arXiv:2303.17659v1 [math.AC] 30 Mar 2023

Macaulay2 中的单纯复形

Ben Hersey、Gregory G. Smith 和 Alexandre Zotine

摘要。本文介绍 *SimplicialComplexes* 软件包在*Macaulay2*中的一些特性。

这个更新版本的 *SimplicialComplexes* 软件包在*Macaulay2* [M2], 由 Sorin Popescu、Gregory G. Smith 和 Mike Stillman 最初开发，新增了许多经典例子的构造函数，实现了一种新的单纯映射数据类型，并对方法和文档进行了多项改进。该软件包着重于组合与代数应用，其主要数据类型用于编码抽象单纯复形——即有限集的一个子集族，并且该族在取子集运算下封闭。这些单纯复形是其几何实现的组合对应物；几何实现由欧几里得空间中的点、线段、填充三角形、实心四面体及其更高维类比构成。单纯复形中的子集称为面。基数为 1 的面是其顶点，而极大面（按包含关系排序）则是其单纯形面。单纯复形的维数等于其面最大基数减一。按照组合学惯例，每个非空单纯复形都以空集为面。

在此软件包中，单纯复形由其 Stanley–Reisner 理想表示。顶点对应于多项式环中一组变量的子集，而每个面对应于相应变量的乘积。非面是指不属于该单纯复形的任意顶点子集，每个非面同样对应于变量的乘积。单纯复形的 Stanley–Reisner 理想由对应于其非面的单项式生成；参见 [BH], 中的定义 5.1.2、 [MS], 中的定义 1.6，或 [S] 中的定义 II.1.1。由于在相关多项式环中进行计算通常会非常耗费资源，因此本软件包并非为顶点数量较多的单纯复形而设计。

构造函数。单纯复形的基本构造函数接受两种不同类型的输入。给定一个单项式列表，它会返回包含相应面的最小单纯复形。给定一个根单项式理想 I，它会返回 Stanley–Reisner 理想为 I 的单纯复形。我们使用图 1 中的“蝴蝶结”复形来展示这两种方法。

```
Macaulay2, version 1.20with packages: ConwayPolynomials, Elimination,
  IntegralClosure, InverseSystems, Isomorphism, LLLBases, MinimalPrimes,
  OnlineLookup, PrimaryDecomposition, ReesAlgebra, Saturation, TangentCone

i1 : needsPackage "SimplicialComplexes"; S = QQ[v..z];

i3 : ⋈ = simplicialComplex {v*w*x, x*y*z}

o3 = simplicialComplex | xyz vwx |

o3 : SimplicialComplex

i4 : I = monomialIdeal ⋈

o4 = monomialIdeal (v*y, w*y, v*z, w*z)

o4 : MonomialIdeal of S
```

2020 数学主题分类。05E45, 13F55, 55U10

SimplicialComplexes版本 2.0。



图 1。左侧是蝴蝶结复形 $\bowtie$，右侧是其 Alexander 对偶 $\bowtie^*$

```
i5 : ⋈ = simplicialComplex I

o5 = simplicialComplex | xyz vwx |

o5 : SimplicialComplex

i6 : assert(⋈ === ⋈)
```

该软件包还提供了一些典型单纯复形的便捷构造函数。例如，我们可以根据某些明确三角剖分的约化同调群识别实射影平面和克莱因瓶；参见 [Mu]。

```
i7 : ℙ = realProjectiveSpaceComplex(2, R = ZZ[a..h])

o7 = simplicialComplex | bef aef cdf adf bcf cde bde ace abd abc |

o7 : SimplicialComplex

i8 : for j from 0 to 2 list prune HH_j ℙ

o8 = {0, cokernel | 2 |, 0}

o8 : List

i9 : for j from 0 to 2 list prune HH_j kleinBottleComplex R

o9 = {0, cokernel | 2 |, 0}
                  | 0 |
o9 : List
```

