

# Exact direct-space asymmetric units for the 230 crystallographic space groups

Ralf W. Grosse-Kunstleve,<sup>a\*</sup> Buddy Wong,<sup>b</sup> Marat Mustyakimov<sup>c</sup> and Paul D. Adams<sup>a</sup>

<sup>a</sup>Physical Biosciences Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, BLDG 64R0121, Berkeley, California, 94720-8118, USA, <sup>b</sup>UC Leads Summer Research Program, University of California, Berkeley, California, 94720, USA, and <sup>c</sup>Bioscience Division, Los Alamos National Laboratory, B-8, MS M888, Los Alamos, New Mexico, 87545, USA. Correspondence e-mail: [rwgrosse-kunstleve@lbl.gov](mailto:rwgrosse-kunstleve@lbl.gov)

It is well known that the direct-space asymmetric unit definitions found in the *International Tables for Crystallography*, Volume A, are inexact at the borders. Face- and edge-specific sub-conditions have to be added to remove parts redundant under symmetry. This paper introduces a concise geometric notation for asymmetric unit conditions. The notation is the foundation for a reference table of exact direct-space asymmetric unit definitions for the 230 crystallographic space-group types. The change-of-basis transformation law for the conditions is derived, which allows the information from the reference table to be used for any space-group setting. We also show how the vertices of an asymmetric unit can easily be computed from the information in the reference table.

## 1. Introduction

In the presence of symmetry, the concept of an *asymmetric unit* (also known as *fundamental region* in mathematics) is important for many practical applications, for example to avoid time-consuming redundant calculations or to suppress redundant output. *International Tables for Crystallography*, Volume A (ITA) (Hahn, 2005) defines the direct-space asymmetric unit of a crystallographic space group (DAU) as ‘the smallest part of space from which, by application of all symmetry operations of the space group, the whole of space is filled exactly’ (ITA §2.2.8). This paper focuses on definitions of exact DAUs. These are refinements of the ITA conditions, which are sets of inequalities for each space group that must be true simultaneously for a point with fractional coordinates  $x, y, z$  to be inside the DAU, for example  $0 \leq x \leq 1$ ;  $0 \leq y \leq 1$ ;  $0 \leq z \leq 1/2$  for space group *P2*. The ITA conditions define the DAU shapes but are inexact for the borders. For example, the ITA *P2* conditions are true for all eight vertices of the DAU parallelepiped, but according to the definition in ITA §2.2.8 only two of these points can be in the exact asymmetric unit. To make a DAU exact, sub-conditions have to be added to the shape conditions, specific to faces and edges. Koch & Fischer (1974) published exact DAU definitions for the cubic space groups. In Grosse-Kunstleve *et al.* (2003) we presented an overview of an online gallery of exact DAUs for all 230 crystallographic space groups. Chapter 1.5 of Shmueli (2008) and the KVEC server at <http://www.cryst.ehu.es> also offer exact DAU definitions, but the DAU shapes are partially incompatible with those of ITA. In this paper we

introduce a concise geometric notation which is the foundation for a reference table of exact DAUs, using the same definitions as in our previous work, which are fully compatible with those of ITA.

## 2. Geometric cut notation and expressions

This section defines a concise geometric notation that has greatly accelerated the progress of this work. As will become apparent below, the notation enables a systematic, intuitive labelling of planes that define an exact DAU.

Similar to the ITA approach, a DAU shape is defined by a list of inequalities. We work with the general form

$$hx + ky + lz + c \geq 0 \quad (1)$$

or

$$hx + ky + lz + c > 0. \quad (2)$$

$h, k, l$  are Miller indices that define the normal vector of a plane,  $c$  is a scalar constant which determines the distance of the plane from the origin, and  $x, y, z$  are fractional coordinates in direct space. We call both equations a *cut* since the geometric interpretation is a division of direct space into two halves. The left-hand side of the equations is exactly zero for points inside the cut plane. The inequalities are defined to be true for points  $x, y, z$  inside the DAU. Equation (1) is used if a region of the cut plane is inside the DAU and equation (2) is used if the entire plane is outside. To facilitate a concise representation of DAU definitions, we introduce a *cut notation*. The general form is

$$\text{cut}((h, k, l), c) \tag{3}$$

or

$$+\text{cut}((h, k, l), c) \tag{4}$$

corresponding to equations (1) and (2), respectively. To obtain intuitive labels for DAU cut planes, we use the *geometric*

**Table 1**

Definition of geometric cut symbols used in Table 2.

See Fig. 1 for geometric illustrations of the main symbols.

Symbol	$h, k, l$	$c$	Expression
$x_1$	(-1, 0, 0)	1	
$x_0$			$-x_1 \cdot 0$
$x_2$			$x_1/2$
$x_3$			$x_1/3$
$x_4$			$x_1/4$
$x_8$			$x_1/8$
$x_{34}$			$x_1 \cdot 3/4$
$y_1$	(0, -1, 0)	1	
$y_0$			$-y_1 \cdot 0$
$y_2$			$y_1/2$
$y_3$			$y_1/3$
$y_4$			$y_1/4$
$y_8$			$y_1/8$
$z_1$	(0, 0, -1)	1	
$z_0$			$-z_1 \cdot 0$
$z_2$			$z_1/2$
$z_3$			$z_1/3$
$z_4$			$z_1/4$
$z_6$			$z_1/6$
$z_8$			$z_1/8$
$z_{12}$			$z_1/12$
$p_1$	(-1, 1, 0)	1	
$p_0$			$-p_1 \cdot 0$
$p_2$			$p_1/2$
$p_3$			$p_1/3$
$p_4$			$p_1/4$
$m_1$	(-1, -1, 0)	1	
$m_0$			$-m_1 \cdot 0$
$m_2$			$m_1/2$
$m_4$			$m_1/4$
$h_1$	(1, -2, 0)	1	
$h_0$			$-h_1 \cdot 0$
$k_1$	(-2, 1, 0)	1	
$k_0$			$-k_1 \cdot 0$
$xz_1$	(1, 0, 1)	1	
$xz_0$			$-xz_1 \cdot 0$
$xz_2$			$xz_1/2$
$xz_4$			$xz_1/4$
$zx_1$	(-1, 0, 1)	1	
$zx_0$			$-zx_1 \cdot 0$
$zx_2$			$zx_1/2$
$yz_1$	(0, 1, 1)	1	
$yz_0$			$-yz_1 \cdot 0$
$yz_2$			$yz_1/2$
$yz_4$			$yz_1/4$
$zy_1$	(0, -1, 1)	1	
$zy_0$			$-zy_1 \cdot 0$
$zy_2$			$zy_1/2$
$zy_4$			$zy_1/4$
$dy_8$	(1, -1, 1)	1/8	
$tx_0$	(-2, 1, 1)	0	
$ty_0$	(-1, 2, -1)	0	
$tz_2$	(-2, 1, -1)	1/2	

