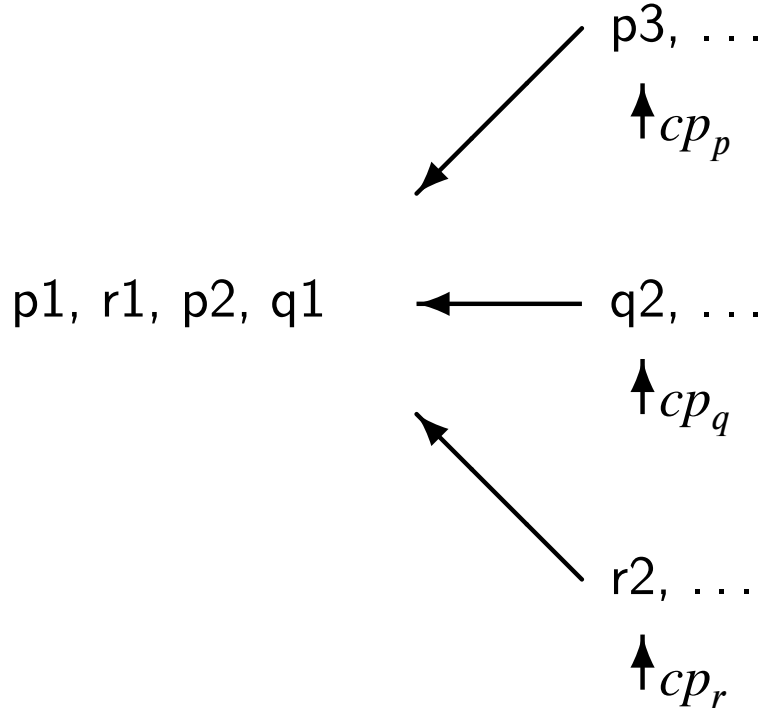
Interleaving as Choosing Among Processes



Possible Interleavings

p1→q1→p2→q2,

p1→q1→q2→p2,

p1→p2→q1→q2,

q1→p1→q2→p2,

q1→p1→p2→q2,

q1→q2→p1→p2.