更全面地说，Frank H. Lutz 列举了顶点数较少的单纯复形；参见 [L]。利用此列表，该软件包创建了一个数据库，其中包含至多有 43138 个单纯 2-流形，且其顶点数不超过 10，以及 1343 个单纯 3-流形，且其顶点数不超过 9。我们通过展示具有 f-向量的分布，来演示这一功能；这些向量对应于有 3-流形，且其顶点数为 9。对于所有非负整数 j，j-向量中的第 f 项表示具有 j 个顶点的面的数量。

```
i10 : tally for j from 0 to 1296 list fVector smallManifold(3, 9, j, ZZ[vars(1..9)])

o10 = Tally{{1, 9, 26, 34, 17} => 7  }
            {1, 9, 27, 36, 18} => 23
            {1, 9, 28, 38, 19} => 45
            {1, 9, 29, 40, 20} => 84
            {1, 9, 30, 42, 21} => 128
            {1, 9, 31, 44, 22} => 175
            {1, 9, 32, 46, 23} => 223
            {1, 9, 33, 48, 24} => 231
            {1, 9, 34, 50, 25} => 209
            {1, 9, 35, 52, 26} => 121
            {1, 9, 36, 54, 27} => 51
o10 : Tally
```

利用同一个循环，我们为这些 3-流形中的每一个构造从环面的一个极小三角剖分到前 7 个顶点所诱导的子复形的单纯映射。

```
i11 : T = smallManifold(2, 7, 6, R = ZZ[a..i])

o11 = simplicialComplex | cfg afg beg aeg cdg bdg def bef adf bcf cde ace abd abc |
```

```
o11 : SimplicialComplex
i12 : for j from 0 to 2 list prune HH_j T_2 1
o12 = {0, ZZ, ZZ }
o12 : List
i13 : for j from 0 to 1296 list (
phi := map(smallManifold(3, 9, j, R), T, gens R);
if not isWellDefined phi then continue else phi);
o13 : {}
o13 : List
```

组合拓扑。我们使用蝴蝶结复形来展示对单纯复形的一些关键操作。将单纯复形视为单纯形的子复形，可以得到一种对偶性理论。对于其顶点属于集合 V 的任意单纯复形 Δ，Alexander 对偶是单纯复形 $\Delta^* := \{F \subseteq V \mid V \setminus F \notin \Delta\}$。由于此软件包中的每个单纯复形都有一个底层多项式环，因此该环中的变量构成顶点的一个规范超集。

```
i14 : dual ⋈
o14 = simplicialComplex | wxz vxz wxy vxy |
o14 : SimplicialComplex
  i15 : assert(dual dual ⋈ === ⋈ and dual monomialIdeal ⋈ === monomialIdeal dual ⋈)
```

在代数上，Alexander 对偶性互换了 Stanley–Reisner 理想的极小生成元和不可约分支的角色。

```
i16 : monomialIdeal dual ⋈
o16 = monomialIdeal (v*w, y*z)
o16 : MonomialIdeal of S
i17 : irreducibleDecomposition monomialIdeal ⋈
o17 = {monomialIdeal (v, w), monomialIdeal (y, z)}
o17 : List
```

Alexander 对偶性的拓扑形式给出了单纯复形的约化同调与其对偶的约化上同调之间的一个 Isomorphism；参见 [MS] 中的定理 5.6。

```
i18 : n = numgens ring ⋈
o18 = 5
i19 : assert all(-1..n-1, j -> prune HH^(n-j-3) dual ⋈ == prune HH_j ⋈)
```

若其关联商环 S/I 是 Cohen–Macaulay，则称单纯复形 Δ 为 Cohen–Macaulay；其中，I 是多项式环 S 中 Δ 的 Stanley–Reisner 理想。为了从拓扑角度刻画这一性质，我们引入一族子复形。对于 Δ 中的任意面 F，其链接是子复形 $\mathrm{link}_\Delta(F) := \{G \in \Delta \mid F \cup G \in \Delta$ 和 $F \cap G = \emptyset\}$。⋈ 中顶点 x 的链接有两个不相交的单纯形面。

```
i20 : L = link(⋈, x)
```

```
o20 = simplicialComplex | yz vw |
o20 : SimplicialComplex
i21 : prune HH_0 L_1o21 = QQ
o21 : QQ-module, free
```

