$$v'=(\boldsymbol{a}^{**},\boldsymbol{b}_{\boldsymbol{a}}^{**},c^{**})=\left(\frac{\boldsymbol{a}^{*}}{abs(v)},\frac{\boldsymbol{b}_{\boldsymbol{a}}^{*}}{abs(v)},\frac{c^{*}}{abs(v)}\right).$$

这里, $abs(v)=\sum i\,|\,v_i\,|$ 为向量 v 的分量的绝对值之和.显然有, $v'\in[-1,1]^l$ 且 $\hat{\rho}^{**}(\hat{\boldsymbol{x}})=\frac{1}{abs(v)}\hat{\rho}^{*}(\hat{\boldsymbol{x}})=\frac{1}{abs(v)}\cdot\boldsymbol{a}^{*T}\cdot\tau_o, M_i^{**}(\hat{\boldsymbol{x}})=\frac{1}{abs(v)}M_i^{*}(\hat{\boldsymbol{x}})=\frac{1}{abs(v)}\cdot(\boldsymbol{b}_{\boldsymbol{a}}^{*T},-c^{*})\cdot(\tau_o'^{T},z^{d_{H_i}})^T$ 满足公式(14). □

根据定义 4、命题 1~命题 3,我们可以建立下面的结论.

定理 4. 给定程序 U,若存在 $(\boldsymbol{a}^{*},\boldsymbol{b}_{\boldsymbol{a}}^{*},c^{*})$ 的一组取值 $(\boldsymbol{a}^{*},\boldsymbol{b}_{\boldsymbol{a}}^{*},c^{*})\in[-1,1]^{\ell}$,使得 $\hat{\rho}^{*}(\hat{\boldsymbol{x}})=\boldsymbol{a}^{*T}\cdot\tau_0, M_i^{*}(\hat{\boldsymbol{x}})=(\boldsymbol{b}_{\boldsymbol{a}}^{*T},-c^{*})\cdot(\tau_0'^{T},z^{d_{H_i}})^T$ 满足公式(14),那么程序 U 必然终止.

在 S5 中,我们需要判断是否存在 $(\boldsymbol{a}^{*},\boldsymbol{b}_{\boldsymbol{a}}^{*},c^{*})$ 的一组取值 $(\boldsymbol{a}^{*},\boldsymbol{b}_{\boldsymbol{a}}^{*},c^{*})$,使得 $\hat{\rho}^{*}(\hat{\boldsymbol{x}}),M_i^{*}(\hat{\boldsymbol{x}})$ 满足公式(14),这等价于在单形 Δ_{n+1} 上探测是否存在正定多项式 $\hat{\rho}^{*}(\hat{\boldsymbol{x}}),M_i^{*}(\hat{\boldsymbol{x}})$.我们将利用上述的定理 2(Polya 定理)和定理 3 去探测单形上的正定多项式.定理 2 表明,若齐次多项式 f 在单形 Δ_n 上是正定的,则必然存在充分大的正整数 N,使得 $(x_1+\ldots+x_n)^N f(x_1,\ldots,x_n)$ 展开后的所有系数均为正.但 Polya 并没有给出 N 的上界.在定理 3 中,Powers 等人根据 f 的次数、系数以及 f 在单形上的最小值,构造了 N 的界.

$$N>\frac{d(d-1)}{2}\frac{L}{\lambda}-d \tag{15}$$

这里, $L=L(f)=\max\{|b_\alpha|:|\alpha|=d\}$, $\lambda=\lambda(f)=\min\{f(\boldsymbol{x}):\boldsymbol{x}\in\Delta_n\}$.根据下面的结论,我们可以将公式(15)中的 L 替换为 1.

命题 4. 记号同上.给定齐 d 次 n 元多项式 $f(\boldsymbol{x})=\sum_{|\alpha|=d}a_\alpha\boldsymbol{x}^\alpha=\sum_{|\alpha|=d}c(\alpha)b_\alpha\boldsymbol{x}^\alpha$, 这里,非负整数向量 $\alpha=(\alpha_1,\ldots,\alpha_n)\in\mathbf{N}^n$,则 $(\alpha_1+\ldots+\alpha_n)!\geqslant\alpha_1!\ldots\alpha_n!$.

证明:考虑 n=2.

$$(\alpha_1+\alpha_2)!=(\alpha_1-0+\alpha_2)(\alpha_1-1+\alpha_2)(\alpha_1-2+\alpha_2)\ldots(\alpha_1-(\alpha_1-1)+\alpha_2)\cdot(\alpha_2)!$$

显然, $(\alpha_1-0+\alpha_2)\geqslant\alpha_1,(\alpha_1-1+\alpha_2)\geqslant\alpha_1-1,(\alpha_1-2+\alpha_2)\geqslant\alpha_1-2,\ldots,(\alpha_1-(\alpha_1-1)+\alpha_2)\geqslant1$.故

$$(\alpha_1+\alpha_2)!\geqslant\alpha_1!\alpha_2! \tag{16}$$

考虑 n=3.

根据公式(16),有 $(\alpha_1+\alpha_2+\alpha_3)!=(\alpha_3+(\alpha_1+\alpha_2))!\geqslant(\alpha_3)!(\alpha_1+\alpha_2)!\geqslant\alpha_3!\alpha_1!\alpha_2!$.