Algorithm 2.1: Trivial concurrent program

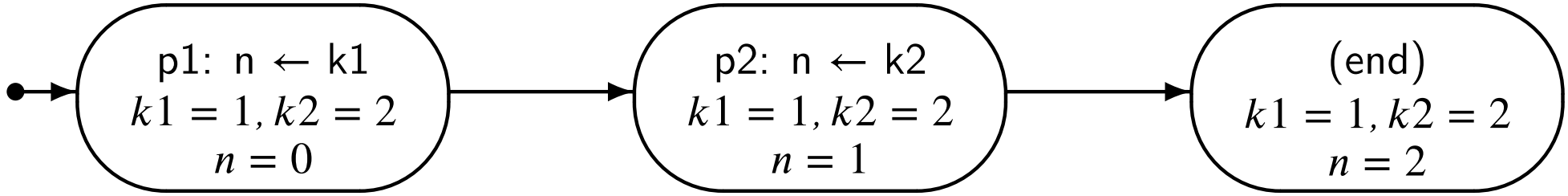
integer n ← 0

p	**q**
integer k1 ← 1	integer k2 ← 2
p1: n ← k1	q1: n ← k2

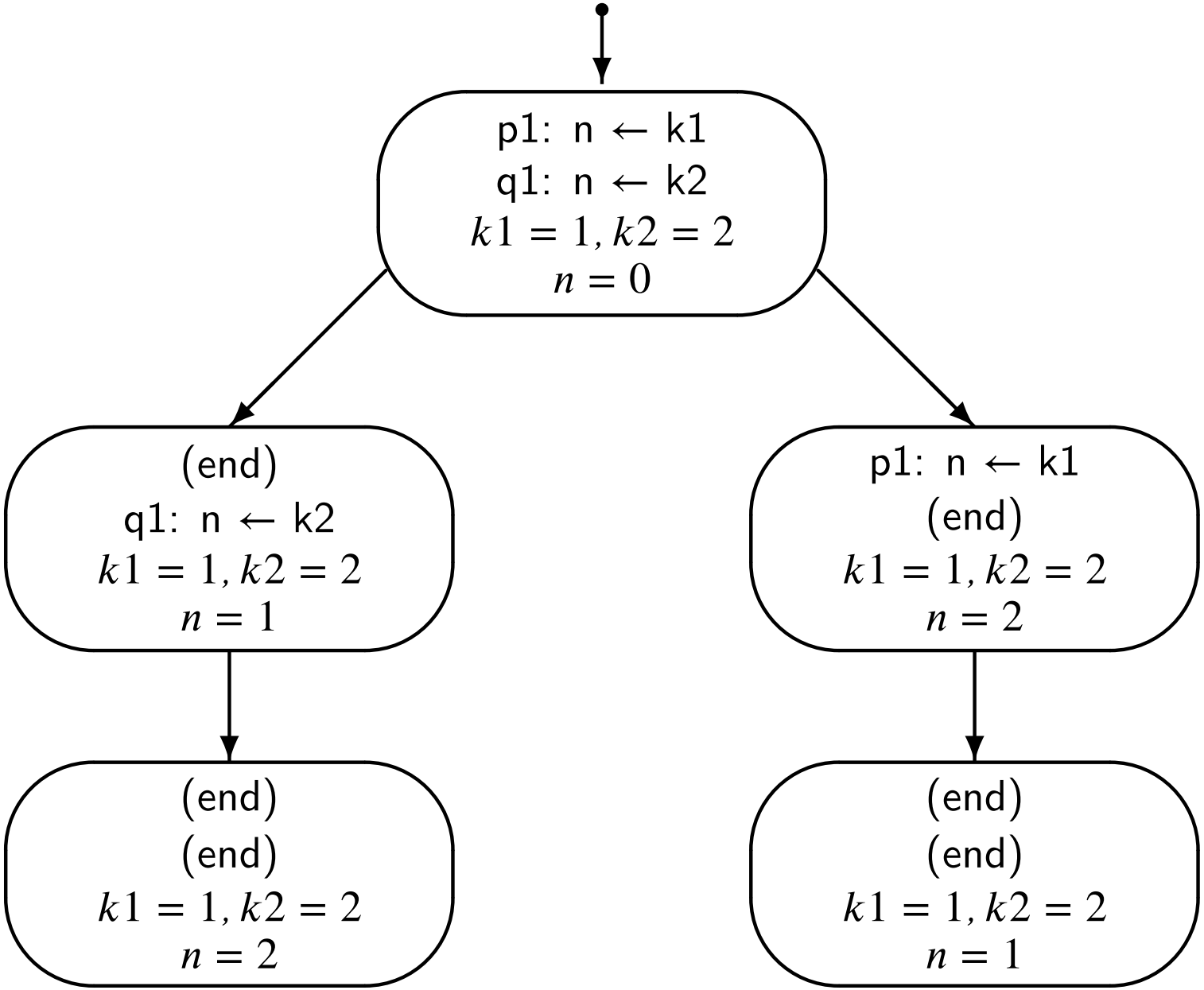
Algorithm 2.2: Trivial sequential program

```
                integer n ← 0
        integer k1 ← 1
        integer k2 ← 2
p1:     n ← k1
p2:     n ← k2
```

State Diagram for a Sequential Program



State Diagram for a Concurrent Program



Scenario for a Concurrent Program

Process p	Process q	n	k1	k2
p1: n←k1	q1: n←k2	0	1	2
(end)	**q1: n←k2**	1	1	2
(end)	(end)	2	1	2

Algorithm 2.3: Atomic assignment statements	
integer n ← 0	
p	**q**
p1: n ← n + 1	q1: n ← n + 1

Scenario for Atomic Assignment Statements

Process p	Process q	n
p1: n←n+1	q1: n←n+1	0
(end)	**q1: n←n+1**	1
(end)	(end)	2

Process p	Process q	n
p1: n←n+1	**q1: n←n+1**	0
p1: n←n+1	(end)	1
(end)	(end)	2

Algorithm 2.4: Assignment statements with one global reference	
integer n ← 0	
p	**q**
integer temp	integer temp
p1: temp ← n	q1: temp ← n
p2: n ← temp + 1	q2: n ← temp + 1

Correct Scenario for Assignment Statements

Process p	Process q	n	p.temp	q.temp
p1: temp←n	q1: temp←n	0	?	?
p2: n←temp+1	q1: temp←n	0	0	?
(end)	**q1: temp←n**	1	0	?
(end)	**q2: n←temp+1**	1	0	1
(end)	(end)	2	0	1

Incorrect Scenario for Assignment Statements

Process p	Process q	n	p.temp	q.temp
p1: temp←n	q1: temp←n	0	?	?
p2: n←temp+1	**q1: temp←n**	0	0	?
p2: n←temp+1	q2: n←temp+1	0	0	0
(end)	**q2: n←temp+1**	1	0	0
(end)	(end)	1	0	0

Algorithm 2.8: Volatile variables

integer n ← 0

p	**q**
integer local1, local2	integer local
p1: n ← *some expression*	q1: local ← n + 6
p2: *computation not using n*	q2:
p3: local1 ← (n + 5) ∗ 7	q3:
p4: local2 ← n + 5	q4:
p5: n ← local1 * local2	q5:

Algorithm 2.9: Concurrent counting algorithm

integer n ← 0

p	**q**
integer temp	integer temp
p1: do 10 times	q1: do 10 times
p2: temp ← n	q2: temp ← n
p3: n ← temp + 1	q3: n ← temp + 1