User specify:

```
1) k0
2) pochoir_if (g1) k1
3) if (g2) k2
4) if (g3) k3
```

Pochoir generates:

```
1) !g1 + !g2 + !g3 => k0
2) g1 + !g2 + !g3 => k0 || k1
3) g2 || g3 => k0; if (g1) k1; if (g2) k2; if (g3) k3
```

Assuming k0, m x pochoir_if's, and some if's, the Pochoir compiler will generate only 2^m + 1 kernels

```
=====================================================================================
```

Three notations for conditional guards in inner loops:
1) exclusive_if: the construct denotes the if () else if () ... structure. The regions guarded by this construct are mutually exclusive with other exclusive_if's
2) inclusive_if: indicates a region that may overlap with other regions guaded by exclusive_if's or inclusive_if's
3) tiny_inclusive_if: the construct denotes a region that may overlap with other regions, while the area / amount of points in the region is tiny.

Example:

Pochoir specification:

```
exclusive_if (g0) {k0;}
exclusive_if (g1) {k1;}
inclusive_if (g2) {k2;}
inclusive_if (g3) {k3;}
tiny_inclusive_if (g4) {k4;}
tiny_inclusive_if (g5) {k5;}
```

Equivalent looping implementation:

```
for ... {
    if (g0) {
        k0;
    } else if (g1) {
        k1;
    }
    if (g2) {k2;}
    if (g3) {k3;}
    // The user's hint indicates that the sub-regions guarded by g4 and g5 are
    // very tiny fraction of the entire domain
    if (g4) {k4;}
    if (g5) {k5;}
} /* end for */
```

The code clones automatically generated by Pochoir:

```
1) if (g0 & !g2 & !g3 & ! g4 & ! g5) : k0
2) if (g0 & !g2 & g3 & ! g4 & ! g5) : k0; k3;
3) if (g0 & g2 & !g3 & ! g4 & ! g5) : k0; k2;
4) if (g0 & g2 & g3 & ! g4 & ! g5) : k0; k2; k3;
5) if (g1 & !g2 & !g3 & ! g4 & ! g5) : k1
6) if (g1 & !g2 & g3 & ! g4 & ! g5) : k1; k3;
7) if (g1 & g2 & !g3 & ! g4 & ! g5) : k1; k2;
8) if (g1 & g2 & g3 & ! g4 & ! g5) : k1; k2; k3;
9) if (!g0 & !g1 & !g2 & !g3 & ! g4 & ! g5) : nop
10) if (!g0 & !g1 & !g2 & g3 & ! g4 & ! g5) : k3;
11) if (!g0 & !g1 & g2 & !g3 & ! g4 & ! g5) : k2;
12) if (!g0 & !g1 & g2 & g3 & ! g4 & ! g5) : k2; k3;
```

```
13) if (g4 || g5) :
    // Since the user indicates that the regions guarded by g4 and g5 are quite
    // small, so it doesn't matter a lot to put all conditional guards check in
    // the inner loops
    if (g0) {
        k0;
    } else if (g1) {
        k1;
    }
    if (g2) k2;
    if (g3) k3;
    if (g4) k4;
    if (g5) k5;
```

So, in summary, if there are 'm' exclusive_if's, 'n' inclusive_if's, and 'k' tiny_inclusive_if's, Pochoir compiler should generate "m x 2^n + 1" code clones.