以此类推,可得 $(\alpha_1+\ldots+\alpha_n)!\geqslant\alpha_1!\ldots\alpha_n!$. □

注:给定齐 d 次 n 元多项式 $f(\boldsymbol{x})=\sum_{|\alpha|=d}a_\alpha\boldsymbol{x}^\alpha=\sum_{|\alpha|=d}c(\alpha)b_\alpha\boldsymbol{x}^\alpha$,在公式(15)中,

$$L=L(f)=\max\left\{|\,b_\alpha\,|=\frac{|\,a_\alpha\,|}{c(\alpha)}:|\,\alpha\,|=d\right\}.$$

根据命题 4 可知, $c(\alpha)=\frac{d!}{\alpha_1!\ldots\alpha_n!}=\frac{(\alpha_1+\ldots+\alpha_n)!}{\alpha_1!\ldots\alpha_n!}\geqslant1$. 假如 f 的所有系数 a_α 均在区间 $[-\eta,\eta]$ 中,那么 $\frac{|\,a_\alpha\,|}{c(\alpha)}\leqslant\eta$. 因此,倘若 f 的所有系数 a_α 均在区间 $[-\eta,\eta]$ 中,那么 $L=L(f)=\max\left\{|\,b_\alpha\,|=\frac{|\,a_\alpha\,|}{c(\alpha)}:|\,\alpha\,|=d\right\}\leqslant\eta$.根据上面的分析,若系数 a_α 在区间 $[-1,1]$,可以得到下列结果:

定理 5. 给定齐 d 次 n 元多项式 $f(\boldsymbol{x})=\sum_{|\alpha|=d}a_\alpha\boldsymbol{x}^\alpha$,且其所有系数 a_α 均在区间 $[-1,1]$ 中.如果 f 在 Δ_n 上是正定的,则当:

$$N>\frac{d(d-1)}{2}\frac{1}{\lambda}-d \tag{17}$$

有 $(x_1+\ldots+x_n)^N f(x_1,\ldots,x_n)$ 的所有系数为正.其中, $\lambda=\lambda(f)=\min\{f(\boldsymbol{x}):\boldsymbol{x}\in\Delta_n$.

根据定理 4 得知,可以在系数空间 $[-1,1]^{\ell}$ 中探测满足公式(14)的齐次多项式 $\hat{\rho}(\hat{\boldsymbol{x}})=\boldsymbol{a}^T\cdot\tau_0, M_i(\hat{\boldsymbol{x}})=(\boldsymbol{b}_{\boldsymbol{a}}^T,-c)\cdot(\tau_0'^T,z^{d_{H_i}})^T$.这里, $(\boldsymbol{a},\boldsymbol{b}_{\boldsymbol{a}},c)\in[-1,1]^{\ell}$.同时,满足公式(14)的 $\hat{\rho}(\hat{\boldsymbol{x}}),M_i(\hat{\boldsymbol{x}})$ 恰好是单形 Δ_{n+1} 上的正定多项式.因此,既然

$\hat{\rho}(\hat{\boldsymbol{x}}),M_i(\hat{\boldsymbol{x}})$ 在单形Δ_{n+1} 上正定且其所有系数均在$[-1,1]^\ell$ 中取值,那么根据定理 5,必然存在一个 N(其值仅仅依赖于最小值λ),使得 $(x_1+...+x_n+z)^N\hat{\rho}(\hat{\boldsymbol{x}}),(x_1+...+x_n+z)^N M_i(\hat{\boldsymbol{x}})$ 的所有系数为正.因此,为了得到$(\boldsymbol{a},\boldsymbol{b_a},c)$的一组取值,我们可以从 $(x_1+...+x_n+z)^N\hat{\rho}(\hat{\boldsymbol{x}}),(x_1+...+x_n+z)^N M_i(\hat{\boldsymbol{x}})$ 中抽取出所有系数(它们均是关于$(\boldsymbol{a},\boldsymbol{b_a},c)$的齐次线性表达式),并令所有系数大于 0,且加上约束 $c>0\wedge(\boldsymbol{a},\boldsymbol{b_a},c)\in[-1,1]^\ell$,从而构造出关于$(\boldsymbol{a},\boldsymbol{b_a},c)$的不等式组 *Sys*.若 *Sys* 有实解,则它的任一组解 $(\boldsymbol{a}^*,\boldsymbol{b}_{\boldsymbol{a}}^*,c^*)$ 就正好给出了单形Δ_{n+1} 上的(半)正定多项式 $\hat{\rho}^*(\hat{\boldsymbol{x}}),M_i^*(\hat{\boldsymbol{x}})$ (这是因为根据定理 2 或定理 3 的注解可知,若存在 N,使得 $\left(\sum_{i=1}^n x_i\right)^N f$ 展开后各项系数为正,则f在单形是半正定的).下面的定理表明,若系统 *Sys* 有实解,则程序 U 可终止.

定理 6. 记号同上.给定程序 U,若系统 *Sys* 有实解,则程序 U 是终止的.

证明:不妨令 $(\boldsymbol{a}^*,\boldsymbol{b}_{\boldsymbol{a}}^*,c^*)$ 为系统 *Sys* 的一组解,则有$c^*>0\wedge(\boldsymbol{a}^*,\boldsymbol{b}_{\boldsymbol{a}}^*,c^*)\in[-1,1]^\ell$;且存在 N,使得$(x_1+...+x_n+z)^N\cdot\hat{\rho}^*(\hat{\boldsymbol{x}}),(x_1+...+x_n+z)^N M_i^*(\hat{\boldsymbol{x}})$ 的各项系数为正.这里,$M_i^*(\hat{\boldsymbol{x}})=(\boldsymbol{b}_{\boldsymbol{a}}^{*T},-c^*)\cdot(\Gamma_0'^T,z^{d_{H_i}})^T,\hat{\rho}^*(\hat{\boldsymbol{x}})=\boldsymbol{a}^{*T}\cdot\Gamma_0$.因此,根据定理 2 或定理 3 的注解可知,$\hat{\rho}^*(\hat{\boldsymbol{x}}),M_i^*(\hat{\boldsymbol{x}})$ 是单形Δ_{n+1} 上的半正定多项式,即

$$\forall\hat{\boldsymbol{x}},(\hat{\boldsymbol{x}}\in\Delta_{n+1}\Rightarrow\hat{\rho}^*(\hat{\boldsymbol{x}})\geqslant 0)\text{且}\mathop{\wedge}_{i=1}^{m}(\forall\hat{\boldsymbol{x}},(\hat{\boldsymbol{x}}\in\Delta_{n+1}\Rightarrow M_i^*(\hat{\boldsymbol{x}})\geqslant 0)).$$