Primary geometric cut symbols are defined directly by a normal and constant. Secondary geometric cut symbols are defined by a cut expression based on a primary symbol. Note that the normal vector of all cuts with a '0' subscript in the geometric cut symbol is the opposite of the normal vector of the other cut planes in the same flock. This convention greatly reduces the number of minus signs in the exact DAU definitions of Table 2.

cut symbols defined in Table 1, for example  $tx_0 = \text{cut}((-2, 1, 1), 0)$ . Relationships between cuts can be formalized via cut expressions using unary and binary operators defined as follows:

$$-\text{cut}((h, k, l), c) \stackrel{\text{def}}{=} \text{cut}((-h, -k, -l), -c) \tag{5}$$

$$\sim \text{cut}((h, k, l), c) \stackrel{\text{def}}{=} \text{cut}((-h, -k, -l), c) \tag{6}$$

$$\text{cut}((h, k, l), c) \cdot s \stackrel{\text{def}}{=} \text{cut}((h, k, l), c \cdot s) \tag{7}$$

$$\text{cut}((h, k, l), c)/s \stackrel{\text{def}}{=} \text{cut}((h, k, l), c/s). \tag{8}$$

The variable  $s$  is a scalar value. Each of the operators defined in equations (5)–(8) has a simple geometric interpretation. The ‘-’ operator defined by equation (5) corresponds to a reversal of ‘inside’ and ‘outside’. The ‘~’ operator defined by equation (6) acts like a centre of inversion at the origin; see Figs. 1(b) and 1(c) for an example. The multiplication and division operators defined by equations (7) and (8) provide a notation for parallel shifts, as highlighted by Fig. 1(a).

The DAU conditions of ITA have a straightforward correspondence to our cut definition. We call the ITA conditions *shape cuts*. We employ the concept of context to avoid redundancy in the definition of sub-conditions specific to a given DAU face by appending the sub-conditions to the corresponding shape cut, surrounded by parentheses; such a cut is a *face cut*. Similarly, sub-conditions specific to a given edge are appended to a corresponding face cut, again surrounded by parentheses, and are called *edge cuts*. In some cases, the DAU choices of ITA necessitate the combination of cuts via logical conjunction or disjunction. Following common practice, we chose the symbol ‘&’ for conjunction and ‘|’ for disjunction. To give an example, the cut expression

$$x_0(z_4 \ \& \ z_0(-y_0)) \tag{9}$$

appears for space group  $P\bar{4}2c$  (No. 112), using the geometric cut notation of Table 1. Here  $x_0$  is a shape cut. The expression in the outer pair of parentheses is a face cut, composed of the logical conjunction  $z_4$  &  $z_0$ . The expression in the inner pair of parentheses ( $-y_0$ ) is an edge cut. As an example, a full step-by-step interpretation of equation (9) is shown in Appendix A.

The symbols in Table 1 include seven main flocks of parallel planes:  $x_d, y_d, z_d, p_d, m_d, h_d, k_d$ . The position of a cut plane relative to the origin of the coordinate system is indicated with the index  $d = 1/c$ , with  $c$  as defined in equations (3) and (4), except if  $c = 0$  or  $c = 3/4$ . Fig. 1 illustrates the geometric interpretation of the main geometric cut symbols. A large majority of the cut planes needed in the DAU definitions presented below can be labelled intuitively with these symbols. The remaining symbols in Table 1 were introduced primarily to condense the DAU definitions in Table 2 below.

### 3. Methods

#### 3.1. Change-of-basis transformation law

In many situations it is essential to be able to transform variables from one basis system to another. Giacovazzo (1992) includes a table of transformation laws (Table 2.E.1) for commonly used variables, for example fractional coordinates, Miller indices or anisotropic displacement parameters. This list can be extended by a transformation law for DAU definitions based on equations (1) and (2). Borrowing the conventions of ITA, let  $(\mathbf{P}, \mathbf{p})$  be a change-of-basis matrix with a  $(3 \times 3)$  rotation part  $\mathbf{P}$  and  $\mathbb{R}^3$  translation vector  $\mathbf{p}$ , and let  $(\mathbf{Q}, \mathbf{q})$  be its inverse. A column vector of fractional coordinates  $\mathbf{x} = (x, y, z)^T$  in a first  $\mathbb{R}^3$  basis system is transformed to coordinates  $\mathbf{x}'$  in a second basis system *via*

$$\mathbf{x}' = \mathbf{Q}\mathbf{x} + \mathbf{q}. \quad (10)$$

We also define the row vector  $\mathbf{n} = (h, k, l)$  in the first basis system. The corresponding  $\mathbf{n}'$  in the second basis system is given by

$$\mathbf{n}' = \mathbf{n}\mathbf{P}. \quad (11)$$

The determination of the scalar constant  $c'$  is based on the rationale that

$$\mathbf{n}' \cdot \mathbf{x}' + c' = 0 \quad (12)$$

must hold for all solutions  $\mathbf{x}$  of

$$\mathbf{n} \cdot \mathbf{x} + c = 0. \quad (13)$$

Setting  $\mathbf{n}' \cdot \mathbf{x}' + c' = \mathbf{n} \cdot \mathbf{x} + c$ , substituting equations (10) and (11), and solving for  $c'$  yields the second part of the transformation law for DAU cuts:

$$c' = c - \mathbf{n}' \cdot \mathbf{q}. \quad (14)$$

#### 3.2. Determination of vertices

Given a list of shape cuts, the DAU vertices can be computed by solving equation (13) for all unique ordered triplets of cuts. Let  $\mathbf{n}_1, \mathbf{n}_2$  and  $\mathbf{n}_3$  be the cut normal vectors of such a triplet. The three cut planes intersect in a point  $\mathbf{v}$  if the determinant of

$$\mathbf{N} = \begin{pmatrix} h_1 & k_1 & l_1 \\ h_2 & k_2 & l_2 \\ h_3 & k_3 & l_3 \end{pmatrix} \quad (15)$$

is not zero. Under this condition the point  $\mathbf{v}$  is found by solving  $\mathbf{N} \cdot \mathbf{v} + (c_1, c_2, c_3)^T = 0$ :

