Potential applications of formal methods in crystallography?

Saulius Gražulis

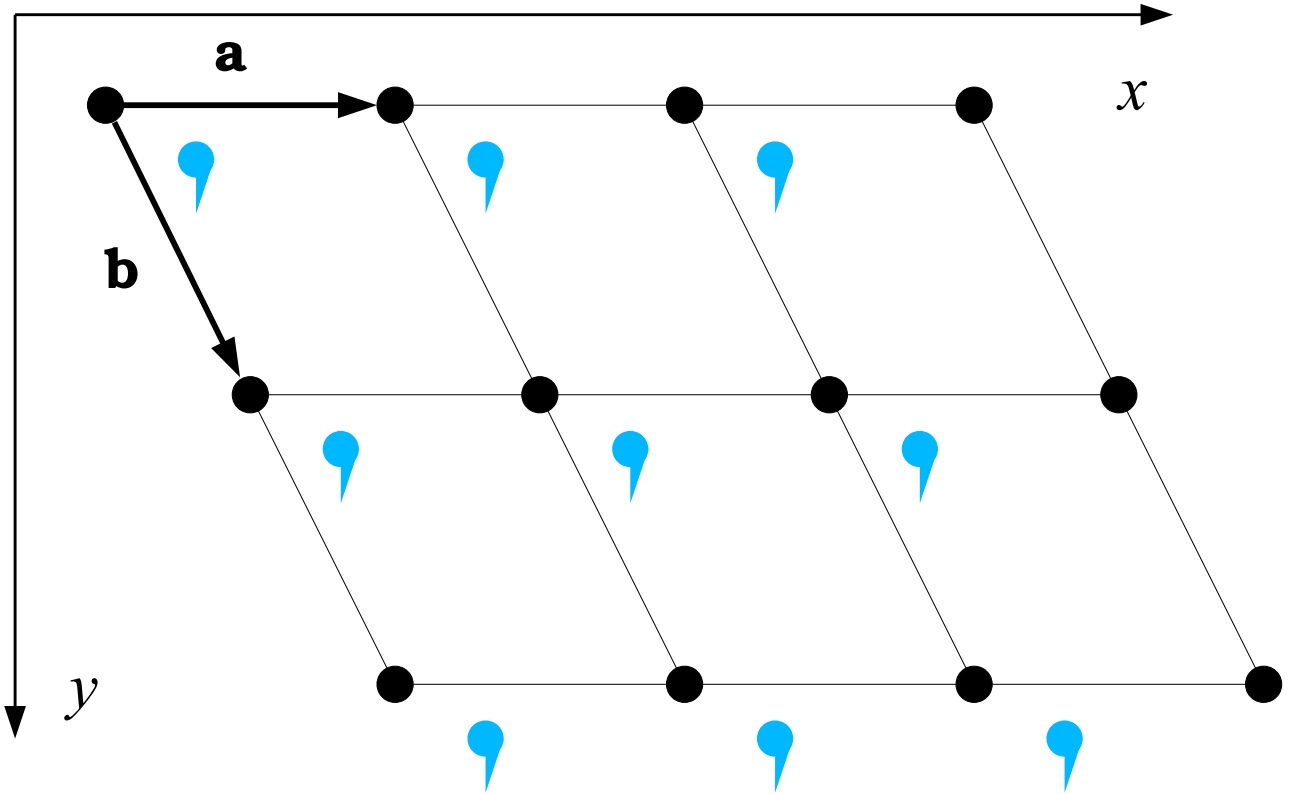
Vilnius, 2020

Vilnius University Institute of Biotechnology

Id: slides.tex 1523 2020-06-04 11:53:22Z saulius June 4, 2020

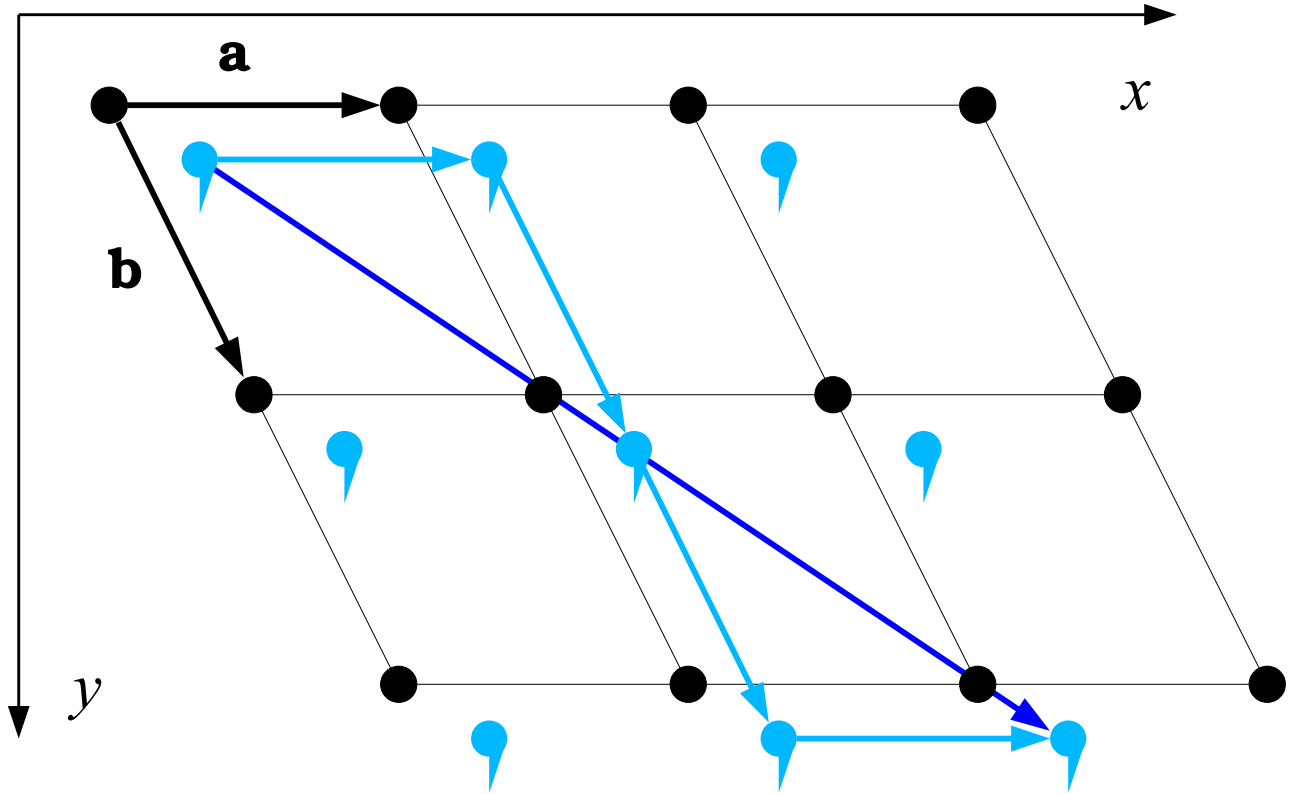
Crystall lattice

Periodiškumas ir postūmiai



Crystall lattice

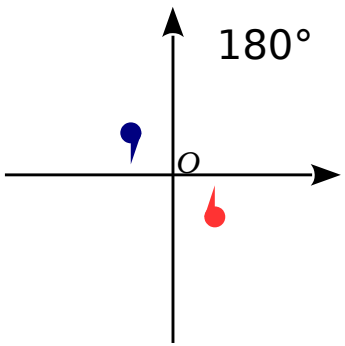
Postūmių grupė



Simetrijos operatoriaus aprašymas

„Bendros padėties“ koordinatėmis:

$$-x, -y, z+\frac{1}{2}$$



Posūkio matrica ir postūmiu:

$$\vec{x}' = \boldsymbol{R}\vec{x} + \vec{\boldsymbol{T}}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Spacegroup Builder