根据命题 2 及其注解,上式等价于

$$\forall\hat{\boldsymbol{x}},(\hat{\boldsymbol{x}}\in\Omega_{\hat{\boldsymbol{x}}}^o\Rightarrow\hat{\rho}^*(\hat{\boldsymbol{x}})\geqslant 0)\text{且}\mathop{\wedge}_{i=1}^{m}(\forall\hat{\boldsymbol{x}},(\hat{\boldsymbol{x}}\in\Omega_{\hat{\boldsymbol{x}}}^o\Rightarrow M_i^*(\hat{\boldsymbol{x}})\geqslant 0)).$$

也即 $\forall\hat{\boldsymbol{x}},(\hat{\boldsymbol{x}}\in\Omega_{\hat{\boldsymbol{x}}}^o\Rightarrow\hat{\rho}^*(\hat{\boldsymbol{x}})\geqslant 0)\text{且}\mathop{\wedge}_{i=1}^{m}(\forall\hat{\boldsymbol{x}},(\hat{\boldsymbol{x}}\in\Omega_{\hat{\boldsymbol{x}}}^o\Rightarrow M_i^*(\hat{\boldsymbol{x}})=M_i^*(\boldsymbol{x},z)=\hat{H}_i^*(\hat{\boldsymbol{x}})-c^*\cdot z^{d_{H_i}}\geqslant 0))$

在上式中令 z=1,得到:

$$\forall\boldsymbol{x},(\boldsymbol{x}\in\Omega_{\boldsymbol{x}}\Rightarrow\rho^*(\boldsymbol{x})\geqslant 0)\text{且}\mathop{\wedge}_{i=1}^{m}(\forall\boldsymbol{x},(\boldsymbol{x}\in\Omega_{\boldsymbol{x}}\Rightarrow\rho^*(\boldsymbol{x})-\rho^*(T_i(\boldsymbol{x}))-c^*\geqslant 0)).$$

根据定义 1 可知,满足上式的$\rho^*(\boldsymbol{x})$是程序 U 的秩函数.故程序 U 可终止. □

注:根据定理 6,若系统 *Sys* 有解,则能够得到一个最小值为预设定值λ^*的具有预定形式的秩函数;反之,若系统 *Sys* 无解,则表明在单形上没有最小值为预设定值λ^*且具有预定形式的正定多项式.

根据定理 4,$\hat{\rho}^*(\hat{\boldsymbol{x}})$ 的存在,表明了程序 U 是终止的.综上所述,N 的计算是关键的.在公式(17)中,N 的值仅依赖于多项式的次数 d 和齐次多项式在单形上的最小值λ.但由于 $\hat{\rho}(\hat{\boldsymbol{x}}),M_i(\hat{\boldsymbol{x}})$ 的所有系数均是关于参数$(\boldsymbol{a},\boldsymbol{b_a},c)$的线性表达式,故它们在单形$\Delta_{n+1}$ 上的最小值是关于$(\boldsymbol{a},\boldsymbol{b_a},c)$的函数,即 $\lambda_{\hat{\rho}(\hat{\boldsymbol{x}})}=\lambda(\boldsymbol{a},\boldsymbol{b_a},c),\lambda_{M_i(\hat{\boldsymbol{x}})}=\lambda(\boldsymbol{a},\boldsymbol{b_a},c)$.因此在实际的计算中,我们需事先给定 $\lambda_{\hat{\rho}(\hat{\boldsymbol{x}})},\lambda_{M_i(\hat{\boldsymbol{x}})}$ 的值,然后根据公式(17)计算得到 N 的值并抽取 $(x_1+...+x_n+z)^N\hat{\rho}(\hat{\boldsymbol{x}}),(x_1+...+x_n+z)^N M_i(\hat{\boldsymbol{x}})$ 的所有系数构造上述的不等式组 *Sys*.再根据定理 6,若 *Sys* 有实解,则获得一个满足预定模板形式的秩函数;但若该不等式组没有实解,则需要再次设定新的最小值 $\lambda_{\hat{\rho}(\hat{\boldsymbol{x}})},\lambda_{M_i(\hat{\boldsymbol{x}})}$,并使其值小于上一次设定的最小值(这是因为根据公式(17)可知,当最小值越小时,N 所需的下界值越大).因此,上述秩函数的探测过程是试探性(heuristic)的.当然,为了保证探测过程终止,可人为固定探测的深度 *depth*.根据上述过程,我们建立下列算法 1.

算法 1.

输入:一个程序 U,探测深度 *depth*, $\lambda_{\hat{\rho}(\hat{\boldsymbol{x}})}=\lambda_{M_i(\hat{\boldsymbol{x}})}=\lambda>0$, 变元个数 n,次数 d.

输出:一个具有预定形式的 n 元 d 次多项式秩函数.

Step 1: 设定多项式秩函数模板$\rho(\boldsymbol{x})=\boldsymbol{a}^T\cdot\Gamma$

Step 2: 构造齐次多项式 $\hat{\rho}(\hat{\boldsymbol{x}})=\boldsymbol{a}^T\cdot\Gamma_0,M_i(\hat{\boldsymbol{x}})=\hat{H}_i(\hat{\boldsymbol{x}})-c\cdot z^{d_{H_i}}$

Step 3: **For** i=1 to *depth*

Step 3.1: 根据 $\lambda_{\hat{\rho}(\hat{\boldsymbol{x}})},\lambda_{M_i(\hat{\boldsymbol{x}})}$ 的值以及公式(17)分别计算 $N_{\hat{\rho}},N_{M_i}$

Step 3.2: 抽取 $(x_1+...+x_n+z)^{N_\rho}\hat{\rho}(\hat{\boldsymbol{x}}),(x_1+...+x_n+z)^{N_{M_i}}M_i(\hat{\boldsymbol{x}})$ 的所有系数构造不等式组 *Sys*

© 中国科学院软件研究所 http://www.jos.org.cn

Step 3.3: 用线性规划工具 Simplex 计算 *Sys*:若 *Sys* 有解,则输出多项式秩函数$\rho^*(\boldsymbol{x})$;否则,转 Step 3.4

Step 3.4: 令 $\lambda := \frac{\lambda}{2}; \lambda_{\hat{\rho}(\hat{\boldsymbol{x}})} := \lambda; \lambda_{M_i(\hat{\boldsymbol{x}})} := \lambda$ (串行赋值).

