2 | p_1 \in L_1 \wedge p_2 \in L_2 \wedge p_1 \sqcap p_2 \neq \bot\}$.根据推论 5,$L_1 \sqsubseteq L_2 \Rightarrow L_1 \sqsubseteq S(L_1,L_2) \Rightarrow L_1 \sqsubseteq Reduce(S(L_1,L_2)) \Rightarrow L_1 \sqsubseteq L_1 \sqcap_G L_2$.根据引理 2,$L_1 \sqcap_G L_2 \sqsubseteq L_1$.根据引理 3,$L_1 \sqsubseteq L_1 \sqcap_G L_2 \wedge L_1 \sqcap_G L_2 \sqsubseteq L_1 \wedge L_1 \in L_G \wedge L_1 \sqcap_G L_2 \in L_G$ 推出 $L_1 = L_1 \sqcap_G L_2$.结论 2)成立. □

定理 1. $(L_G, \sqcap_G)$是关于 CFG G 的一个交半格(meet semi-lattice),并且格的高度有穷.

证明:$(L_G, \sqcap_G)$是交半格当且仅当,对所有 $L_1,L_2,L_3 \in L_G$:

1) $L_1 \sqcap_G L_1 = L_1$;

2) $L_1 \sqcap_G L_2 = L_2 \sqcap_G L_1$;

3) $(L_1 \sqcap_G L_2) \sqcap_G L_3 = L_1 \sqcap_G (L_2 \sqcap_G L_3)$;

4) $L_1 \sqsubseteq L_2 \Leftrightarrow L_1 = L_1 \sqcap_G L_2$.

结论 1)、结论 2)和结论 4)通过引理 4 和 $\sqcap_G$ 定义可证.令 $S(L_1,L_2) = \{p_1 \sqcap p_2 | p_1 \in L_1 \wedge p_2 \in L_2 \wedge p_1 \sqcap p_2 \neq \bot\}$.要证明结论 3),只需要证明:

a) $(L_1 \sqcap_G L_2) \sqcap_G L_3 = Reduce(S(S(L_1,L_2),L_3))$;

b) $Reduce(S(S(L_1,L_2),L_3)) = Reduce(S(L_1,S(L_2,L_3)))$;

c) $Reduce(S(L_1,S(L_2,L_3))) = L_1 \sqcap_G (L_2 \sqcap_G L_3)$.

$(L_1\sqcap_G L_2)\sqcap_G L_3\sqsubseteq L_1\sqcap_G L_2\sqsubseteq S(L_1,L_2)$.根据引理 2,$(L_1\sqcap_G L_2)\sqcap_G L_3\sqsubseteq S(L_1,L_2)\sqcap_G L_3$.同理可以证明 $S(L_1,L_2)\sqcap_G L_3\sqsubseteq(L_1\sqcap_G L_2)\sqcap_G L_3$.因此,$(L_1\sqcap_G L_2)\sqcap_G L_3=S(L_1,L_2)\sqcap_G L_3=Reduce(S(S(L_1,L_2),L_3))$.同理可以证明 $Reduce(S(L_1,S(L_2,L_3)))=L_1\sqcap_G(L_2\sqcap_G L_3)$.根据函数 S 的定义,$S(S(L_1,L_2),L_3)=S(L_1,S(L_2,L_3))$可证.因此,结论 3)可证.$L_G=\{S|S\subseteq ProperP\wedge\forall p_1,p_2\in S, p_1\preceq p_2\Rightarrow p_1=p_2\}$.因此 $L_G\subseteq 2^{ProperP}$.根据推论 1,$ProperP$ 是有穷的,因此$(L_G,\sqcap_G)$的高度有穷. □

定理2的证明

证明文中的定理 2 之前,首先证明下面的引理:

引理 5. 对任何 $L_1,L_2\in L_G$,如果 $L_1\sqsubseteq L_2$,那么 $Semantics(n,L_1)\sqsubseteq Semantics(n,L_2)$.