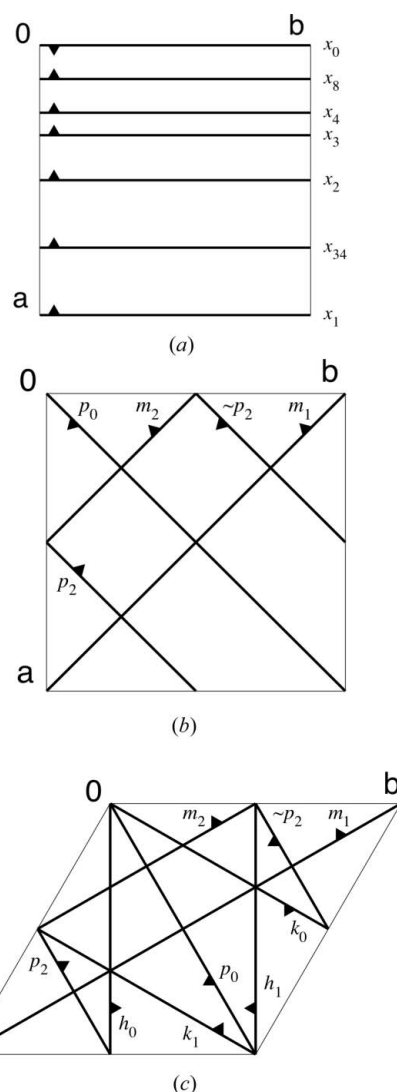
$$\mathbf{v} = -\mathbf{N}^{-1} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}. \quad (16)$$

$\mathbf{v}$  is a vertex of the DAU if all inequalities given by the shape cuts are also simultaneously true. If more than three planes intersect in a given vertex it is obtained multiple times and duplicates are discarded. We note that the largest number of shape cuts using the ITA definitions is nine, for space group

$Ia\bar{3}d$  (No. 230). In this case the determinant of  $\mathbf{N}$  is evaluated 84 times, equation (16) is evaluated 56 times and the final number of unique vertices is nine, in accordance with ITA.

#### 3.3. Validation of exact conditions

The exact conditions shown in Table 2 are validated with a sampling procedure to establish that the DAU is neither too small nor too large. The procedure is intentionally unsophisticated to maximize robustness. It is intrinsically highly inefficient, which is compounded by the use of a dynamically typed scripting language for its implementation. Nonetheless,



**Figure 1** Illustrations of selected cut planes with geometric cut symbols  $x_d, p_d, m_d, h_d$  and  $k_d$ . The  $(x, y, 0)$  plane of a unit cell is outlined with thin lines. The unit-cell origin is labelled with  $\mathbf{0}$ , the basis vectors with  $\mathbf{a}$  and  $\mathbf{b}$ . The traces of cut planes are indicated with thick lines; small attached triangles indicate the ‘inside’ direction. (a) Geometric interpretation of equations (7) and (8) using the  $x_d$  flock as an example. (b) Cut planes in the  $p_d$  and  $m_d$  flocks as they appear in a tetragonal system. The  $p_2$  and  $\sim p_2$  cut planes illustrate the geometric interpretation of equation (6). (c) Cut planes in the  $p_d, m_d, h_d$  and  $k_d$  flocks as they appear in a trigonal or hexagonal system.

**Table 2**

Exact DAU definitions for the 230 space groups, based on the geometric cut definitions of Table 1.

Following *International Tables*, Vol. B (Shmueli, 2001) Table A1.4.2.7, the first column, *n:c*, lists the space-group numbers and setting codes separated by a colon. For monoclinic space groups, the setting code 'b' indicates 'unique axis b'; 'b1' indicates 'unique axis b, cell choice 1'. For orthorhombic, tetragonal and cubic space groups, '2' indicates 'origin choice 2'. For rhombohedral space groups, 'h' indicates 'hexagonal axes'. The conditions for space groups 78, 95, 145, 154, 170, 172, 181 and 213 are defined by change-of-basis operations transforming the conditions of the enantiomorphic mates.