正如 Gerald Reisner 所发现的，单纯复形 Δ 是 Cohen–Macaulay 的，当且仅当对于所有面 F 属于 Δ，以及所有小于 j 的维数的整数 $\mathrm{link}_\Delta(F)$，j阶约化同调群 $\mathrm{link}_\Delta(F)$ 均为零；参见 [BH], 中的推论 5.3.9、[MS], 中的定理 5.53，或 [S]中的推论 II.4.2。利用这一判据，0阶约化同调证明了 ⋈不是 Cohen–Macaulay 的。

```
i22 : assert(HH_0 L != 0)
i23 : assert(dim(S^1/monomialIdeal ⋈) =!= n - pdim(S^1/monomialIdeal ⋈))
```

然而， ⋈ 的1-骨架是 Cohen–Macaulay 的。

```
i24 : ⋈ = skeleton(1, ⋈)
o24 = simplicialComplex | yz xz xy wx vx vw |
o24 : SimplicialComplex
i25 : faceList = rsort flatten values faces ⋈
o25 = {v*w, v*x, w*x, x*y, x*z, y*z, v, w, x, y, z, 1}
o25 : List
i26 : assert all(faceList, F -> (L := link(⋈, F); all(dim L, j -> HH_j L == 0)))
i27 : assert(dim(S^1/monomialIdeal ⋈) === n - pdim(S^1/monomialIdeal ⋈))
```

另一种方法是证明 ⋈ 不是 Cohen–Macaulay 的：只需说明其 h-向量含有负项；参见 [BH] 中的定理 5.1.10，或 [S]中的推论 II.2.5。根据定义，单纯复形 h的 Δ -向量是其 f-向量的二项式变换：对于所有 $0 \leqslant j \leqslant d := \dim \Delta$，都有$h_j = \sum_{k=0}^{j} (-1)^{j-1} \binom{d+1-k}{j-k} f_{k-1}$。 h-向量编码了 S/I的希尔伯特级数的分子。

```
i28 : d = dim ⋈
o28 = 2
i29 : faces ⋈
o29 =  HashTable{-1 => {1}                          }
                 0 => {v, w, x, y, z}
                 1 => {v*w, v*x, w*x, x*y, x*z, y*z}
                 2 => {v*w*x, x*y*z}
o29 : HashTable
i30 : fVec = fVector ⋈
o30 = {1, 5, 6, 2}
o30 : List
i31 : hVec = for j from 0 to d list
          sum(j+1, k -> (-1)^(j-k) * binomial(d+1-k, j-k) * fVec#k)
```

```
o31 = {1, 2, -1}

o31 : List

i32 : hilbertSeries(S^1/monomialIdeal ⋈, Reduce => true)

                  2
      1 + 2T - T
o32 = -----------
                3
        (1 - T)

o32 : Expression of class Divide
```

单项式理想的自由分解单项式理想的自由分解单项式理想的自由分解。正如 David Bayer、Irena Peeva 和 Bernd Sturmfels [BPS]所揭示的，单项式理想的极小自由分解通常由一个单纯复形编码。考虑多项式环 $R := \mathbb{Q}[y_1, y_2, \ldots, y_m]$中的单项式理想 J。假设 R配备了由 $\deg(y_i) = \mathbf{e}_i$给出的 $\mathbb{N}^m$-分次，其中每个 $1 \leqslant i \leqslant m$都如此，并且 $\mathbf{e}_1, \mathbf{e}_2$、…、是标准基。设 Δ 是一个单纯复形，其顶点以 J的生成元为标记。我们用其顶点的最小公倍数 $y^{\mathbf{a}_F} \in R$ 标记 Δ 的每个面 F；空面标记为单项式 $1 = y^{\mathbf{a}_\varnothing}$。由带标记的单纯复形 Δ 支撑的链复形 $C(\Delta)$，是由面对应的基构成的自由 $\mathbb{N}^m$-分次 R-模的链复形。更准确地说，链复形 $C(\Delta)$ 由以下数据确定