证明:根据 *Semantics* 的定义:

(1) 如果 n 是 *cond*,那么结论显然成立;

(2) 如果 n 是 $lh:=e$,对任何 $lh_i\in LH$,如果$\wedge L_1\Rightarrow\&lh\neq\&lh_i$,那么$\wedge L_2\Rightarrow\&lh\neq\&lh_i$.因此:

$$Semantics(n,L_1)\sqsubseteq Semantics(n,L_2).$$

综上,$Semantics(n,L_1)\sqsubseteq Semantics(n,L_2)$. □

引理 6. 对任何性质集合 S_1 和 S_2,如果 $S_1\sqsubseteq S_2$,那么 $Propagated(S_1)\sqsubseteq Propagated(S_2)$.

证明:对任何 $ins_1\subseteq S_1$,如果规则 r 可以应用 ins_1,并且产生输出性质 out_1,根据 $S_1\sqsubseteq S_2$ 和传播规则约束,一定存在 $ins_2\subseteq S_2$ 和一个对应的规则 r',使得:(a) $ins_1\sqsubseteq ins_2$;(b) r'可以应用到 ins_2;(c) r'和 ins_2 产生性质 out_2 并且 $out_1\preceq out_2$.因此,$Propagated(S_1)\sqsubseteq Propagated(S_2)$. □

引理 7. 对任何 $x,y\in L_G$,如果 $x\sqsubseteq y$,$Transfer(e_1[e_2]:=e_3,x)\sqsubseteq Transfer(e_1[e_2]:=e_3,y)$.

证明:令 $TS(x)=\{p|p\in x\wedge p$ 是全称量词$\wedge\&e_1[e_2]\notin M(p)\}$.根据 *Transfer* 函数的定义,只需要证明 $TS(x)\sqsubseteq TS(y)$.对任何 $p_1\in TS(x)$,肯定存在 $p_2\in y$ 并且 $p_1\preceq p_2$.根据推论 2,$M(p_1)\subseteq M(p_2)$.$\wedge x\Rightarrow\&e_1[e_2]\notin M(p_1)$推出$\wedge y\Rightarrow\&e_1[e_2]\notin M(p_2)$,因此,$p_2\in TS(y)$.综上,$TS(x)\sqsubseteq TS(y)$. □

引理 8. 对任何 $x,y\in L_G$,如果 n 是 $i:=i+c$,$x\sqsubseteq y$,那么 $HandleInterval(n,x)\sqsubseteq HandleInterval(n,y)$.

证明:下面分情况讨论:

(1) 如果$\forall k(k\in[init_i,c,i+c)\Rightarrow p)$在 x 中,那么肯定存在$\forall k(k\in[init_i,c,i+c)\Rightarrow p')$在 y 中,因为$\{\forall k(k\in[init_i,c,i)\Rightarrow p)\}\sqsubseteq\{\forall k(k\in[init_i,c,i)\Rightarrow p')\}$,因此 $HandleInterval(n,x)\sqsubseteq HandleInterval(n,y)$;

(2) 如果$\forall k(k\in[init_i,c,i)\Rightarrow p)$在 x 中,那么肯定存在$\forall k(k\in[init_i,c,i)\Rightarrow p')$在 y 中,因为$\{\forall k(k\in[init_i,c,i-c)\Rightarrow p)\}\sqsubseteq\{\forall k(k\in[init_i,c,i-c)\Rightarrow p')\}$,因此 $HandleInterval(n,x)\sqsubseteq HandleInterval(n,y)$;

(3) 如果 $i=init_i$ 在 x 中,那么肯定存在 $i=init_i$ 在 y 中,因此 $HandleInterval(n,x)\sqsubseteq HandleInterval(n,y)$;

(4) 其他情况,$HandleInterval(n,x)\sqsubseteq HandleInterval(n,y)$.

综上,$HandleInterval(n,x)\sqsubseteq HandleInterval(n,y)$. □

引理 9. 对任何 $S_1,S_2\in 2^{ProperP}$,如果 $S_1\sqsubseteq S_2$,那么 $GenAQ(S_1)\sqsubseteq GenAQ(S_2)$.

