v: **{x=x0&y=y0}**  if (x<y) {x=x+y; y=x-y; x=x-y} **{x=max(x0,y0) & y=min(x0,y0)}**

according to the conditional rule, we must prove two premises:

v0: **{x=x0&y=y0 & x<y}**  {x=x+y; y=x-y; x=x-y} **{x=max(x0,y0) & y=min(x0,y0)}**

v1: **x=x0&y=y0& !(x<y) 🡺 x=max(x0,y0) & y=min(x0,y0)**

v1 is a tautology. Hence we focus on v0. We apply the sequence rule twice, by introducing two intermediate predicates:

v01: **{x=x0&y=y0 & x<y}**  x=x+y **{x=x0+y0&y=y0 & x0<y0}**

v02: **{x=x0+y0&y=y0 & x0<y0}**  y=x-y **{x=x0+y0&y=x0 & x0<y0}**

v03: **{x=x0+y0&y=x0 & x0<y0}** x=x-y **{x=max(x0,y0) & y=min(x0,y0)}**

To prove v01, we apply the assignment rule:

v010: **x=x0&y=y0 & x<y 🡺 x+y=x0+y0&y=y0 & x0<y0. This is a tautology.**

We apply the assignment rule to v02:

**V020 : x=x0+y0&y=y0 & x0<y0 🡺 x=x0+y0&x-y=x0 & x0<y0. This is a tautology.**

We apply the assignment rule to v03:

v030: **x=x0+y0&y=x0 & x0<y0 🡺 x-y=max(x0,y0) & y=min(x0,y0).**

**x-y= y0. y=x0. x0<y0. This is a tautology.**

**--------------------------------------------------------**

v: **{x=x0 & y=y0}** while (y !=0) {x=x+1 ; y=y-1} **{x=x0+y0}**

|  |  |  |  |
| --- | --- | --- | --- |
| iteration | x | y | x+y |
|  | 5 | 8 | 13 |
|  | 6 | 7 | 13 |
|  | 7 | 6 | 13 |
|  | 8 | 5 | 13 |
|  | 9 | 4 | 13 |

We apply the while rule, with inv= (x+y=x0+y0)

v0: **x=x0&y=y0 🡺 x+y = x0+y0. This is a tautology.**

v1: **{x+y=x0+y0 & y !=0}** x=x+1 ; y=y-1 **{x+y=x0+y0}**

v2: **x+y=x0+y0 & y=0 🡺 x=x0+y0. This is a tautology.**

We focus on v1 ; we apply the sequence rule to v1. We use int = (x+y=x0+y0+1)

v10: **{x+y=x0+y0 & y !=0}** x=x+1 **{x+y=x0+y0+1}**

v11: **{x+y=x0+y0+1}** y=y-1 **{x+y=x0+y0}**

We apply the assignment rule to v10 and v11 :

v100: **x+y=x0+y0 & y !=0 🡺 x+1+y=x0+y0+1. This is a tautology.**

v110: **x+y=x0+y0+1 🡺 x+y-1=x0+y0. This is a tautology.**

Since v100 is valid, so is v10. (assignment rule)

Since v110 is valid, so is v11. (assignment rule)

Since v10 and v11 are valid, so is v1. (sequence rule)

Since v0, v1, v2 are all valid, so is v. (while rule).

int n, f, k;

**v: {n=n0}**  {f=1; k=1; while (k!=n+1) {f=f\*k; k=k+1;}} **{f=n0!}**

Sequence rule applied to v, with int: n=n0 & f=1 & k=1.

v0: **{n=n0}** f=1; k=1 **{ n=n0 & f=1 & k=1}**

v1: **{ n=n0 & f=1 & k=1}** while (k!=n+1) {f=f\*k; k=k+1;} **{f=n0!}**

We apply the assignment rule to v0,

v00: **n=n0 🡺 n=n0 & 1=1 & 1=1. This is valid.**

We apply the while rule to v1, with inv: n=n0 & f=(k-1)!

v10: **n=n0 & f=1 & k=1 🡺 n=n0 & f=(k-1)!** . 0!=1. This is a tautology.

v11: **{n=n0 & f=(k-1)! & (k!=n+1)}** f=f\*k; k=k+1 **{n=n0 & f=(k-1)!}**

v12: **n=n0 & f=(k-1)! & (k=n+1) 🡺 f=n0!. This is a tautology.**

We apply the sequence rule to v11, with int: n=n0 & f=k!.

v110: **{n=n0 & f=(k-1)! & (k!=n+1)}** f=f\*k **{n=n0 & f=(k)!}**

v111: **{n=n0 & f=(k)!}** k=k+1 **{n=n0 & f=(k-1)!}**

We apply the assignment rule to v110, v111:

v110: **n=n0 & f=(k-1)! & (k!=n+1) 🡺 n=n0 & f\*k=(k)!. This is a tautology..**

v111: **n=n0 & f=(k)! 🡺 n=n0 & f=(k+1-1)!. This is a tautology.**