ISL Specification

ISL code is a set of *moves* over a canvas.

We start with a description of ISL grammar.

ISL Grammar

```
<program>       ::=     <move> | <move> <newline> <program>
<move>          ::=     <pcut-move> | <lcut-move> | <color-move> | <swap-
move> | <join-move>
<pcut-move>     ::=     "cut" <block> <position>
<lcut-move>     ::=     "cut" <block> <orientation> <line>
<color-move>    ::=     "color" <block> <color>
<swap-move>     ::=     "swap" <block> <block>
<join-move>     ::=     "join" <block> <block>
<orientation>   ::=     "x" | "y"
<block>         ::=     "[" <block-id> "]"
<position>      ::=     "[" <x> "," <y> "]"
<color>         ::=     "[" <r> "," <g> "," <b> "]"
<block-id>      ::=     <id> | <id> "." <block-id>
<x> | <y>       ::=     "0", "1", "2"...
<id> | <line>   ::=     "0", "1", "2"...
<r> | <g> | <b> ::=     "0", "1", "2"..."255"
<newline>       ::=     "\n"
```

Naming/Identification Schemes

ISL code mutates *the canvas*, a global map consisting of a number of blocks.

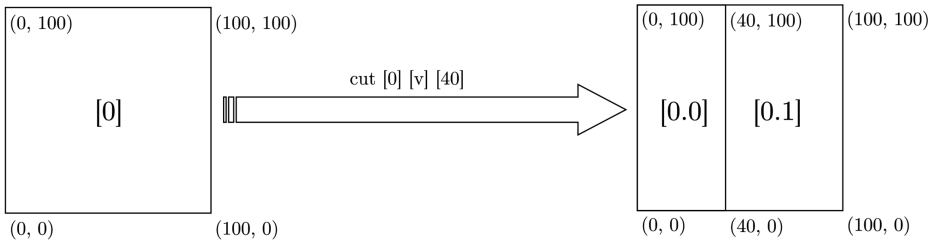
Each line mutates on or more of the blocks. **A cut move** divides a block into two or four blocks depending on the *cut type*; **a color move** changes the color of a block; **a swap move** changes the color contents of two blocks; **a join move** merges two blocks together to create a new block.

Cut and **Join** moves generate new entities, which effectively needs naming. This section explains how those pieces are named.

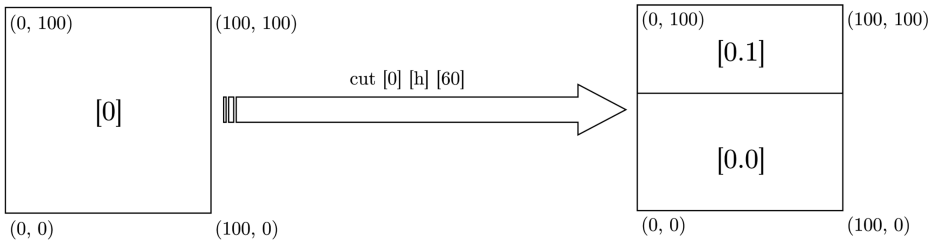
There are three types of cuts, **vertical lcut**, **horizontal lcut** and **pcut**.

Below, we see examples of each of these cut types and how the child blocks are named.