证明:令 $\psi(k)$表示 $\psi(\ldots,e_1[f_1(k)],\ldots)$的缩写:

(1) 如果在 x 中条件(1)成立,$GenAQ(S_1)=S_1\cup\{\forall k(k\in[init_i,c,i+c)\Rightarrow\psi(k))\}$.$S_1\sqsubseteq S_2$ 推出 $i=init_i\in S_2$ 并且$\forall k(k\in[init_i,c,i)\Rightarrow$false)在 S_2 中并且$\psi'(i)\in S_2$,其中,$\psi(i)\preceq\psi'(i)$.$GenAQ(S_2)=S_2\cup\{\forall k(k\in[init_i,c,i+c)\Rightarrow\psi'(k))\}$.因此,$GenAQ(S_1)\sqsubseteq GenAQ(S_2)$;

(2) 如果在 x 中条件(2)成立,$GenAQ(S_1)=S_1\cup\{\forall k(k\in[init_i,c,i+c),\psi(k)\}$.$S_1\sqsubseteq S_2$ 推出$\forall k(k\in[init_i,c,i)\Rightarrow\psi_1'(k))$在

S_2 中并且 $\psi'_2(i)\in S_2$,其中,$\psi_1(k)\preceq\psi'_1(k)\wedge\psi_2(i)\preceq\psi'_2(i).\psi(k)=\psi_1(k)\sqcap\psi_2(k)\Rightarrow\psi(k)\preceq\psi_1(k)\wedge\psi(k)\preceq\psi_2(k)\Rightarrow\psi(k)\preceq\psi'_1(k)\wedge\psi(k)\preceq\psi'_2(k)\Rightarrow\psi(k)\preceq\psi'_1(k)\sqcap\psi'_2(k).$令 $\psi'(k)=\psi'_1(k)\sqcap\psi'_2(k).GenAQ(S_2)=S_2\cup\{\forall k(k\in[init_i,c,i+c)\Rightarrow\psi'(k))\}$.因此,$GenAQ(S_1)\sqsubseteq GenAQ(S_2)$;

(3) 如果在 x 中条件(3)和条件(4)成立,证明过程类似条件(1)和条件(2);

(4) 其他情况,$GenAQ(S_1)\sqsubseteq GenAQ(S_2)$显然成立.

综上,$GenAQ(S_1)\sqsubseteq GenAQ(S_2)$. □

推论 6. 如果 n 是循环控制变量初始化语句,$GenSpecial(n)\sqsubseteq GenSpecial(n)$.

推论 7. 令 G 表示 CFG.令 a 表示语句 n 之前的数据流值.如果 $a\in L_G$,那么 $F_n(a)\in L_G$.

定理 2. 令 n 表示 CFG G 的语句.F_n 是单调的.

证明:F_n 是单调的当且仅当$\forall x,y\in L_G:x\sqsubseteq y\Rightarrow F_n(x)\sqsubseteq F_n(y)$.根据 F_n 的定义,只需要证明 F_n 中出现的函数都是单调的.根据引理 5~引理 9 和推论 6、推论 7,F_n 是单调的. □

定理4的证明

引理 10. 令 c_i 表示语句 n 之前的状态,令 a_i 表示语句 n 之前的数据流值.如果 $G\vdash c_i\rightsquigarrow c_{i+1}$ 并且$[\![a_i]\!](c_i)$成立,那么下面的条件成立:

1) $[\![Semantics(n,a_i)]\!](c_{i+1})$成立;

2) 如果 n 是 $e_1[e_2]:=e_3$,那么$[\![Transfer(n,a_i)]\!](c_{i+1})$成立;

3) 如果 n 是循环控制变量初始化语句 $i:=init$,那么$[\![GenSpecial(n)]\!](c_{i+1})$成立;

4) 如果 n 是循环控制变量初始化语句 $i:=i+c$,那么$[\![HandleInterval(n,a_i)]\!](c_{i+1})$成立;

5) 对任何 $S\in 2^{ProperP}$,如果$[\![S]\!](c_{i+1})$成立,那么$[\![GenAQ(S)]\!](c_{i+1})$成立;

6) 对任何 $S\in 2^{ProperP}$,如果$[\![S]\!](c_{i+1})$成立,那么$[\![Recude(S)]\!](c_{i+1})$.