<i>n:c</i>	Cuts
1	$x_0; +x_1; y_0; +y_1; z_0; +z_1$
2	$x_0(y_0(z_2) \& y_2(z_2)); x_2(y_0(z_2) \& y_2(z_2)); y_0; +y_1; z_0; +z_1$
3:b	$x_0; +x_1; y_0; +y_1; z_0(x_2); z_2(x_2)$
4:b	$x_0; +x_1; y_0; +y_1; z_0(x_0(+y_2) \& x_2(+y_2)); z_2(x_0(+y_2) \& x_2(+y_2))$
5:b1	$x_0(z_2); x_2(z_2); y_0; +y_2; z_0; +z_1$
6:b	$x_0; +x_1; y_0; y_2; z_0; +z_1$
7:b1	$x_0; +x_1; y_0(+z_2); y_2(+z_2); z_0; +z_1$
8:b1	$x_0; +x_1; y_0; y_4(+x_2); z_0; +z_1$
9:b1	$x_0; +x_1; y_0(+z_2); y_4(+z_2); z_0; +z_1$
10:b	$x_0(z_2); x_2(z_2); y_0; y_2; z_0; +z_1$
11:b	$x_0; +x_1; y_0(z_0(x_2) \& z_2(x_2)); y_4; z_0; +z_1$
12:b1	$x_0(z_2); x_2(z_2); y_0; y_4(x_4(z_2)); z_0; +z_1$
13:b1	$x_0(z_0(y_2) \& z_4); x_2(z_0(y_2) \& z_4); y_0; +y_1; z_0; +z_2$
14:b1	$x_0(y_0(z_2)); +x_1; y_0(x_2(z_2)); y_4(+z_2); z_0; +z_1$
15:b1	$x_0(z_4); x_2(z_4); y_0; +y_2; z_0(y_4(x_4)); z_2(-y_4(x_4))$
16	$x_0(z_2); x_2(z_2); y_0(z_2); y_2(z_2); z_0; +z_1$
17	$x_0(-z_4 \& z_1 \cdot 3/4); x_2(-z_4 \& z_1 \cdot 3/4); y_0(z_2); y_2(z_2); z_0; +z_1$
18	$x_0; x_2(-y_0); y_0; +y_2; z_0; +z_1$
19	$x_0; +x_2; y_0(-z_2); y_2(z_2); z_0(+y_2); +z_1$
20	$x_0(z_4); x_2(-z_4); y_0; y_2(-z_0); z_0; +z_2$
21	$x_0(z_2); x_4(y_4); y_0(z_2); y_2(z_2); z_0; +z_1$
22	$x_0(z_2); x_4(-z_4 \& z_1 \cdot 3/4); y_0(z_2); y_4(-z_4 \& z_1 \cdot 3/4); z_0; +z_1$
23	$x_0; x_2(-y_0); y_0; y_2(-z_0); z_0; z_2(-x_0)$
24	$x_0(y_4); x_2(y_4); y_0(z_4); y_2(z_4); z_0(x_4); z_2(x_4)$
25	$x_0; x_2; y_0; y_2; z_0; +z_1$
26	$x_0; x_2; y_0(+z_2); y_2(+z_2); z_0; +z_1$
27	$x_0(+z_2); x_2(+z_2); y_0(+z_2); y_2(+z_2); z_0; +z_1$
28	$x_0(y_2); x_4; y_0; +y_1; z_0; +z_1$
29	$x_0(+z_2); x_4(+z_2); y_0; +y_1; z_0; +z_1$
30	$x_0(y_2); x_2(y_2); y_0; +y_1; z_0; +z_2$
31	$x_0; x_2; y_0(+z_2); y_2(+z_2); z_0; +z_1$
32	$x_0; x_2(-y_0); y_0; +y_2; z_0; +z_1$
33	$x_0; +x_2; y_0(+z_2); y_2(+z_2); z_0; +z_1$
34	$x_0; x_2(-y_0); y_0; +y_2; z_0; +z_1$
35	$x_0; x_4(y_4); y_0; y_2; z_0; +z_1$
36	$x_0; x_2; y_0; +y_2; z_0; +z_2$
37	$x_0(+z_2); x_4(y_4); y_0(+z_2); y_2(+z_2); z_0; +z_1$
38	$x_0; x_2; y_0; y_2; z_0; +z_2$
39	$x_0(+z_2); x_2(+z_2); y_0(+z_2); y_4; z_0; +z_1$
40	$x_0(+z_2); x_4; y_0(+z_2); y_2(+z_2); z_0; +z_1$
41	$x_0; x_2(-y_0); y_0; +y_2; z_0; +z_2$
42	$x_0; x_4(+z_2); y_0; y_4(+z_2); z_0; +z_1$
43	$x_0; x_4(-y_0(+z_2)); y_0; +y_4; z_0; +z_1$
44	$x_0; x_2; y_0; y_2; z_0; +z_2$
45	$x_0; x_2(-y_0); y_0; +y_2; z_0; +z_2$
46	$x_0(y_2); x_4; y_0; +y_1; z_0; +z_2$
47	$x_0; x_2; y_0; y_2; z_0; z_2$
48:2	$x_0(-y_0(z_2)); x_4(-z_4 \& z_1 \cdot 3/4); \sim y_4(-z_4 \& z_1 \cdot 3/4); y_4(-z_4 \& z_1 \cdot 3/4); z_0; +z_1$
49	$x_0(z_4); x_2(z_4); y_0(z_4); y_2(z_4); z_0; z_2$
50:2	$x_0(-y_2); x_4(-y_4 \& y_1 \cdot 3/4); y_0; +y_1; z_0(-y_4 \& y_1 \cdot 3/4); z_2(-y_4 \& y_1 \cdot 3/4)$
51	$x_0(z_2); x_4; y_0; y_2; z_0; +z_1$
52	$x_0; +x_1; y_0(-x_4 \& x_{34}); y_4(z_4); z_0(-x_2); z_2(-x_2)$
53	$x_0; x_2; y_0; +y_1; z_0(y_2); z_4(x_4)$
54	$x_0(-z_4); x_2(z_4); y_0(-x_4); y_2(-x_4); z_0; +z_2$
55	$x_0; x_2(-y_0); y_0; +y_2; z_0; z_2$
56	$x_0(y_2(-z_0)); x_4(-y_4 \& y_1 \cdot 3/4); y_0; +y_1; z_0; +z_2$
57	$x_0(-y_2); x_2(-y_2); y_0; +y_1; z_0(-y_4 \& y_1 \cdot 3/4); z_4$
58	$x_0; x_2(-y_0); y_0; +y_2; z_0; z_2$

