Strict Alternation: Sample Process Execution shows Why it Fails.

Initially:

- turn = 0
- Index, *i* is used to indicate a process. If *i* = 0 => Process 0; if *i* = 1 => Process 1

Examine the execution of both process. Time progresses in the downward direction in this table...

Process 0 (i = 0)	Process 1 (i = 1)
Entry section: While(turn != 0) --- UNTRUE since turn = 0	
Critical section	Entry section: While(turn != 1) --- TRUE since turn = 0 (i.e. 0 != 1 is a true statement)
Critical section continues	; // Busy wait
Critical section continues	Process 1 continually checks in the while statement to see if turn != 1
Exit section: turn = 1 – i = 1 – 0 = 1	
Remainder section	While(turn != 1) --- UNTRUE since turn = 1 now (thanks to process 0). (i.e. 1 != 0 is an **untrue** statement) So skip NULL statement *" ; // Busy wait "*, and enter critical section...
Remainder section continues	Critical section
Entry section: While(turn != 0) --- TRUE since turn = 1	Critical section continues
; // Busy wait	Critical section continues
	Exit section: turn = 1 – i = 1 – 1 = 0
Entry section: While(turn != 0) --- UNTRUE since turn = 0 now (thanks to process 1). So skip NULL statement *" ; // Busy wait "*, and enter critical section...	Remainder section
Critical section	Remainder section continues
Critical section continues	Remainder section continues
Exit section: turn = 1 – i = 1 – 0 = 1	Remainder section continues
Process 0 loops and wishes to enter its critical section again. It tests turn: Entry section: while(turn != 0) --- TRUE since turn = 1 (i.e. 1 != 0 is a true statement)	Remainder section continues
; // Busy waiting	Remainder section continues

Process 0 (i = 0)	Process 1 (i = 1)
• Process 0 busy waits eventhough Process 1 is only executing its Remainder Section (a non-critical section) • Process 0 is being blocked by Process 1 because Process 0 is waiting on Process 1 to change the value of turn to 0. • Since Process 0 is being blocked by Process 1 => violates the Progress condition: No process executing a non-critical section may block a process from entering its critical section	Remainder section continues
	Remainder section continues
	Remainder section continues