$$C_i(\Delta) := \bigoplus_{\dim(F)=i-1} R(-\mathbf{a}_F) \qquad \text{and} \qquad \partial(F) = \sum_{\dim(G)=\dim(F)-1} \operatorname{sign}(G,F)\, y^{\mathbf{a}_F - \mathbf{a}_G}\, G\,.$$

符号 F和 G 既表示 Δ 中的面，也表示$C(\Delta)$的底层自由模中的基向量。对 (G,F)所对应的符号属于 $\{-1,0,1\}$，并且是 Δ的链复形的边界映射中的数据。更多信息请参见 [MS] 中的第4.1小节，或 [P]中的第55章。

我们通过一个具体示例来说明这一构造。考虑图 Γ 中的单纯复形 2 以及 $J = (y_0y_1, y_0y_2, y_0y_3, y_1y_2y_3)$ 中的单项式理想 $R = \mathbb{Q}[y_0, y_1, y_2, y_3]$。将



图2。左侧是 Γ，右侧是其顶点的标记

将 Γ 的顶点按 J 的生成元标记为：$x_0 \mapsto y_0y_1$、$x_1 \mapsto y_0y_2$、$x_2 \mapsto y_0y_3$ 和 $x_3 \mapsto y_1y_2y_3$。

```
i33 : x = getSymbol "x"; S = ZZ[x_0..x_3];
i35 : Δ = simplicialComplex{x_0*x_1*x_2, x_2*x_3}
o35 = simplicialComplex | x_2x_3 x_0x_1x_2 |
o35 : SimplicialComplexi36 : chainComplex Δ1 4 4 1
o36 = ZZ  <-- ZZ  <-- ZZ  <-- ZZ
-1      0       1       2

o36 : ChainComplex

i37 : y = getSymbol "y"; R = QQ[y_0..y_3, DegreeRank => 4];
```

```
i39 : J = ideal(y_0*y_1, y_0*y_2, y_0*y_3, y_1*y_2*y_3)
o39 = ideal (y y , y y , y y , y y y )
              0 1  0 2  0 3  1 2 3

o39 : Ideal of R

i40 : C = chainComplex(Δ, Labels => J_*)
       1      4      4      1
o40 = R  <-- R  <-- R  <-- R

      0      1      2      3

o40 : ChainComplex

i41 : C.dd
           1                                          4
o41 = 0 : R  <---------------------------------------- R  : 1
              | y_0y_1 y_0y_2 y_0y_3 y_1y_2y_3 |
           4                                            4
      1 : R  <-------------------------------------------- R  : 2
              {1, 1, 0, 0} | -y_2 -y_3 0    0       |
              {1, 0, 1, 0} | y_1  0    -y_3 0       |
              {1, 0, 0, 1} | 0    y_1  y_2  -y_1y_2 |
              {0, 1, 1, 1} | 0    0    0    y_0     |
           4                              1
      2 : R  <------------------------- R  : 3
              {1, 1, 1, 0} | y_3  |
              {1, 1, 0, 1} | -y_2 |
              {1, 0, 1, 1} | y_1  |
              {1, 1, 1, 1} | 0    |

o41 : ChainComplexMap

i42 : assert(res(R^1/J) == C)
```

链复形 $C(\Delta)$ 取决于标记，并不总是一个分解。

```
i43 : C' = chainComplex(Δ, Labels => reverse J_*)
       1      4      4      1
o43 = R  <-- R  <-- R  <-- R

      0      1      2      3

o43 : ChainComplex

i44 : prune homology C'

o44 = 0 : cokernel | y_0y_3 y_0y_2 y_0y_1 y_1y_2y_3 |

      1 : cokernel {1, 1, 0, 1} | y_2 |

      2 : 0

      3 : 0

o44 : GradedModule
```