证明:

(1) 要证明结论 1),需要证明:

a) 如果 n 是 $lh:=e$,那么 $[\![lh]\!](c_{i+1})=[\![e]\!](c_i)\wedge\underset{[\![\&lh\neq\&lh_i]\!](c_i)\wedge lh_i\in LH}{\wedge}[\![lh_i]\!](c_{i+1})=[\![lh_i]\!](c_i)$ 成立;

b) 如果 n 是 $cond$,并且:

i c_{i+1} 是 true 分支语句之前的状态,那么$[\![cond]\!](c_{i+1})\wedge\underset{lh_i\in LH}{\wedge}[\![lh_i]\!](c_{i+1})=[\![lh_i]\!](c_i)$;

ii c_{i+1} 是 false 分支语句之前的状态,那么$[\![\neg cond]\!](c_{i+1})\wedge\underset{lh_i\in LH}{\wedge}[\![lh_i]\!](c_{i+1})=[\![lh_i]\!](c_i)$.

根据语句语义的定义,结论 a)、结论 b)成立.

(2) 要证明结论 2),需要证明:

$$[\![\{p|p\in a\wedge p\text{ 是全称量词性质}\wedge\&e_1[e_2]\notin M(p)\}]\!](c_{i+1}).$$

$[\![a_i]\!](c_i)\wedge p\in a_i\Rightarrow[\![p]\!](c_i).(\wedge a_i\Rightarrow\&e_1[e_2]\notin M(p))\wedge[\![a_i]\!](c_i)$推出$[\![\&e_1[e_2]\notin M(p)]\!](c_i)$.因为$[\![\&e_1[e_2]\notin M(p)]\!](c_i),p$ 中所有内存单元的值保持不变,因此$[\![p]\!](c_{i+1})$成立.

(3) 为了证明结论 3),需要证明$[\![\{\forall k(k\in[init,c,i)\Rightarrow\text{false}\}]\!](c_{i+1})$.因为 n 是 $i:=init$ 并且 i 不出现在 $init$ 中,所以$[\![i=init]\!](c_{i+1})$成立.$[\![i=init]\!](c_{i+1})\Rightarrow[\![[init,step,i)=\varnothing]\!](c_{i+1})$,因此$[\![\forall k(k\in[init,c,i)\Rightarrow\text{false}]\!](c_{i+1})$成立.

(4) 为了证明结论 4):

a) 如果$\forall x(x\in[init_i,c,i+c)\Rightarrow p)$在 a_i 中,那么只需要证明$[\![\forall x(x\in[init_i,c,i)\Rightarrow p)]\!](c_{i+1}).(\forall x(x\in[init_i,c,i+c)\Rightarrow p)$在 a_i 中并且$[\![a_i]\!](c_i)$推出$[\![\forall x(x\in[init_i,c,i+c)\Rightarrow p]\!](c_i)$.因为 n 是 $i:=i+c,p$ 不包含 $i,init_i$ 不包含 i,并且$[\![i]\!](c_{i+1})=[\![i+c]\!](c_i)$,所以$[\![\forall x(x\in[init_i,c,i)\Rightarrow p]\!](c_{i+1})$;

b) 如果$(\forall x(x\in[init_i,c,i)\Rightarrow p)$在 a_i 中,那么只需要证明$\forall x(x\in[init_i,c,i-c)\Rightarrow p)(c_{i+1}).\forall x(x\in[init_i,c,i)\Rightarrow p)$在 a_i 中

$\wedge[\![a_i]\!](c_i)$推出$[\![\forall x(x\in[init_i,c,i)\Rightarrow p]\!](c_i)$.因为 n 是 $i:=i+c$,p 不包含 i,$init_i$ 不包含 i,并且$[\![i-c]\!](c_{i+1})=[\![i]\!](c_i)$,所以$[\![\forall x(x\in[init_i,c,i-c)\Rightarrow p)]\!](c_{i+1})$;

c) 如果$(i=init_i)$在 a_i 中,那么只需要证明$[\![i-c=init_i]\!](c_{i+1})$.$(i=init_i)\in a_i\wedge[\![a_i]\!](c_i)$推出$[\![i=init_i]\!](c_i)$.因为 n 是 $i:=i+c$,$init_i$ 不包含 i,并且$[\![i-c]\!](c_{i+1})=[\![i]\!](c_i)$,所以$[\![i-c=init_i]\!](c_{i+1})$;

d) 其他情况,结论 4)显然成立.

综上,结论 4)成立.