下面通过一个例子来阐述本文的方法.

例 1:考虑循环程序:

$$\left.\begin{array}{l} U_1 \textbf{ while } x \geqslant 0 \wedge y \geqslant 0 \textbf{ do} \\ \tau_1 : x := 1-3x-4y-x^3; \\ y := -x-y; \\ \text{or} \\ \tau_2 : x := -y-1; \\ y := -7x^2+y; \end{array}\right\} \tag{18}$$

记 $T_1=(1-3x-4y-x^3,-x-y), T_2=(-y-1,-7x^2+y)$.

(a) 设定秩函数模板$\rho(\boldsymbol{x})=a_1x^3+a_2x^2y+a_3xy^2+a_4x^2+a_5xy+a_6y^2+a_7x+a_8y+a_9$,令

$$H_1(\boldsymbol{x})=\rho(\boldsymbol{x})-\rho(T_1(\boldsymbol{x})), H_2(\boldsymbol{x})=\rho(\boldsymbol{x})-\rho(T_2(\boldsymbol{x})).$$

(b) 对 $H_i(\boldsymbol{x})-c$ 进行齐次化,得到 $\hat{\rho}(\hat{\boldsymbol{x}}), M_1(\hat{\boldsymbol{x}}), M_2(\hat{\boldsymbol{x}})$,其中,

$$\hat{\rho}(x,y,z) = a_1x^3 + a_2x^2y + a_3xy^2 + a_4x^2z + a_5xyz + a_6y^2z + a_7xz^2 + a_8yz^2 + a_9z^3.$$

由于 $M_1(\hat{\boldsymbol{x}}), M_2(\hat{\boldsymbol{x}})$ 表达式过长,在此省略.

(c) 判定是否存在$(a_1,...,a_9)$的一组取值 $(a_1^*,...,a_9^*)$,使得 $\hat{\rho}(x,y,z)$ 和 $M_1(x,y,z), M_2(x,y,z)$在单形Δ_{2+1}上都同时正定.要计算得到那样的一组$(a_1,...,a_9)$的取值,根据定理 5,需首先计算出 N 的值.根据公式(17),N 依赖于 $\hat{\rho}, M_1, M_2$ 各自次数以及各自在Δ_3 上的最小值λ.但这里, $\hat{\rho}(x,y,z)$ 和 $M_1(x,y,z), M_2(x,y,z)$均是参系数齐次多项式,故其在单形上的最小值是随着参数取值的变动而变动的.因此,在 N 的实际计算中,我们需要事先固定λ的值,比如可令 $\lambda_{\hat{\rho}} = \lambda_{M_1} = \lambda_{M_2} = \lambda = \frac{1}{3}$,然后,既然该例中 $\hat{\rho}(x,y,z)$ 和$M_1(x,y,z), M_2(x,y,z)$的次数分别为3,9,5,那么根据公式(17),分别计算对应的 N 的取值范围为

$$N_{\hat{\rho}} > N\left(\hat{\rho}, \lambda = \frac{1}{3}\right) = 6,\ N_{M_1} > N\left(M_1, \lambda = \frac{1}{3}\right) = 99,\ N_{M_2} > N\left(M_2, \lambda = \frac{1}{3}\right) = 25.$$

故取 $N_{\hat{\rho}} = 7, N_{M_1} = 100, N_{M_2} = 26$ (当然,我们也可以令 $N_{\hat{\rho}} = N_{M_1} = N_{M_2} = 100$,即取公共界).根据定理 5,分别展开多项式 $(x+y+z)^{N_{\hat{\rho}}}\hat{\rho}(x,y,z), (x+y+z)^{N_{M_1}}M_1(x,y,z), (x+y+z)^{N_{M_2}}M_2(x,y,z)$, 合并同类项后,各自抽取出关于 x,y,z 项的所有系数(均为关于 $a_1,...,a_9,c$ 的线性表达式),然后再令所有系数大于 0,构造出关于 $a_1,...,a_9,c$ 的严格不等式组(为参系数的第 1 个约束系统),分别记为 $coe_{\hat{\rho}}, coe_{M_1}, coe_{M_2}$. 同时,注意定理 5 成立的前提条件是, $\hat{\rho}, M_1, M_2$ 关于 x,y,z 的所有系数都被限定在[−1,1]时,公式(17)才成立——即 N 的值才仅与其次数 d 和其在单形上的最小值λ相关.因此,再次分别从多项式 $\hat{\rho}, M_1, M_2$ 中提取其关于 x,y,z 的所有系数(均为关于 $a_1,...,a_9,c$ 的线性表达式),然后分别令每个系数≥−1 且≤1,由此构造出 $\hat{\rho}, M_1, M_2$ 关于 x,y,z 的系数的第 2 个约束系统,分别记为 $\Theta_{\hat{\rho}}, \Theta_{M_1}, \Theta_{M_2}$. 比如,在该例中,我们有:

$$\Theta_{\hat{\rho}} := a_1 \geqslant -1 \wedge a_1 \leqslant 1 \wedge a_2 \geqslant -1 \wedge a_2 \leqslant 1 \wedge a_3 \geqslant -1 \wedge a_3 \leqslant 1 \wedge a_4 \geqslant -1 \wedge a_4 \leqslant 1 \wedge a_5 \geqslant -1 \wedge a_5 \leqslant 1 \wedge$$
$$a_6 \geqslant -1 \wedge a_6 \leqslant 1 \wedge a_7 \geqslant -1 \wedge a_7 \leqslant 1 \wedge a_8 \geqslant -1 \wedge a_8 \leqslant 1 \wedge a_9 \geqslant -1 \wedge a_9 \leqslant 1.$$

