main() { p() { q() {
n00 ENTRY n05 ENTRY n11 ENTRY
n01 c_1 : p() n06 a * b n13 a = 5 n15 b = 6
n21 ret p() n07 c_3 : q() n08 c_4 : q() n16 EXIT
n02 c * d n27 ret q() n28 ret q()}
n03 c_2 : q() n09 c = 5
n23 ret q() n10 EXIT
n04 EXIT }
}

-¿n00n01 -¿n21n02 -¿n02n03 -¿n23n04
-¿n05n06 -¿n06n07 -¿n06n08 -¿n27n09 -¿n28n09 -¿n09n10
-¿n11n13 -¿n11n15 -¿n13n16 -¿n15n16
[linestyle=dashed,angleB=90]-¿n01n05
[linestyle=dashed,angleB=75]-¿n03n11
[linestyle=dashed,angleB=90]-¿n07n11
[linestyle=dashed,angleB=135]-¿n08n11
[linestyle=dashed,angleB=-135]¡-n21n10
[linestyle=dashed,angleB=-45]¡-n23n16
[linestyle=dashed,angleB=-90]¡-n27n16
[linestyle=dashed,angleB=-135]¡-n28n16

Pgm	**gen**	**kill**	**IN**	**OUT**
main() {				
ENTRY			$\lambda, 00$	$\lambda, 00$
c_1 : call p()			$\lambda, 00$	$c_1, 00$
c * d	c*d		$\lambda, 00$	$\lambda, 01$
c_2 : call q()			$\lambda, 01$	$c_2, 01$
EXIT			$\lambda, 01$	$-$
}				
p() {				
ENTRY			$c_1, 00$	$c_1, 00$
a * b	a*b		$c_1, 00$	$c_1, 10$
if(...) then			$c_1, 10$	$c_1, 10$
c_3 : call q()			$c_1, 10$	$c_1c_3, 10$
else			$-$	$-$
c_4 : call q()			$c_1, 10$	$c_1c_4, 10$
c = 5		c*d	$c_1, 00;\ c_1, 00$	$c_1, 00$
EXIT			$c_1, 00$	$\lambda, 00$
}				
q() {				
ENTRY			$c_1c_3, 10;\ c_1c_4, 10;\ c_2, 01$	$c_1c_3, 10;\ c_1c_4, 10;\ c_2, 01$
if(...) then			$c_1c_3, 10;\ c_1c_4, 10;\ c_2, 01$	$c_1c_3, 10;\ c_1c_4, 10;\ c_2, 01$
a = 5		a*b	$c_1c_3, 10;\ c_1c_4, 10;\ c_2, 01$	$c_1c_3, 00;\ c_1c_4, 00;\ c_2, 01$
else				
b = 6		a*b	$c_1c_3, 10;\ c_1c_4, 10;\ c_2, 01$	$c_1c_3, 00;\ c_1c_4, 00;\ c_2, 01$
EXIT			$c_1c_3, 00;\ c_1c_4, 00;\ c_2, 01$	$c_1, 00;\ c_1, 00;\ \lambda, 01$
}				

bit1 bit2: bit1 represents availability of a*b, bit2 of c*d you can also initialize boundary with $\perp$ instead of 0. Meet at each point, ignoring call string part will give the availability.