ALUMNAS: Claudia Casado Poyates

Olga Rodríguez Acouedo

Watalia Carcia Dominguez

SOUTH NORTH NORTH SOUTH NORTH TODEL DE KIYOTAKI NORTH TODEL DE KIYOTAKI

Coso básico

COLT-NORTH: COICHES del norte dentro del time!

CON-SOUTH: coches del sur dentro del time!

INVARIANTE: 7 (COR-NORTH >O A COR-SOUTH >O) A

COL-MORTH > O V COL- PONTH > O

no-inside-NORTH: Condición (cor-NORTH == 0)

no-inside-south: Condicion (car-south == 0)

gg: monts - super - south :

1 car. south >0 4

no-inside - NORTH . wait (car- NORTH == 0)

COC- SOUTH += 1 } COC- SOUTH > 0 9

DOM : COLT = SOUTH > 0 1 COLT NORTH == 0 => 7 (COLT NORTH > 0 1 COLT = SOUTH > 0) 1

^ con-NORTH > 0 ^ con-south > 0 <=> 7 (golso ^ cierte) ^ cierte

^ cierto (=> > Salso ^ cierto (=> cierto

```
def: Leaves - tunnel - SOUTH:
     } car-south > G ~ car-north == 0 }
      car - south -= 1
      no-inside - SOUTH . SIGNOR () 1 COL-SOUTH > 0 8
   DOMO: COL- 2001 H > O V COL- NOBIN == 0 =>
      7 (car-NORTH 20 1 car south 2011 car-NORTH 20 1 car-south 2020)
    (=>7 ( galso 1 car_south >0) 1 cierto 1 cierto (=>
    (=) 7 (galso) 1 cierto (=) cierto
 as: wants - enter - MORTH:
    1 COX- NORTH >, O &
     no-inside - SOUTH . wort (cor-south = = 0)
    COX-NORTH == 4 1 COX-NORTH > 04
  DOMO: COL-NOGIM >0 V COL- EOULH == 0 =>
     7 (car_ worth >0 1 car_ south >0 ) 1 car_ worth >,0 1 car_ south >,0 6.
    7 ( cierto a goilso) a cierto a cierto (=> cierto
 des: leaves - tunnel - NORTH:
      } cor- NORTH > 0 ~ cor- south = = 0 4
      COX- NORTH -= 1
      no - inside - NORTH. signal book - NORTH > 0 9
 Dono: car-NORTH >0 . car-south ==0 =
  7 ( COT- NORTH >0 1 COT- SOUTH >0 ) 1 COT- NORTH >0 1 COT- SOUTH >0 =>
  1 (con-moranso a golso) à ciente à ciente les
  ? (gaiso) à cierto (=> cierto.
```

Caso turnos

```
car-NORTH: coches del norte dentro del time!
car-south: coches del sur dentro del timel
queue north: coches del norte esperando para entrar
queue-soury: coches del sur esperando para entrar
turn: variable que determina la dirección de los
      coches que pueden pasar
               0 -> si pasan cos del norte
               1 -> si posan los del sur
INVARIANTE: 7 ( CORTH > 0 ~ CORTH > 0 ) A
             COL-MORIN 30 V COL-800EM 30 V
             (turn == 0 v turn == 1) A
            queue - NORTH > 0 , queue - 300 EN > 0
no-inside-NORTH: Condicion (car-NORTH == 0 ^ (turn == 1 v
                            queue_NORTH == 0))
no-inside-south: Condicion (car-south == 0 1 (turn == 0 V
                           Queue - 200TH = =0))
ces wants enter-south:
    {cor-2001+1 > 0 v duens-2001+1 > 0 {
    L = + HTUOS - = 1
    no-inside-MORTH. wait ( COE-NORTH == 0 1 ( turn = 1 )
                            queue - NORTH = = 01 )
    JUEUR-SOUTH -= 1
    car_south += 1
    ] car-south > 0 ~ queue. south > 0 ~ queue - worth > 0 4
```

Demo: car_NOQTH == 0 ^ (turn == 1 ~ queue_NORTH == 0) ^ car_south > 0 ^

queue_south > 0^queue_NORTH > 0 =>

T (car_NOQTH > 0 ^ car_south > 0) ^ car_NORTH > 0 ^

car_south > 0 ^ (turn == 0 ~ turn == 1) ^

queue_NORTH > 0 ^ queue_south > 0 <=>

T (galso ^ cierte) ^ cierte ^ cierte ^ (turno == 0 ~ turn == 1)

^ cierte ^ cierte (=> cierto (*)

1 • si turn == 1 -> (turn == 0 ~ turn == 1) = cierte

* 1 • si turn!=1 ^ queue_NORTH == 0 => turn == 0 => cierto

Demo: Car_south >0 ^ Car_North=0 ^ turn ==0 ^

queue - South >0 ^ queue_North>0 =)

7 (Car_North>0 ^ Car_south>0) ^ Car_North>0 ^

Car_South>0 ^ (turn == 0 v turn == 1) ^

queue_South>0 ^ queue_North>0 (=>

queue_South>0 ^ queue_North>0 (=>

queue_South>0 ^ car_south>0) ^ cierto ^ cierto ^

7 (galso ^ car_south>0) ^ cierto ^ cierto ^

cierto ^ cierto (=> cierto

Dono car_500tH ==0 ^ (turn==0 v queue_500TH ==0) ^ car_ NORTH > 0

^ queue_NORTH > 0 ^ queue_500TH > 0

7 (car_NORTH>0 ^ car_500TH > 0) ^ car_NORTH > 0 ^

car_500TH > 0 ^ (turn==0 v turn==1) ^ queue_NORTH>0 ^

^ queue - 500 tH ≥ 0 (=> 1 (cierto 1 8also) 1 cierto 1

cierto 1 (turn ==0 v turn ==1) 1 cierto 1 cierto 1 cierto 1

(o si turn ==0 => (turn ==0 v turn ==1) = cierto

* (o si turn != 0 1 queue - 500 tH ==0 => turn =1 => cierto

Demo: car NORTH > 0 × car - SOUTH == 0 × turn == 1 ×

T (con-NORTH>C ^ con-SOUTH>O) ~ con-NORTH>O ~

con-SOUTH>O ~ (turn==0 v turn==1) ~

queue-South>O ~ queue-North>O (=>

r(con-NORTH>O ~ galso) ~ cierto ~ cierto ~ cierto ~

cierto ~ cierto ~ cierto ~ cierto ~ cierto ~

cierto ~ cierto ~ cierto ~ cierto ~