(5) 为了证明结论 5):

a) 如果第 4.3.5 节中的条件(1)或者条件(2)成立,那么只需要证明$[\![\{\forall x(x\in[init_i,c,i+c)\Rightarrow\psi(...,e_1[f_1(x)],...)\}]\!](c_{i+1})$成立,其中,$(...,e_1[f_1(x)],...)$不包含 i.令 $\psi(k)$表示 $\psi(...,e_1[f_1(x)],...)$的缩写:

1) 如果条件(1)成立,那么$(\wedge S\Rightarrow(1))\wedge[\![S]\!](c_{i+1})$推出$[\![(1)]\!](c_{i+1})$.$[\![(1)]\!](c_{i+1})\Rightarrow[\![i=init_i\wedge\psi(i)]\!](c_{i+1})$.因为$[\![i=init_i]\!](c_{i+1})$成立,那么$[\![[init_i,c,i+c)]\!](c_{i+1})=[\![\{i\}]\!](c_{i+1})$.因此,$[\![\forall x(x\in[init_i,c,i+c)\Rightarrow\psi(x)]\!](c_{i+1})$成立;

2) 如果条件(2)成立,那么$(\wedge S\Rightarrow(2))\wedge[\![S]\!](c_{i+1})$推出$[\![(2)]\!](c_{i+1})$.因为$[\![(2)]\!](c_{i+1})$,因此$[\![\forall x(x\in[init_i,c,i)\Rightarrow\psi_1(x)\wedge\psi_2(i)]\!](c_{i+1})$成立.因为 $\psi(x)=\psi_1(x)\sqcap\psi_2(x)$,所以 $\psi_2(x)\Rightarrow\psi(x)\wedge\psi_1(x)\Rightarrow\psi(x)$成立.因此,$\forall x(x\in[init_i,c,i)\Rightarrow\psi_1(x))$推出$\forall x(x\in[init_i,c,i)\Rightarrow\psi(x)$.$[\![\forall x(x\in[init_i,c,i)\Rightarrow\psi_1(x)\wedge\psi_2(i)]\!](c_{i+1})$推出$[\![\forall x(x\in[init_i,c,i)\Rightarrow\psi(x)\wedge\psi(i)]\!](c_{i+1})$.因为$[\![[init_i,c,i+c)]\!](c_{i+1})=[\![[init_i,c,i)\cup\{i\}]\!](c_{i+1})$,故$[\![\forall x(x\in[init_i,c,i+c)\Rightarrow\psi(x)]\!](c_{i+1})$成立;

b) 如果第 3.3.6 节中的条件(3)或者条件(4)成立,那么只需要证明$[\![\{\forall x(x\in[init_i,c,i)\Rightarrow\psi(...,e_1[f_1(x)],...)\}]\!](c_{i+1})$成立,其中,$(...,e_1[f_1(x)],...)$不包含 i.它的证明过程类似证明条件(1)和条件(2).

综上,结论 5)成立.

(6) $[\![S]\!](c_{i+1})\Rightarrow[\![Reduce(S)]\!](c_{i+1})$,因此结论 6)成立. □

定理 4. 令 c_i 表示语句 n 之前的状态,a_i 表示语句 n 之前的数据流值.如果 $G\vdash c_i\rightsquigarrow c_{i+1}$ 并且$[\![a_i]\!](c_i)$成立,那么$[\![F_n(a_i)]\!](c_{i+1})$成立.

证明:传播规则正确,所以$[\![Semantics(n,a_i)]\!](c_i\cup c_{i+1})$成立,因此$[\![Propagated(a_i\cup Semantics(n,a_i)]\!](c_{i+1})$成立.根据引理 10 以及 F_n 的定义,$[\![F_n(a_i)]\!](c_{i+1})$成立. □

李彬(1988－),男,河北邯郸人,博士生,主要研究领域为软件工程,程序分析,程序验证.

翟娟(1988－),女,博士,CCF 专业会员,主要研究领域为软件工程,程序分析,程序验证,程序合成.

汤震浩(1989－),男,博士生,主要研究领域为软件工程,程序分析,程序验证.

汤恩义(1982－),男,博士,助理研究员,CCF 专业会员,主要研究领域为软件工程,新型软件测试方法与程序分析方法.

赵建华(1971－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为形式化方法,软件工程,程序设计语言.