Require: $\mathbb{H}$ – a subgroup of a finite group $\mathbb{G}$
Require: g – an element of the finite group $\mathbb{G}$, $g \in \mathbb{G}$
Ensure: The list L of the operators of a subgroup $\mathbb{L} \leq \mathbb{G}$ without duplicates
Ensure: $\mathbb{L} = \langle \mathbb{H}, \langle g \rangle \rangle$
procedure SIMPLEBUIDER($\mathbb{H}, g$)
▷ Build a spacegroup generated by $\mathbb{H}$ and g
$L \leftarrow (e, h_1, h_2, \ldots, h_n)$, where $h_i \in \mathbb{H}\ \forall i$
$L_{new} \leftarrow (g)$
while L_{new} is not empty **do**
$g' \leftarrow \text{shift}(L_{new})$
$L \leftarrow (L, g')$
for all $h' \in L$ **do**
$g'' \leftarrow h'g'$
if $g'' \notin L \cup L_{new}$ **then**
$L_{new} \leftarrow (L_{new}, g'')$
end if
end for
end while
return L
end procedure

Algorithm correctness issues

Questions:

- Is this algorithm “correct”?
- Are the specified invariants specified?
- Under which assumptions the invariants hold (e.g. is it essential that the group $\mathbb{H}$ is finite)?

General program correctness

spec# Microsoft Research

Is this program correct?

```
1  class Example {
2    int x;
3
4    void Inc(int y)
5      ensures old(x) < x;
6    {
7      x += y;
8    }
9  }
10
```

home video permalink

'▶' shortcut: Alt+B

		Description
	1	Method Example.Inc(int y), unsatisfied postcondition: old(x) < x

```
c.ssc(4,8): warning CS2663: Method Example.Inc(int y), unsatisfied postcondition: old(x) < x
```

General program correctness

spec# Microsoft Research

Is this program correct?

```
 1 class Example {
 2   int x;
 3
 4   void Inc(int y)
 5     requires y > 0;
 6     ensures old(x) < x;
 7   {
 8     x += y;
 9   }
10 }
11
```

home video permalink

'▶' shortcut: Alt+B

No verification errors

General program correctness

```
using System;
class Example {
    int x;
    public Example() {
        x = 0;
    }
    public void Inc(int y)
    // requires y > 0;
    // ensures old(x) < x;
    {
        x += y;
    }
    public int value()
    {
        return x;
    }
}
```

General program correctness

```
using System;
class Example {
    int x;
    public Example() {
        // 32 bit signed int maximum value
        x = 2147483647;
    }
    public void Inc(int y)
    // requires y > 0;
    // ensures old(x) < x;
    {
        x += y;
    }
    public int value()
    {
        return x;
    }
}
```

Real life...

```
using System;
class Example {
    int x;
    public Example() {
        // 32 bit signed int maximum value
        x = 2147483647;
    }
    public void Inc(int y)
    // requires y > 0;
    // ensures old(x) < x;
    {
        x += y;
    }
    public int value()
    {
        return x;
    }
}
```

```
saulius@kolibris integer-overflow-check/ $ mono spec
-2147483648
```

Euklido algoritmas pasibaigia:

programos/perl/gcd-finitness-proof.perl:

```
sub gcd($$)
{
    my ($x, $y) = @_;
    # inv: $x is int && $y is int
    # pre: $x > 0 && $y > 0
    while( $x != $y ) {
        # pre: $x > 0 && $y > 0 && $x != $y
        if( $x > $y ) {
            # pre: $x > $y
            $x -= $y;
            # post: $x > 0
        } else {
            # pre: $x < $y ( <== $x != $y && !$x > $y )
            $y -= $x;
            # post: $y > 0
        }
        # inv: $x > 0 && $y > 0 => max( $x, $y ) > 0
        # post: max( new $x, new $y ) < max( old $x, old $y )
    }
    # post: $x == $y && $x > 0 && $y > 0
    return $x;
}
```