同时,根据秩函数的定义 4 可知,$c>0$.最后,求解不等式组 $Sys := coe_{\hat{\rho}} \wedge coe_{M_1} \wedge coe_{M_2} \wedge c > 0 \wedge \Theta_{\hat{\rho}} \wedge \Theta_{M_1} \wedge \Theta_{M_2}$. 若 *Sys* 有解,则表明该程序具有预定形式的 3 次秩函数.根据定理 4 可知,该程序是终止的.但是因为 Maple 自带的线性规划工具 Simplex 仅能够求解非严格的不等式组——不等式组中的所有不等式都是非严格的,所以为了使用工具 Simplex,在实际计算中,我们让 $(x+y+z)^{N_{\hat{\rho}}}\hat{\rho}(x,y,z), (x+y+z)^{N_{M_1}}M_1(x,y,z), (x+y+z)^{N_{M_2}}M_2(x,y,z)$, 合并同类项后,各自抽取出关于 x,y,z 项的所有系数,然后再令所有系数≥某个正数δ.这相当于对 $coe_{\hat{\rho}}, coe_{M_1}$,

© 中国科学院软件研究所 http://www.jos.org.cn

coe_{M_2} 中的所有不等式做了一个小小的扰动,即将>0 替换为≥δ(δ>0).记扰动后的系数集为 $\widetilde{coe}_{\hat{\rho}},\widetilde{coe}_{M_1},\widetilde{coe}_{M_2}$.同时,将 c>0 扰动为 $c\geqslant\delta$.显然,扰动后的系统 $\widetilde{Sys}:=\widetilde{coe}_{\hat{\rho}}\wedge\widetilde{coe}_{M_1}\wedge\widetilde{coe}_{M_2}\wedge c\geqslant\delta\wedge\Theta_{\hat{\rho}}\wedge\Theta_{M_1}\wedge\Theta_{M_2}$ 为非严格不等式组,如果它有解,则原系统 Sys 也必有解.为方便,我们让 $Sys,\widetilde{Sys}$ 不仅表示不等式系统,而且还表示不等式系统所对应的解集.在本例中,我们取 $\delta=\frac{1}{1000}$.利用 Maple 中的线性规划工具包 Simplex,求解不等式组 $\widetilde{Sys}$,得到:

$$\left\{\begin{array}{l}a_1=\frac{29}{7000},a_2=\frac{51}{12250},a_3=\frac{1}{980},a_4=-\frac{43}{7000},a_5=0,a_6=\frac{1}{1000},a_7=0,\\ a_8=\frac{955875294522846847882517}{154330444367487258828048000},a_9=\frac{9}{875},c=\frac{1}{1000}\end{array}\right\} \tag{19}$$

由于采用单纯形算法,故该点是满足 $\widetilde{Sys}$ 的一个精确点而非浮点向量.因此,我们可以得到程序 U_1 的一个 3 次秩函数 $\rho(x,y)=\frac{29x^3}{7000}+\frac{51x^2y}{12250}+\frac{xy^2}{980}-\frac{43x^2}{7000}+\frac{y^2}{1000}+\frac{955875294522846847882517}{154330444367487258828048000}y+\frac{9}{875}$.由于单纯形算法给出的是满足系统 $\widetilde{Sys}(\subseteq Sys)$ 的精确解而非数值解,故求得的函数必然是秩函数,因而不必再对其验证是否为秩函数.该秩函数的存在表明,该循环程序是终止的.

注:对于该程序,文献[3,5,6,20]中的方法已不能计算其预定形式的 3 次秩函数.因为这些方法仅能计算线性循环程序的线性秩函数,而本例中的循环是非线性的,且所要计算的秩函数也是 3 次的.此外,对该程序,我们也尝试利用一些基于量词消去技术[22]的工具,如 Redlog、RegularChains,去计算该程序的预定形式的 3 次秩函数(其中,RegularChains 集成了当前功能强大的实解分类工具 DISCOVERER[23],作者此前的诸多工作也是在该工具的支持下予以开展).这等价于利用这些量词消去工具从秩函数定义 4 中的两个蕴含公式(8)和公式(9)中消去变元 x,y 即可.但由于量词消去算法的双指数复杂度以及所处理的系统都是非线性的,从而使得这两个工具均无法计算得到该程序的 3 次秩函数.同时,我们也尝试用 Cousot 在文献[9]中的方法——半正定规划(SDP),并借助半正定规划求解器 YALMIP[24]计算得到一个函数:

$$42.87498608+4.1519x+25.9747y+1.5329x^3-0.8212y^2-8.7046x^2+3.9678xy-1.8358x^2y+1.8393xy^2.$$

但经过检验发现,该函数并不满足秩函数定义中的有界条件,因此它不是程序的秩函数.

例 2:考虑循环程序:

U_2 **while** $x\geqslant 0\wedge y\geqslant 0$ **do**
$\tau_1: x:=1+x-y;$
$y:=2x+y+3;$
or
$\tau_2: x:=x-y;$
$y:=y+2;$

通过本文方法,我们得到该循环的一个 3 次秩函数:

$$\rho(x,y)=\frac{56978573}{514034040000}x^3-\frac{10604929}{257017020000}x^2y+\frac{1155787}{79082160000}xy^2-\frac{16927733}{171344680000}xy+\frac{1}{52000}y^2+\frac{7408419}{26360720000}x-\frac{7}{13000}y+\frac{203}{26000}.$$

例 3:考虑循环程序:

$$\left.\begin{array}{l}U_3\ \textbf{while}\ x\geqslant 0\wedge y\geqslant 0\ \textbf{do}\\ \tau_1: x:=-x-3y+2;\\ y:=-4x^2+y-1;\\ \text{or}\\ \tau_2: x:=x-1;\\ y:=-y^2+x-3;\end{array}\right\} \tag{20}$$

© 中国科学院软件研究所 http://www.jos.org.cn

通过本文方法,我们得到该循环的一个 3 次秩函数 $\rho(x,y)=\frac{1}{4600}x^3+\frac{63}{23000}y$.