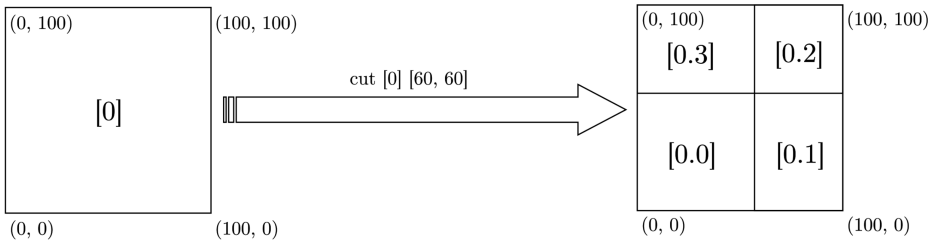
Vertical Line-Cut



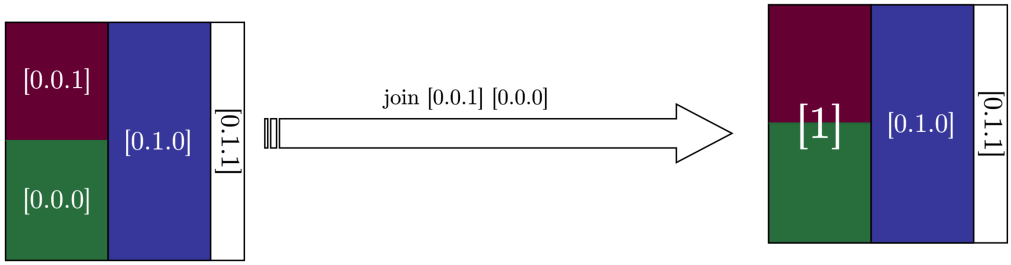
Horizontal Line-Cut



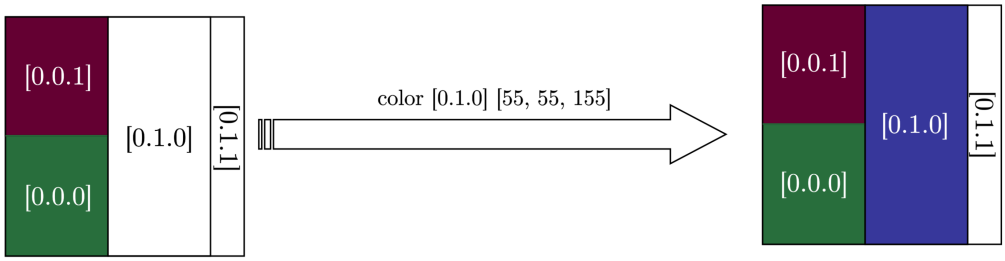
Point-Cut



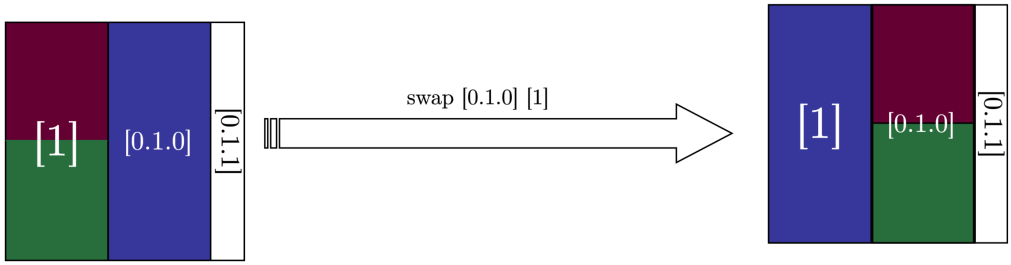
Join



Color



Swap



How ISL Moves Transform The Global Map

```
{
    "0": {
        "shape": [[0, 0], [255, 255]],
        "color": null
    }
}
```

```
(0, 255) ---------- (255, 255)
  |                      |
  |                      |
  |                      |
  |                      |
  |                      |
  |                      |
(0, 0) ------------ (255, 0)
```

A cut generates these so called *children blocks*.

ISL line: `cut [0] [100, 100]`

Output:

```
{
    "0.0": {
        "shape": [[0, 0], [100, 100]],
        "color": null
    },
    "0.1": {
        "shape": [[101, 0], [255, 100]],
        "color": null
    },
    "0.2": {
        "shape": [[0, 101], [100, 255]],
        "color": null
    },
    "0.3": {
        "shape": [[101, 101], [255, 255]],
        "color": null
    },
}
```

```
(0, 255) ----- (100, 255) ----- (255, 255)
 |                 |                 |
 |                 |                 |
 |                 |                 |
(0, 100) ----- (100, 100) ----- (255, 100)
 |                 |                 |
 |                 |                 |
 |                 |                 |
(0, 0) ------- (100, 0) ------- (255, 0)
```

ISL line: `color [0.1] [120, 120, 120]` ISL line: `color [0.2] [160, 80, 120]`

Output:

```
{
    "0.0": {
        "shape": [[0, 0], [100, 100]],
        "color": null
    },
    "0.1": {
        "shape": [[101, 0], [255, 100]],
        "color": [120, 120, 120]
    },
    "0.2": {
        "shape": [[0, 101], [100, 255]],
        "color": [160, 80, 120]
    },
```

```
    "0.3": {
        "shape": [[101, 101], [255, 255]],
        "color": null
    },
}
```

```
(0, 255) ----- (100, 255) ------ (255, 255)
 |                  |                 |
 | [160, 80, 120]   |                 |
 |                  |                 |
(0, 100) ----- (100, 100) ------ (255, 100)
 |                  |                 |
 |                  | [120, 120, 120] |
 |                  |                 |
(0, 0) ------- (100, 0) -------- (255, 0)
```

ISL line: `swap [0.2] [0.1]`

Output:

```
{
    "0.0": {
        "shape": [[0, 0], [100, 100]],
        "color": null
    },
    "0.1": {
        "shape": [[101, 0], [255, 100]],
        "color": [160, 80, 120]
    },
    "0.2": {
        "shape": [[0, 101], [100, 255]],
        "color": [120, 120, 120]
    },
    "0.3": {
        "shape": [[101, 101], [255, 255]],
        "color": null
    },
}
```

```
(0, 255) ----- (100, 255) ------ (255, 255)
 |                  |                 |
 | [120, 120, 120] |                 |
 |                  |                 |
(0, 100) ----- (100, 100) ------ (255, 100)
 |                  |                 |
 |                  | [160, 80, 120]  |
```

```
  |                   |                   |
(0, 0) ------- (100, 0) -------- (255, 0)
```