**Table 2 (continued)**

<i>n:c</i>	Cuts
59:2	$x_0(-y_0(z_2)); x_4; \sim y_4; y_4; z_0; +z_1$
60	$x_0(z_4); x_2(-z_4); y_0; y_2(-x_0(-z_0)); z_0; +z_2$
61	$x_0; x_2(-y_0(-z_0)); y_0; +y_2; z_0; +z_2$
62	$x_0; x_2(-y_0(-z_0)); y_0(+z_2); y_4; z_0; +z_1$
63	$x_0; x_2; y_0; +y_2; z_0(y_4(x_4)); z_4$
64	$x_0; x_4(z_4); y_0; +y_2; z_0(y_4); z_2(+y_4)$
65	$x_0; x_4(y_4); y_0; y_2; z_0; z_2$
66	$x_0(z_4); x_4(y_4); y_0(z_4); y_2(z_4); z_0; z_2$
67	$x_0; x_2; y_0(x_4); y_4; z_0(x_4); z_2(x_4)$
68:2	$x_0(z_4); x_2(z_4); y_0(x_4); y_4(z_4); z_0(+x_2 \& y_4(x_4)); +z_2$
69	$x_0; x_4(z_4); y_0; y_4(z_4); z_0; z_2$
70:2	$x_0(-y_0(z_2)); x_8(-z_8 \& z_1 \cdot 5/8); \sim y_8(-z_1 \cdot 3/8 \& z_1 \cdot 7/8); y_8(-z_8 \& z_1 \cdot 5/8); z_0; +z_1$
71	$x_0; x_4(y_4(z_4)); y_0; y_2; z_0; z_2$
72	$x_0(z_4); x_4(y_4(z_4)); y_0(z_4); y_2(z_4); z_0; z_2$
73	$x_0(y_4); x_4(z_4(y_4)); y_0(z_4); y_2(-z_4); z_0; +z_2$
74	$x_0; x_4(-z_4 \& z_1 \cdot 3/4); y_0(z_2); y_4; z_0; +z_1$
75	$x_0(-y_0); x_2; y_0; y_2(-x_2); z_0; +z_1$
76	$x_0(+z_4); x_2(+z_4); y_0(+z_1 \cdot 3/4); y_2(+z_1 \cdot 3/4); z_0; +z_1$
77	$x_0(+z_2); x_2(+z_2); y_0(+z_2); y_2(+z_2); z_0; +z_1$
78	$76 \rightarrow a, b, -c + 1$
79	$x_0(-y_0); x_2; y_0; y_2(-x_2); z_0; +z_2$
80	$x_0(y_2); x_2(y_2); y_0; +y_1; z_0; +z_4$
81	$x_0(-y_0(z_2)); x_2; y_0; y_2(-x_2(z_2)); z_0; +z_1$
82	$x_0(z_0(-y_0)); x_2(-y_0(z_4)); y_0; y_2(-x_0(z_4)); z_0; z_2(-y_0)$
83	$x_0(-y_0); x_2; y_0; y_2(-x_2); z_0; z_2$
84	$x_0(-y_0(z_4)); x_2; y_0; y_2(-x_2(z_4)); z_0; z_2$
85:2	$\sim x_4(-\sim y_4); x_4(z_0(-\sim y_4) \& z_2(-\sim y_4)); \sim y_4; y_4(-x_4); z_0(-y_0(-x_0)); z_2(-y_0(-x_0))$
86:2	$\sim x_4(-\sim y_4(z_4)); x_4(z_0(-\sim y_4) \& z_2(+\sim y_4)); \sim y_4; y_4(-x_4(z_4)); z_0(-y_0(-x_0)); z_2(-y_0(-x_0))$
87	$x_0(-y_0); x_2; y_0; y_2(-x_2); z_0; z_4(y_4(x_4) \& x_2(-y_0))$
88:2	$x_0; x_4; y_0(-x_0(z_2)   -x_4(+z_4)); y_4(-x_0(-z_8 \& z_1 \cdot 5/8)); z_0; +z_1$
89	$x_0(p_0); x_2; y_0; y_2(-x_2); z_0(p_0); z_2(p_0)$
90	$x_0; x_2(-y_0); y_0; y_2(-x_0); z_0(p_0); z_2(p_0)$
91	$x_0(z_8(-y_0)); +x_1; y_0; +y_1; z_0(x_2); z_8(m_1)$
92	$x_0; +x_1; y_0; +y_1; z_0(p_0); z_8(-y_2)$
93	$x_0(y_2); x_2(y_2); y_0; +y_1; z_0(y_2); z_4(-p_0 \& m_1)$
94	$x_0(-y_0); x_2(z_2(-y_2)); y_0(z_2(-x_0)); +y_2; z_0(p_0); z_2(p_0)$
95	$91 \rightarrow -a + 1, b, c$
96	$x_0; +x_1; y_0; +y_1; z_0(p_0); z_8(-x_2)$
97	$x_0(-y_0); x_2; y_0; y_2(-x_2); z_0(p_0); z_4(m_2)$
98	$x_0(y_2); x_2(y_2); y_0; +y_1; z_0(m_1 \& -p_0); z_8(-y_4 \& y_1 \cdot 3/4)$
99	$x_0; y_2; z_0; +z_1; -p_0$
100	$x_0(-y_0); y_0; z_0; +z_1; m_2$
101	$x_0(+z_2); y_2(+z_2); z_0; +z_1; -p_0$
102	$x_0(+z_2); y_2(+z_2); z_0; +z_1; -p_0$
103	$x_0(-y_0); x_2; y_0; y_2(-x_2); z_0; +z_2$
104	$x_0(-y_0); x_2; y_0; y_2(-x_2); z_0; +z_2$
105	$x_0; x_2; y_0; y_2; z_0; +z_2$
106	$x_0(-y_0); x_2; y_0; +y_2; z_0; +z_2$
107	$x_0; y_2; z_0; +z_2; -p_0$
108	$x_0(-y_0); y_0; z_0; +z_2; m_2$
109	$x_0; x_2; y_0; y_2; z_0; +z_4$
110	$x_0(-y_0); x_2; y_0; +y_2; z_0; +z_4$
111	$x_0(z_2); y_2(z_2); z_0; +z_1; -p_0$
112	$x_0(z_4 \& z_0(-y_0)); x_2(z_4); y_0(z_4); y_2(z_4 \& z_0(-x_2)); z_0; +z_2$
113	$x_0(-y_0(z_2)); y_0; z_0; +z_1; m_2$
114	$x_0(-y_0); x_2; y_0; y_2(-x_2(-z_0)); z_0; +z_2$
115	$x_0; x_2; y_0; y_2; z_0(p_0); z_2(p_0)$
116	$x_0(y_2); x_2(y_2 \& z_0(-y_2)); y_0; +y_1; z_0(y_2 \& y_0(-x_0)); z_4(m_1 \& -p_0)$
117	$x_0(z_0(-y_0) \& z_2(-y_0)); x_2(-y_0); y_0; +y_2; z_0(m_2); z_2(m_2)$
118	$x_0(y_2); x_2(y_2); y_0; +y_1; z_0(y_2(-x_2) \& x_0(-y_0)); z_4(\sim p_2 \& -m_2)$
119	$x_0; x_2; y_0; y_2; z_0(p_0); z_4(m_2)$
120	$x_0(z_0(-y_0)); x_2(-y_0); y_0; +y_2; z_0(m_2); z_4(p_0)$
121	$x_0; y_2(-x_0(z_4)); z_0; z_2(-x_0); -p_0$
122	$x_0(y_2 \& z_0(-y_0)); x_2(y_2); y_0; +y_1; z_0(y_2(-x_2)); z_8(-y_4 \& y_1 \cdot 3/4)$
123	$x_0; y_2; z_0; z_2; -p_0$
124	$x_0(-y_0); x_2; y_0; y_2(-x_2); z_0; z_4(p_0)$
125:2	$\sim x_4(-\sim y_4); \sim y_4; z_0(p_0); z_2(p_0); -m_0$
126:2	$\sim x_4(-\sim y_4); x_4; \sim y_4; y_4(-x_4); z_0(-y_0(-x_0) \& x_4(-\sim y_4)); z_4(p_0)$

Table 2 (continued)