基于 Polya 定理,算法 1 提供了一个试探性方法去探测程序 U 的多项式秩函数.下面我们将证明,给定齐 d 次 n 元多项式 $f(\boldsymbol{x})=\sum_{|\alpha|=d}a_\alpha\boldsymbol{x}^\alpha=\sum_{|\alpha|=d}c(\alpha)b_\alpha\boldsymbol{x}^\alpha$, 若

(A) 所有系数均为整数,即 $a_\alpha\in\mathbf{Z}$,

(B) 其系数的绝对值被界定,即存在正数 η,使得 $|a_\alpha|\leqslant\eta$,

则 Polya 定理中关于 N 的下界公式可以被改写为

$$N>\frac{d(d-1)}{2}\frac{\eta}{\sigma(n,d,\eta)}-d \tag{21}$$

显然,上述 N 的下界公式仅依赖于多项式 $f(\boldsymbol{x})$的次数、变元个数以及系数的绝对值上界.根据公式(21),我们将建立不同于算法 1 的新算法.在证明上述结论之前,我们首先引入文献[25]中建立的关于整系数正定多项式在单形上的最小值下界的一个重要结果.

定理 7[25]. 给定 d 次 n 元齐次整系数多项式 $f(\boldsymbol{x})=\sum_{|\alpha|=d}a_\alpha\boldsymbol{x}^\alpha=\sum_{|\alpha|=d}c(\alpha)b_\alpha\boldsymbol{x}^\alpha\in Z[\boldsymbol{x}]$,其系数均被正数 η 界定,即 $|a_\alpha|\leqslant\eta$.记 $\Delta_n=\{(x_1,...,x_n):x_i\geqslant 0,\sum_{i=1}^n x_i=1\}$ 为单形.如果 $f(\boldsymbol{x})$在单形 Δ_n 上是正定的,那么有:

$$\min_{\Delta_n}f(\boldsymbol{x})\geqslant(2\eta)^{-d^n}n^{-d^{n+1}-d}d^{-nd^n} \tag{22}$$

注:因本文仅考虑单形上的正定多项式,故定理 7 只引述了文献[25]中引理 3.3 中的前部分结论,其后部分所涉及到单形上的负定多项式的内容在此被舍去.

根据定理 2、定理 3 及其定理 7,我们建立下列定理 8.

定理 8. 给定 d 次 n 元齐次整系数多项式 $f(\boldsymbol{x})=\sum_{|\alpha|=d}a_\alpha\boldsymbol{x}^\alpha=\sum_{|\alpha|=d}c(\alpha)b_\alpha\boldsymbol{x}^\alpha\in\mathbf{Z}[\boldsymbol{x}]$,且其系数均被正数 η 界定,即 $|a_\alpha|\leqslant\eta$.如果 f 在 Δ_n 上是正定的,那么当

$$N>\frac{d(d-1)}{2}\frac{\eta}{\sigma(n,d,\eta)}-d \tag{23}$$

时,有 $(x_1+\ldots+x_n)^N f(x_1,\ldots,x_n)$的所有系数为正.这里, $\sigma(n,d,\eta)=(2\eta)^{-d^n}n^{-d^{n+1}-d}d^{-nd^n}$.

证明:首先,根据定理 3 可知,如果 n 元齐 d 次多项式 $f(\boldsymbol{x})$在单形上正定,那么存在正整数 N,当其满足:

$$N>\frac{d(d-1)}{2}\frac{L}{\lambda}-d \tag{24}$$

时,有 $(x_1+\ldots+x_n)^N f(x_1,\ldots,x_n)$的所有系数为正.其中,$L=L(f)=\max\{|b_\alpha|:|\alpha|=d\}$,$\lambda=\lambda(f)=\min\{f(\boldsymbol{x}):\boldsymbol{x}\in\Delta_n\}$.再根据命题 4 及其注解可知,倘若 f 的所有系数 a_α均在区间 $[-\eta,\eta]$中,那么,

$$L=L(f)=\max\left\{|b_\alpha|=\frac{|a_\alpha|}{c(\alpha)}:|\alpha|=d\right\}\leqslant\eta \tag{25}$$

同时,根据已知题设,既然 f 在单形上正定且其所有系数均为整数并均被正数 η 界定,那么由定理 7 可得,f 在单形 Δ_n 上的最小值:

$$\lambda_f=\min_{\Delta_n}f(\boldsymbol{x})\geqslant(2\eta)^{-d^n}n^{-d^{n+1}-d}d^{-nd^n} \tag{26}$$

根据公式(25)和公式(26)将公式(24)进行放缩,即得到公式(23). □

注:定理 8 中,公式(23)中 N 的下界公式仅仅依赖多项式的次数、变元个数以及系数绝对值上界.

给定带参数的多项式 $G(y_1,...,y_n)=\sum_{|\alpha|=d}g_\alpha(\boldsymbol{v})\boldsymbol{y}^\alpha$ (这里,$\boldsymbol{v}$ 为参数向量,$g_\alpha(\boldsymbol{v})$为参数 $\boldsymbol{v}$ 的齐次整系数线性函数),倘若我们限定其系数 $g_\alpha(\boldsymbol{v})$在某个区间取值,即 $-\eta\leqslant g_\alpha(\boldsymbol{v})\leqslant\eta$,那么根据定理 8,完成下列两个步骤便可以构造出使得多项式 G 在单形 Δ_n 上正定且其系数满足 $|g_\alpha(\boldsymbol{v})|\leqslant\eta$的参数 $\boldsymbol{v}$ 应满足的必要条件 $Sys_G^{\mathbf{z}}$ (与之前的 Sys 以示区别).

(i) 根据公式(23)计算 N 的值,记为 N_G;

© 中国科学院软件研究所 http://www.jos.org.cn

(ii) 抽取 $(y_1+...+y_n)^{N_G}G(y_1,...,y_n)$ 的所有系数 $g_\alpha(\boldsymbol{v})$,分别令 $g_\alpha(\boldsymbol{v})>0$,且$|g_\alpha(\boldsymbol{v})|\leqslant\eta$,构造不等式组 $Sys_G^{\mathbf{z}}$.