给定单项式理想 J，有几种算法可以返回一个带标记的单纯复形 Δ，使得链复形 $C(\Delta)$ 是 R/J的一个自由分解。下面介绍其中几种。

```
i45 : J' = monomialIdeal(y_1*y_3, y_2^2, y_0*y_2, y_1^2, y_0^2);

o45 : MonomialIdeal of R

i46 : T = taylorResolution J'
```

```
    1 5 10 10 5 1o46 = R  <-- R  <-- R   <-- R   <-- R  <-- R0 1 2 3 4 5
o46 : ChainComplexi47 : gensJ' = first entries mingens J'2 2 2
o47 = {y y , y , y y , y , y }1 3 2 02 1 0o47 : Listi48 : S = ZZ[x_0..x_4];
i49 : assert(T == chainComplex(simplexComplex(4, S), Labels => gensJ'))
i50 : assert(lyubeznikSimplicialComplex(J', S) === simplexComplex(4, S))
i51 : Γ = buchbergerSimplicialComplex(J',S)
o52 = simplicialComplex | x_0x_2x_3x_4 x_0x_1x_2x_3 |
o52 : SimplicialComplexi53 : B = buchbergerResolution J'1 5 9 7 2
o53 = R  <-- R  <-- R  <-- R  <-- R0 1 2 3 4o53 : ChainComplex
i54 : assert all(3, i -> HH_(i+1) B == 0)
i55 : assert(betti B == betti res J')
i56 : assert(B == chainComplex(Γ, Labels => first entries mingens J'))
i57 : assert(Γ === lyubeznikSimplicialComplex(J', S, MonomialOrder => {2,
                                                                    1,0,3,4}))
i59 : assert(Γ === scarfSimplicialComplex(J', S))
```

有关 Taylor 分解、Lyubeznik 分解和 Scarf 复形的更多信息，请参见 [Me]。Buchberger 分解的相关描述见 [OW]。

致谢。三位作者均获得了加拿大自然科学与工程研究理事会（NSERC）的部分资助。

参考文献

[BH] Winfried Bruns 和 Jürgen Herzog，《Cohen-Macaulay 环》，《数学进阶研究》（Cambridge Studies in Advanced Mathematics），第 39 卷，剑桥大学出版社，1993 年。[BPS] Dave Bayer、Irena Peeva 和 Bernd Sturmfels，《单项式分解》，《数学研究通讯》5（1998），第 1–2 期，31–46 页。[L] Frank H. Lutz，《The Manifold Page》（2017），http://page.math.tu-berlin.de/~lutz/stellar/。[M2] Daniel R. Grayson 和 Michael E. Stillman，Macaulay2，一款用于代数几何研究的软件系统，参见 http://www.math.uiuc.edu/Macaulay2/。[Me] Jeff Mermin，《三个单纯分解》，《交换代数进展》1，德古意特出版社，2012 年，第 127–141 页。[MS] Ezra Miller 和 Bernd Sturmfels，《组合交换代数》，《数学研究生教材》，第 227 卷，施普林格出版社，2005 年。[Mu] James R. Munkres，《代数拓扑基础》，艾迪生-韦斯利出版社，1984 年。

[OW] Anda Olteanu 和 Volkmar Welker，布赫伯格分解，《交换代数杂志》**8**（2016），第 4 期，571–587 页。[P] Irena Peeva，《分次 *syzygy*》，《代数与应用》，第 14 卷，施普林格出版社，2011 年。[S] Richard P. Stanley，《组合学与交换代数》，第 2 版，《数学进展》，第 41 卷，比克豪泽出版社，1996 年。

康考迪亚大学数学与统计系，蒙特利尔，QB，H3G 1M8，加拿大benjamin.hersey@concordia.ca。

女王大学数学与统计系，金斯顿，ON，K7L 3N6，加拿大ggsmith@mast.queensu.ca， 18az45@queensu.ca。