<i>n:c</i>	Cuts
127	$x_0(-y_0); y_0; z_0; z_2; m_2$
128	$x_0(-y_0); x_2; y_0; y_2(-x_2); z_0; z_4(m_2)$
129:2	$\sim x_4; y_4; z_0(-m_0); z_2(-m_0); -p_0$
130:2	$\sim x_4(-\sim y_4); x_4(z_0(-\sim y_4)); \sim y_4; y_4(-x_4); z_0(-y_0(-x_0)); z_4(-m_0)$
131	$x_0; x_2; y_0; y_2; z_0; z_4(p_0)$
132	$x_0(z_4); y_2(z_4); z_0; z_2; -p_0$
133:2	$\sim x_4; x_4(-z_0   -\sim y_4); \sim y_4; +y_4; z_0(-y_0(-x_0)); z_4(p_0)$
134:2	$\sim x_4(z_4); \sim y_4(z_4); z_0(p_0); z_2(p_0); -m_0$
135	$x_0(z_4(-y_0)); x_2(-y_0); y_0; +y_2; z_0; z_4(m_2)$
136	$x_0(z_4); y_2(z_4(-x_0)); z_0; z_2; -p_0$
137:2	$\sim x_4; x_4; \sim y_4; y_4; z_0(-y_0(-x_0)); z_4(-m_0)$
138:2	$\sim x_4; y_4(-\sim x_4(z_4)); z_0(-m_0); z_2(-m_0 \& \sim x_4(-y_4)); -p_0$
139	$x_0; y_2; z_0; z_4(m_2); -p_0$
140	$x_0(-y_0); y_0; z_0; z_4(p_0); m_2$
141:2	$x_0; x_2; \sim y_4; y_4; z_0(-y_0); z_8(-p_4)$
142:2	$x_0(z_8(-\sim y_4) \& z_0(-y_0)); x_2(-\sim y_4); \sim y_4; +y_4; z_0(x_4); z_8(m_4)$
143	$x_0(-y_0); y_0; z_0; +z_1; k_1; m_1(-h_1   -k_1); h_1$
144	$x_0; +x_1; y_0; +y_1; z_0; +z_3$
145	$144 \rightarrow b, a, c$
146:h	$x_0(-y_0); y_0; z_0; +z_3; k_1; m_1(-h_1   -k_1); h_1$
147	$x_0(-y_0); y_0; z_0(p_0(-y_0)); z_2(p_0(-y_0)); k_1; m_1(-h_1   -k_1); h_1$
148:h	$x_0(-y_0); y_0; z_0(p_0(-y_0)); z_6(-h_0(x_3)   -k_0(-y_0   -m_1)); k_1; m_1(-h_1   -k_1); h_1$
149	$x_0(-y_0); y_0; z_0(-h_0   -k_0); z_2(-h_0   -k_0); k_1; m_1(-h_1   -k_1); h_1$
150	$x_0(-y_0); y_0; z_0(p_0); z_2(p_0); k_1; m_1(-h_1   -k_1); h_1$
151	$x_0; +x_1; y_0; +y_1; z_0(-h_0   -h_1); z_6(-k_0   -k_1)$
152	$x_0; +x_1; y_0; +y_1; z_0(-p_0); z_6(-p_0)$
153	$x_0; +x_1; y_0; +y_1; z_0(-h_0   -h_1); z_6(x_0(-y_0) \& m_1)$
154	$152 \rightarrow b, a, c$
155:h	$x_0(-y_0); y_0; z_0(p_0); z_6(x_3 \& \sim p_3); k_1; m_1(-h_1   -k_1); h_1$
156	$z_0; +z_1; h_0; m_1; k_0$
157	$y_0; z_0; +z_1; k_1; m_1(y_3); p_0$
158	$x_0(-y_0); y_0; z_0; +z_2; k_1; m_1(-h_1   -k_1); h_1$
159	$x_0(-y_0); y_0; z_0; +z_2; k_1; m_1(-h_1   -k_1); h_1$
160:h	$z_0; +z_3; h_0; m_1; k_0$
161:h	$x_0(-y_0); y_0; z_0; +z_6; k_1; m_1(-h_1   -k_1); h_1$
162	$y_0; z_0(-h_0); z_2(-h_0); k_1; m_1(y_3); p_0$
163	$x_0(-y_0); y_0; z_0(p_0(-y_0)); z_4(-h_0   -k_0); k_1; m_1(-h_1   -k_1); h_1$
164	$y_0(z_2); z_0; +z_1; k_1; -h_0$
165	$x_0(-y_0); y_0; z_0(p_0(-y_0)); z_4(p_0); k_1; m_1(-h_1   -k_1); h_1$
166:h	$z_0(p_0); z_6(x_3); h_0; m_1; k_0$
167:h	$x_0(-y_0); y_0; z_0(p_0(-y_0)); z_{12}(y_3 \& p_3); k_1; m_1(-h_1   -k_1); h_1$
168	$y_0; z_0; +z_1; k_1; m_1(y_3); p_0(-y_0)$
169	$x_0; +x_1; y_0; +y_1; z_0; +z_6$
170	$169 \rightarrow b, a, c$
171	$x_1(y_2); y_0(x_2); z_0; +z_3; p_0(y_2)$
172	$171 \rightarrow -b + 1, -a + 1, c$
173	$x_0(-y_0); y_0; z_0; +z_2; k_1; m_1(-h_1   -k_1); h_1$
174	$x_0(-y_0); y_0; z_0; z_2; k_1; m_1(-h_1   -k_1); h_1$
175	$y_0; z_0; z_2; k_1; m_1(y_3); p_0(-y_0)$
176	$x_0(-y_0); y_0; z_0(p_0(-y_0)); z_4; k_1; m_1(-h_1   -k_1); h_1$
177	$y_0; z_0(-h_0); z_2(-h_0); k_1; m_1(y_3); p_0(-y_0)$
178	$x_0; +x_1; y_0; +y_1; z_0(p_0); z_{12}(-h_0   -h_1)$
179	$x_0; +x_1; y_0; +y_1; z_0(p_0); z_{12}(m_1 \& x_0(-y_0))$
180	$x_1(y_2); y_0(x_2); z_0(k_1); z_6(-h_0); p_0(y_2)$
181	$180 \rightarrow b + 1, -a - b, -c + 1/6$
182	$x_0(-y_0); y_0; z_0(p_0); z_4(-h_0   -k_0); k_1; m_1(-h_1   -k_1); h_1$
183	$y_0; z_0; +z_1; k_1; -h_0$
184	$y_0; z_0; +z_2; k_1; m_1(y_3); p_0(-y_0)$
185	$y_0; z_0; +z_2; k_1; m_1(y_3); p_0$
186	$y_0(+z_2); z_0; +z_1; k_1; -h_0$
187	$z_0; z_2; h_0; m_1; k_0$
188	$x_0(-y_0); y_0; z_0(-h_0   -k_0); z_4; k_1; m_1(-h_1   -k_1); h_1$
189	$y_0; z_0; z_2; k_1; m_1(y_3); p_0$
190	$x_0(-y_0); y_0; z_0(p_0); z_4; k_1; m_1(-h_1   -k_1); h_1$
191	$y_0; z_0; z_2; k_1; -h_0$
192	$y_0; z_0; z_4(-h_0); k_1; m_1(y_3); p_0(-y_0)$
193	$y_0; z_0(-h_0); z_4; k_1; m_1(y_3); p_0$
194	$z_0(p_0); z_4; h_0; m_1; k_0$
195	$z_0(y_2 \& x_2); m_1(-y_2); z_{y_0}(-z_{x_0}); z_{x_0}$
196	$p_0(m_2); \sim xz_2(-zy_0); xz_2(yz_0); -yz_0; zy_0$
197	$z_0(x_2); p_0(-zy_0); +m_1; zy_0$

Table 2 (continued)

