

P	Q	n	stdout
P1 while $n < 2$	Q1 $n += 1$	0	
P2 print n	Q1 $n += 1$	0	0
P1 while $n < 2$	Q1 $n += 1$	0	
P1 while $n < 2$	Q2 $n += 1$	1	
P2 print n	Q2 $n += 1$	1	1
P1 while $n < 2$	Q2 $n += 1$	1	
P2 print n	Q2 $n += 1$	1	
P2 print n	(end)	2	2 1a)
P1	Q1	0	
P2	Q1	0	0
P1	Q1	0	
P2	Q1	0	0
P1	Q1	0	
P2	Q1	0	
P2	Q1	1	
P2	(end)	2	2
P1	(end)	2	
(end)	(end)	2	1b)
P1	Q1	0	
P2	Q1	0	0
P1	Q1	0	
P2	Q1	0	
P2	Q2	1	
P2	(end)	2	2
P1	(end)	2	
(end)	(end)	2	1c)

Ad)

Mit Fairness (jeder Prozess läuft irgendwann)
Der Algorithmus terminiert in jedem Szenario

Ohne Fairness

Es gibt Szenarien in denen α_2 nicht ausgeführt wird

P	Q	n	stdout
p1	Q1	0	
p1	Q2	1	
p1	(end)	2	
(end)	(end)	2	

1e)

Sobald ein mal 2 ausgegeben wurde, kann die Bedingung von p1 nicht mehr für einen 2. durchgezählt werden.

1f)

P	Q	n	flag
p1	Q1	0	false
p1	Q2	0	false
p1	Q3	0	false
p1	(end)	0	true
(end)	(end)	0	true

29)

Mögliche Endwerte sind $\{0, 1\}$
 Lösung $n=0$ siehe 2a)
 Lösung $n=1$

p1	Q1	0	false
p1	Q2	0	false
p1	Q2	0	false
p2	Q2	0	false
p1	Q3	1	false
p1	(end)	1	true
(end)	(end)	1	true

26)

2c)

Ohne Fairness: Dauerschleife Prozess p

Mit Fairness: Obwohl beide Prozesse durchlaufen,
gibt es Szenarien bei denen
Prozess Q nur lauft wenn Flag und
n nicht false, 0 sind.

p1 p2 Q1 Q2 p1 Q1 Q2 p2