这里,$g_\alpha(\boldsymbol{v})$为齐次整系数线性函数这一限定具有一般性.这是因为,若多项式 G 的系数均是有理数,那么总可以通过乘上所有有理数的分母将其系数变为整数.根据公式(5),程序 U 中的所有系数均为有理数,故参数模板多项式 $\hat{\rho}(\hat{\boldsymbol{x}})=\boldsymbol{a}^T\cdot\Gamma_0, M_i(\hat{\boldsymbol{x}})=\sum_{|\alpha|=d_{M_i}}g_\alpha(\boldsymbol{b_a},\boldsymbol{c})\hat{\boldsymbol{x}}^\alpha$ 的所有系数均为有理数.因此,可以将所有有理数系数的分母(若有负号,则将负号放到分子上)都乘在一起,记为β.则在公式(14)中的不等式两端同时乘上β并不会改变不等式的符号,故有 $\beta\cdot\hat{\rho}(\hat{\boldsymbol{x}}),\beta\cdot M_i(\hat{\boldsymbol{x}})$ 的系数均为整数.也即$\beta\cdot g_\alpha(\boldsymbol{b_a},\boldsymbol{c})$是关于参数 $\boldsymbol{b_a},\boldsymbol{c}$ 的齐次整系数线性函数.因此,若 $\hat{\rho}(\hat{\boldsymbol{x}}),M_i(\hat{\boldsymbol{x}})$ 含有有理系数,则可以乘上一个正数β,使得其所有系数为整数.根据公式(23),分别计算 $N_{\hat{\rho}},N_{M_i}$ 并抽取 $(x_1+...+x_n+z)^{N_{\hat{\rho}}}\hat{\rho}(\hat{\boldsymbol{x}}),(x_1+...+x_n+z)^{N_{M_i}}M_i(\hat{\boldsymbol{x}})$ 的所有系数分别构造不等式组 $Sys_{\hat{\rho}}^{\mathbf{z}},Sys_{M_i}^{\mathbf{z}}$. 令 $Sys_{\mathbf{z}}=Sys_{\hat{\rho}}^{\mathbf{z}}\wedge(\wedge_{i=1}^{m}Sys_{M_i}^{\mathbf{z}})$.既然所有参数均被设定为整数,故需要在整数环上求解系统 $Sys_{\mathbf{z}}$.类似定理 6,下面的定理 9 表明,若 $Sys_{\mathbb{Z}}$ 有整数解,则程序 U 是终止的.

定理 9. 记号同上.给定程序 U.若系统 $Sys_{\mathbf{z}}$ 有整数解,那么程序 U 有预定形式的且系数绝对值上界为η的秩函数.

证明:该证明完全类似于定理 6 的证明.在此省略. □

注:定理 9 表明,若系统 $Sys_{\mathbf{z}}$ 有整数解,则程序 U 有一个整系数多项式秩函数;反之,如果系统 $Sys_{\mathbf{z}}$ 没有整数解,则表明在单形上没有预定形式的且系数在$[-\eta,\eta]$的正定多项式.此时需要增大上界η的值继续构造新的系统并求解.因此,整个过程仍是试探性的.

算法 2.

输入:一个程序 U,变元个数 n,次数 d,系数绝对值上界η.

输出:一个具有预定形式的 n 元 d 次整系数多项式秩函数;不确定(unknown).

Step 1: 设定多项式秩函数模板$\rho(\boldsymbol{x})=\boldsymbol{a}^T\cdot\Gamma$

Step 2: 构造齐次多项式 $\hat{\rho}(\hat{\boldsymbol{x}})=\boldsymbol{a}^T\cdot\Gamma_0, M_i(\hat{\boldsymbol{x}})=\hat{H}_i(\hat{\boldsymbol{x}})-c\cdot z^{d_{H_i}}$

Step 3: 如果 $\hat{\rho}(\hat{\boldsymbol{x}}),M_i(\hat{\boldsymbol{x}})$ 含有有理系数,则将所有有理数系数的分母都乘在一起,记为β.令 $\hat{\rho}(\hat{\boldsymbol{x}}):=\beta\cdot\hat{\rho}(\hat{\boldsymbol{x}})$, $M_i(\hat{\boldsymbol{x}}):=\beta\cdot M_i(\hat{\boldsymbol{x}})$,转 Step 4;

Step 4: 根据公式(23)分别计算 $N_{\hat{\rho}},N_{M_i}$

Step 5: 抽取 $(x_1+...+x_n+z)^{N_{\hat{\rho}}}\hat{\rho}(\hat{\boldsymbol{x}}),(x_1+...+x_n+z)^{N_{M_i}}M_i(\hat{\boldsymbol{x}})$ 的所有系数构造分别构造不等式组:

$$Sys_{\hat{\rho}}^{\mathbf{z}},Sys_{M_i}^{\mathbf{z}}$$

Step 6: 利用整数线性规划算法计算 $Sys_{\mathbf{z}}=Sys_{\hat{\rho}}^{\mathbf{z}}\wedge(\wedge_{i=1}^{m}Sys_{M_i}^{\mathbf{z}})$ 中的一个整数解.若 $Sys_{\mathbf{z}}$ 有整数解,则输出整系数多项式秩函数$\rho^*(\boldsymbol{x})$;否则,输出 unknown

注:算法 2 中,我们仅是简单地使 $\hat{\rho}(\hat{\boldsymbol{x}}),M_i(\hat{\boldsymbol{x}})$ 的所有系数绝对值具有相同的上界η.实际上,可以对它们设置不同的系数绝对值上界.在判定 $Sys_{\mathbf{z}}$ 是否有整数解时,可以利用 Pugh 在文献[26]中提出的 Omega test 方法.

3 总 结

针对一类多项式循环程序,本文给出了一种新的方法去计算这类程序的多项式秩函数.该方法将这类程序的秩函数计算归结为单形上的正定多项式的探测问题;然后,利用 Polya 定理,将单形上的正定多项式探测问题归结为线性不等式约束系统的可行问题.从这一角度看,我们的方法是一个“线性化”的方法.而线性不等式系统的可行问题则可以利用(整数)线性规划工具进行求解.相对于现有诸如 Redlog、RegularChains 等基于柱形代数分解的量词消去工具,本文的算法 1 可以在可接受时间内进行复杂秩函数的计算.同时,也不同于基于 SDP 的方法,通过本文方法计算得到的函数是精确的秩函数,因此不必再次验证计算所得的函数是否为秩函数.