<i>n:c</i>	Cuts
198	$x_0(-y_0); x_2; y_2(+z_0 \& x_2(+z_2)); xz_2(m_2); z_{x_0}(p_0); -yz_0; zy_0$
199	$x_2(-y_4); y_2(-z_4); z_0(x_4); z_{x_0}(-zy_0(+x_2)); zy_0$
200	$x_2; y_2; z_0; z_{x_0}(-zy_0); zy_0$
201:2	$\sim z_4(x_4); p_0(-zy_0(-x_0)); m_2(-zy_0(x_2)); zy_0$
202	$z_0; p_0(x_4); \sim xz_2(-zy_0); zy_0$
203:2	$p_0(-zy_0(-x_0)); m_4(-zy_0   -yz_4(\sim z_4)); zy_0; yz_4$
204	$x_2; z_0; p_0(-zy_0(x_4)); zy_0$
205	$x_2(-z_0(-zy_0)); y_2(-zy_0); z_0; z_{x_0}(-zy_0); zy_0$
206	$z_0(x_4); z_{x_0}(-zy_0); \sim xz_2; zy_0; \sim yz_2(-z_{x_0})$
207	$z_0(x_2); p_0; m_1(-p_0); zy_0(x_2)$
208	$z_{x_0}(-zy_0); -xz_0(yz_0); xz_2(y_4); \sim xz_2(y_4); zy_0; -yz_0; zy_2(-x_4); \sim yz_2(-x_4)$
209	$p_0(z_0); m_2(z_0); zy_0; -yz_0(-zy_0)$
210	$y_8(-\sim xz_4); z_8(m_4); p_0(-z_{x_0}); m_2(-\sim xz_2); -yz_0(z_0); z_{x_0}; \sim xz_2$
211	$z_0(p_0); z_{x_0}(-zy_0); \sim xz_2(y_4); zy_0; \sim yz_2(-x_4)$
212	$xz_2; -yz_0(-z_{x_2}); \sim yz_2(tx_0); -tx_0(x_8); ty_0(y_8); tz_2(-x_1 \cdot 3/8)$
213	$212 \rightarrow -b, c + 1/2, a - 1/2$
214	$x_8(\sim yz_4); y_8(\sim xz_4); \sim y_8(-\sim z_{x_1}/4); -z_{x_0}(zy_0); -zy_0; \sim zy_4(-y_0); dy_8(\sim p_4)$
215	$z_0(x_2); p_0; m_1; zy_0$
216	$p_0; m_2; zy_0; -yz_0$
217	$x_2(-z_0(y_4)); z_0; p_0; zy_0$
218	$x_2(-z_0(y_4)); y_2(-z_0(x_4)); z_0; z_{x_0}(-zy_0); zy_0$
219	$p_0; m_2(-p_0(z_0)); zy_0; -yz_0(-p_0   -zy_0(x_4))$
220	$-x_4(-z_0(-y_1 \cdot 3/8)); x_2; -y_4(-x_2(-z_8)); y_2(-z_4); z_0; z_{x_0}(-zy_0); zy_0$
221	$x_2; z_0; p_0; zy_0$
222:2	$x_{34}(z_4(y_2)   -zy_0); -z_4; p_0(-zy_0(z_2)); zy_0$
223	$z_0; z_{x_0}(-zy_0); \sim xz_2(y_4); zy_0; \sim yz_2(x_4)$
224:2	$p_0; \sim xz_1(y_2); z_{x_2}(y_2); \sim \sim yz_2; zy_0$
225	$z_0; p_0; m_2; zy_0$
226	$z_0; p_0; m_2(-p_0); zy_0(x_4)$
227:2	$-y_0(-xz_0); p_0; m_4; yz_4; zy_0$
228:2	$-y_0(z_{x_1}/4); p_0(-zy_0); m_4; yz_4(-zy_0(x_8)); zy_0$
229	$z_0; p_0; \sim xz_2(y_4); zy_0$
230	$x_8(\sim zy_4 \& \sim yz_4); \sim x_8(y_0(-z_4)); y_8(-\sim xz_4); \sim y_8(\sim z_{x_1}/4); z_4(y_0); -z_{x_0}; -xz_0(-z_0); -zy_0(z_{x_0}); -yz_0$

given current computing hardware, the entire Table 2 can be re-validated in less than 2 min.

The first part of the validation procedure samples the DAU conditions using two grids over the unit cell, given a user-defined number of sampling points *N* per unit in fractional coordinate space. To simplify this presentation, without loss of generality, we assume that *N* is identical in all three dimensions. *N* is always chosen to be even. The first *ugrid* covers the unit cell from 0 to *N* - 1, corresponding to the range [0.0, 1.0] in fractional coordinate space. All *ugrid* points are initialized with zero. The second *rgrid* covers space more redundantly from -*N*/2 to *N*, corresponding to the range [-0.5, 1.0]. The vertex determination of §3.2 is used to assure that the DAU to be validated falls entirely into the *rgrid*. For each *rgrid* point, the inequalities defined by the DAU cuts are evaluated. A value of one is assigned if the point is inside the DAU (all inequalities are true) and zero otherwise. If the point is inside the DAU, the crystallographic unit translations, in the form of the modulus operation, are applied to the grid indices of the point to determine the symmetry-equivalent grid point in the *ugrid*, which is then also set to one. If it was set already, an error message reports that the point is redundant.

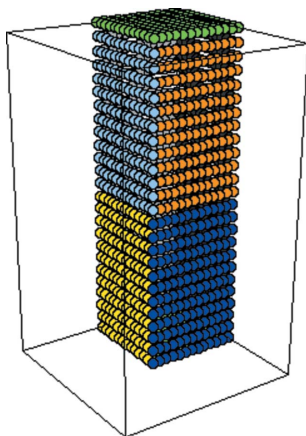
At the end of the first part of the validation procedure the *ugrid* has a value of one for all grid points inside the DAU and zero for all points outside; note that the *ugrid* has discon-

nected regions of grid points with value one if the DAU has points with negative coordinates. The second part of the validation procedure visits each point in the *ugrid*. The symmetry operations of the space group, taking the crystallographic unit translations and any centring translations into account, are applied to enumerate all equivalent points in the *ugrid*. If a point is flagged as inside the DAU, all equivalent points must be flagged as outside; otherwise an error message reports that the DAU has redundant points. For each point flagged as outside the DAU, one equivalent point must be flagged as inside; otherwise an error message reports that the point has no equivalent in the DAU.

If no error messages are shown, the validation procedure establishes conclusively that the DAU conditions have complete coverage and that the covered space is non-redundant under symmetry. The only critical parameter is the number  $N$  of sampling points per unit in fractional space. Based on an inspection of the locations of the symmetry elements, we found that  $N = 24$  is sufficiently large for all space groups. However, as a final validation we also ran the procedure for all space groups with  $N = 72$ , which takes about 45 min on a current 48-core system.

### 3.4. Visually assisted determination of exact conditions

The exact DAU conditions shown in Table 2 were determined manually. Progress was greatly accelerated by visual tools developed specifically for this purpose. A full presentation of these tools is beyond the scope of this paper [Grosse-Kunstleve *et al.* (2003) includes pointers to the openly available implementation]. The main idea is to colour-code pairs of redundant points on the DAU surface as they are detected in the sampling procedure described in §3.3; for example, the first point is coloured dark blue and the equivalent redundant point light blue. A very simple colour-selection procedure



**Figure 2**

Visualization of redundant pairs of points at the surface of the DAU of space group  $P4_2$  (No. 77). Colour-coded pairs of redundant points on the DAU surface are shown as they are detected in the sampling procedure described in §3.3. For example, the first point is coloured dark blue and the equivalent redundant point light blue. With the help of the colours it is immediately obvious how the redundant points are related and where to place the missing face cuts to obtain the exact DAU.

using only a small palette of colours was found to be sufficient in practice. An example is shown in Fig. 2. We added the face- or edge-specific cuts one at a time, updating the visualization after each step. In this way we could determine exact DAU definitions in a matter of a few minutes for most space groups.