Euklido algoritmas tikrai duoda DBD:

programos/perl/gcd-correctness-proof.perl:

```
sub gcd($$)
{
    my ($x, $y) = @_;
    # assume: $X0 == initial $x, $Y0 == initial $y
    # pre: GCD( $x, $y ) == GCD( $X0, $Y0 )
    while( $x != $y ) {
        if( $x > $y ) {
            # pre:  GCD( $x, $y ) == GCD( $X0, $Y0 );
            $x -= $y;
            # post: GCD( $x, $y ) == GCD( $X0, $Y0 );
        } else {
            # pre:  GCD( $x, $y ) == GCD( $X0, $Y0 );
            $y -= $x;
            # post: GCD( $x, $y ) == GCD( $X0, $Y0 );
        }
        # post: GCD( new $x, new $y ) == GCD( old $x, old $y )
        # inv: GCD( $x, $y ) == GCD( $X0, $Y0 )
    }
    # post: $x == $y
    # post: $x == GCD($x,$x) == GCD($x,$y) == GCD($X0,$Y0)
    return $x;
}
```

Pradinių sąlygų užtikrinimas

programos/perl/gcd-with-asserts.perl:

```
sub gcd($$)
{
    my ($x, $y) = @_;

    assert( $x > 0 );
    assert( $y > 0 );

    while( $x != $y ) {
        if( $x > $y ) {
            $x -= $y;
        } else {
            $y -= $x;
        }
    }
    return $x;
}
```

Floating point vs BigRat

Real numbers:

$$\sum_{n=1}^{3} \frac{1}{3} = 1$$

Floats:

$$\sum_{n=1}^{3} \frac{1}{3} \neq 1$$

Real numbers:

$$\sum_{n=1}^{256} \frac{1}{256} = 1$$

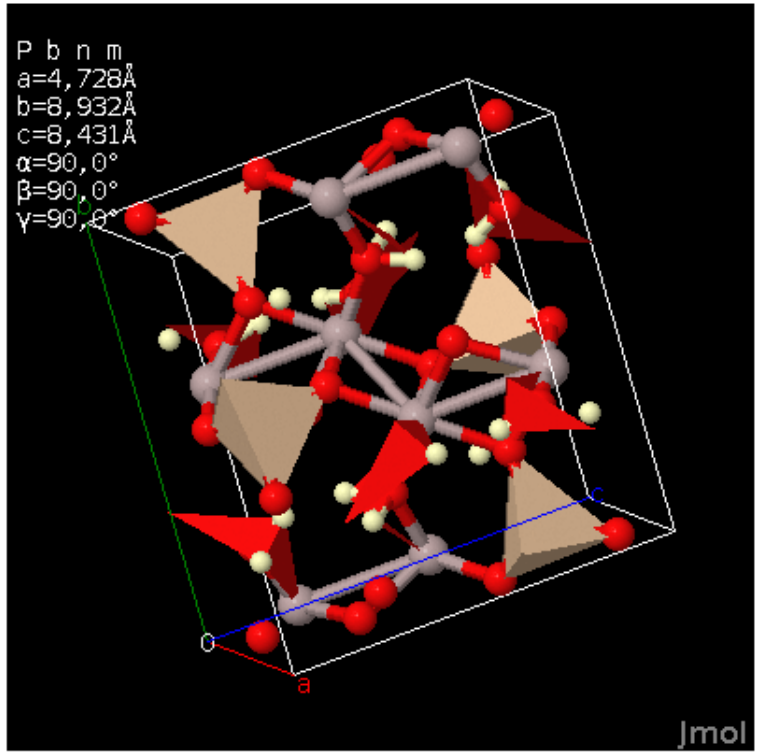
Floats:

$$\sum_{n=1}^{256} \frac{1}{256} = 1$$

Questions

- will my float lose precission?
- will my rational number overlow RAM?

Thank you!



Coordinates	2207377.cif
Original IUCr paper	HTML

http://en.wikipedia.org/wiki/Topaz

http://www.crystallography.net/2207377.html

References I

Gražulis, S., Chateigner, D., Downs, R. T., Yokochi, A. F. T., Quirós, M., Lutterotti, L., Manakova, E., Butkus, J., Moeck, P., and Le Bail, A. (2009).
Crystallography Open Database – an open-access collection of crystal structures.
Journal of Applied Crystallography, 42:726–729.