致谢 感谢匿名审稿人对本文工作提出的宝贵意见.同时,也感谢中国科学院成都计算机应用研究所的杨路先

© 中国科学院软件研究所 http://www.jos.org.cn

生对本文的修改提供的好建议.

References:

[1] Cook B, Podelski A, Rybalchenko A. Proving program termination. Communications of the ACM, 2011,54(5):88−98.

[2] Chen YH, Xia BC, Yang L, Zhou CC. Discovering non-linear ranking functions by solving semi-algebraic systems. In: Proc. of the 4th Int'l Colloquium on Theoretical Aspects of Computing. Berlin, Heidelberg: Springer-Verlag, 2007. 34−49.

[3] Colo´n M, Sipma HB. Practical methods for proving program termination. In: Proc. of the Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 2002. 227−240.

[4] Podelski A, Rybalchenko A. A complete method for the synthesis of linear ranking functions. In: Proc. of the 5th Int'l Conf. on Verification, Model Checking, and Abstract Interpretation. Berlin, Heidelberg: Springer-Verlag, 2004. 239−251.

[5] Bradley A, Manna Z, Sipma H. The polyranking principle. In: Caires L, Italiano G, Monteiro L, Palamidessi C, Yung M, eds. Proc. of the Automata, Languages and Programming. Berlin, Heidelberg: Springer-Verlag, 2005. 1349−1361.

[6] Bradley A, Manna Z, Sipma H. Linear ranking with reachability. In: Proc. of the Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 2005. 491−504.

[7] Bagnara R, Mesnard F, Pescetti A, Zaffanella E. A new look at the automatic synthesis of linear ranking functions. Information and Computation, 2012,215:47−67.

[8] Ben-Amram AM, Genaim S. On the linear ranking problem for integer linear-constraint loops. In: Proc. of the 40th Annual ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages. New York: ACM Press, 2013. 51−62.

[9] Cousot P. Proving program invariance and termination by parametric abstraction, langrangian relaxation and semidefinite programming. In: Proc. of the 6th Int'l Conf. on Verification, Model Checking, and Abstract Interpretation. Berlin, Heidelberg: Springer-Verlag, 2005. 1−24.

[10] Yang L, Zhou CC, Zhan NJ, Xia BC. Recent advances in program verification through computer algebra. Frontiers of Computer Science in China, 2010, 4(1):1−16.

[11] Chen HY, Cook B, Fuhs C, Nimkar K, *et al*. Proving nontermination via safety. In: Proc. of the 20th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems. Berlin, Heidelberg: Springer-Verlag, 2014. 156−171.

[12] Gupta A, Henzinger T, Majumdar R, *et al*. Proving non-termination. In: Proc. of the 35th Annual ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages. New York: ACM Press, 2008. 147−158.

[13] Tiwari A. Termination of linear programs. In: Proc. of the 16th Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 2004. 70−82.

[14] Braverman M. Termination of integer linear programs. In: Proc. of the 18th Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 2006. 372−385.

[15] Xia BC, Yang L, Zhan NJ, Zhang ZH. Symbolic decision procedure for termination of linear programs. Formal Aspects of Computing, 2009,23(2):171−190.

[16] Bradley A, Manna Z, Sipma H. Termination of polynomial programs. In: Proc. of the 6th Int'l Conf. on Verification, Model Checking, and Abstract Interpretation. Berlin, Heidelberg: Springer-Verlag, 2005. 113−129.

[17] Xia BC, zhang ZH. Termination of linear programs with nonlinear constraints. Journal of Symbolic Computation, 2010,45(11): 1234−1249.

[18] Babic D, Cook B, Hu AJ, *et al*. Proving termination of nonlinear command sequences. Formal Aspects of Computing, 2013,25(3): 389−403.

[19] Liu J, Xu M, Zhan NJ, Zhao HJ. Discovering non-terminating inputs for multi-path polynomial programs. Journal of Systems Science and Complexity, 2014,27(4):1284−1304.

[20] Colo´n M, Sipma HB. Synthesis of linear ranking functions. In: Proc. of the 7th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems. Berlin, Heidelberg: Springer-Verlag, 2001. 67−81.

[21] Powers V, Reznick B. A new bound for Polya's theorem with applications to polynomials positive on polyhedra. Journal of Pure and Applied Algebra, 2001,164(1-2):221−229.

© 中国科学院软件研究所 http://www.jos.org.cn

[22] Collins GE. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In: Proc. of the Automata Theory and Formal Languages. Berlin, Heidelberg: Springer-Verlag, 1975. 134−165.

[23] Yang L, Xia BC. Mechanical Inequality Proving and Automated Discovering. Beijing: Science Press, 2008 (in Chinese).

[24] Löfberg J. YALMIP: A MATLAB toolbox for rapid prototyping of optimization problems. Automatic Control Laboratory, Eidgenössische Technische Hochschule Zürich, 2004. http://control.ee.ethz.ch/joloef/yalmip.msql

[25] Hou XR, Shao JW. Bounds on the number of steps of WDS required for checking the positivity of integral forms. Applied Mathematics and Computation, 2011,217(2):9978−9984.

[26] Pugh W. The omega test: A fast and practical integer programming algorithm for dependence analysis. In: Martin JL, ed. Proc. of the Supercomputing. ACM Press, 1991. 4−13.

附中文参考文献:

[23] 杨路,夏壁灿.不等式机器证明与自动发现.北京:科学出版社,2008.

李轶(1980－),男,重庆人,博士,副研究员,CCF 专业会员,主要研究领域为程序验证,符号计算.

冯勇(1965－),男,博士,研究员,博士生导师,CCF 专业会员,主要研究领域为符号数值计算.

© 中国科学院软件研究所 http://www.jos.org.cn