## 4. Results

Table 2 defines exact DAUs for 230 *reference settings*, chosen to be compatible with the reference settings used in the IUCr symCIF dictionary (Brown, 2005). Using the change-of-basis transformation law of §3.1 in combination with the algorithms of Grosse-Kunstleve (1999), it is possible to automatically obtain an exact DAU for any setting.

Koch & Fischer (1974) and ITA §2.8 explain that the shape for a DAU is not uniquely determined and that the best choice is application specific. Similarly, the face- and edge-specific sub-conditions required for an exact DAU are also not uniquely determined. The choices we made for Table 2 aim at obtaining compact sets of sub-conditions, which is also expected to minimize the runtime needed for evaluating if a given point is inside the DAU. For the cubic space groups, we attempted to adopt the sub-conditions of Koch & Fischer (1974) but it turned out to be challenging in some cases. In six cases (space-group numbers 195, 198, 210, 220, 227, 228) the ITA shape conditions are incompatible with those of Koch & Fischer (1974). In some other cases their sub-conditions lead to complicated cut expressions. Using the approach of §3.4 it was only a small effort to determine simpler alternatives for Table 2.

For eight enantiomorphic space groups (the numbers are listed in the caption of Table 2) the exact DAU is defined through a change-of-basis transformation of the DAU of the enantiomorphic mate. The three remaining enantiomorphic pairs of space groups cannot be handled in this way because the ITA shape DAU conditions are pairwise incompatible. The change-of-basis matrices in Table 2 are expressed using the notation as defined in Zwart *et al.* (2008).

## 5. Conclusion

Table 2 is the first complete and uniform definition of exact DAUs for all 230 space-group types. The table is concise owing to the geometric cut notation introduced in this work. At the same time, the cut expressions lend themselves to automatic processing, with results as demonstrated already in Grosse-Kunstleve *et al.* (2003). In the meantime we have found other practical uses in the context of the *PHENIX* suite (Adams *et al.*, 2010), such as the search for interactions between pairs of atoms (Grosse-Kunstleve *et al.*, 2004) and a bulk-solvent-mask determination procedure.

In this work we have used a manual approach for the determination of the face- and edge-specific sub-conditions required for exact DAUs. We believe an algorithmic approach is possible but will require significantly more initial effort than our manual approach. The cut plane formalism presented here could serve as a basis for future automation work.

## APPENDIX A

## Example step-by-step interpretation of a geometric cut expression

Equation (9) in §2, which appears for space group  $P\bar{4}2c$  (No. 112) in Table 2, was shown as an example of a geometric cut expression:

$$x_0(z_4 \& z_0(-y_0)). \quad (17)$$

The shape cut in this example is  $x_0$ . According to Table 1 and employing equation (7), this geometric cut symbol expands to  $\text{cut}((1, 0, 0), 0)$ . Use of equation (1) yields  $1x + 0y + 0z \geq 0$ , which simplifies to  $x \geq 0$ . The outer pair of parentheses encloses a face cut expression that applies only if  $x = 0$  [see text following equation (8) in §2]. The first part of the face cut expression,  $z_4$ , translates to  $z \leq 1/4$  [lookup in Table 1, use of equations (8) and (1), and simplification]. The second part of the face cut expression,  $z_0$ , translates to  $z \geq 0$ . The logical conjunction symbol '&' (defined in §2) indicates that a point in the plane is in the DAU only if both  $z \geq 0$  and  $z \leq 1/4$ , with *and* in the Boolean sense. The inner pair of parentheses encloses an edge cut expression specific to the  $(0, y, 0)$  line defined by the conditions  $x = 0$  and  $z = 0$ . Lookup in Table 1, use of equations (7), (5) and (1), and simplification lead to the sub-condition  $y \leq 0$ . In combination with the  $y_0$  shape cut of space group  $P\bar{4}2c$ , this means that only the point  $(0, 0, 0)$  on the  $(0, y, 0)$  line is in the DAU.

Graphical illustrations of the DAU conditions are available at [http://cci.lbl.gov/asu\\_gallery/](http://cci.lbl.gov/asu_gallery/). An expanded DAU notation that is more similar to the notation of ITA is shown along with the graphical illustrations.

We thank Michael M. J. Treacy for sending us an electronic file with a table of the DAU shape conditions of ITA. We thank the anonymous referees for corrections and suggestions that have led us to an improved presentation. We gratefully acknowledge the financial support of NIH/NIGMS through grant Nos. 5P01GM063210 and 1R01GM071939. Our work was supported in part by supplemental funding from the American Recovery and Reinvestment Act (ARRA) to NIH/NIGMS grant No. P01GM063210 and by the US Department of Energy under contract Nos. DE-AC03-76SF00098 and DE-AC02-05CH11231.7237.

## References

- Adams, P. D. *et al.* (2010). *Acta Cryst.* **D66**, 213–221.  
 Brown, I. D. (2005). [ftp://ftp.iucr.org/cifdics/cif\\_sym\\_1.0.1.dic](ftp://ftp.iucr.org/cifdics/cif_sym_1.0.1.dic).  
 Giacovazzo, C. (1992). *Fundamentals of Crystallography*. IUCr/Oxford University Press.  
 Grosse-Kunstleve, R. W. (1999). *Acta Cryst.* **A55**, 383–395.  
 Grosse-Kunstleve, R. W., Afonine, P. V. & Adams, P. D. (2004). *Newsletter of the IUCr Commission on Crystallographic Computing*, **4**, 19–36.  
 Grosse-Kunstleve, R. W., Wong, B. & Adams, P. D. (2003). *Newsletter of the IUCr Commission on Crystallographic Computing*, **2**, 10–16.  
 Hahn, T. (2005). *International Tables for Crystallography*, Vol. A. Heidelberg: Springer.  
 Koch, E. & Fischer, W. (1974). *Acta Cryst.* **A30**, 490–496.  
 Shmueli, U. (2001). *International Tables for Crystallography*, Vol. B, 2nd ed. Dordrecht: Kluwer.  
 Shmueli, U. (2008). *International Tables for Crystallography*, Vol. B, 3rd ed. Heidelberg: Springer.  
 Zwart, P. H., Grosse-Kunstleve, R. W., Lebedev, A. A., Murshudov, G. N. & Adams, P. D. (2008). *Acta Cryst.* **D64**, 99–107.