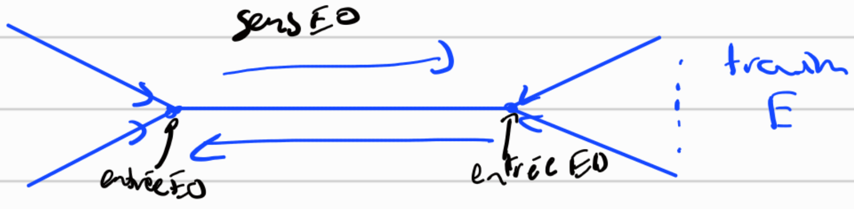
Voie unique



```
C chan bool

func prog () { msg chan string
               c ← string
               c ← true;
               v =← c
```

```
train (direction) {

      while (1) {
            arriver voie unique;
            Entrée (direction);
            rouler sur la voie unique;
            sortir ();
            direction ← inverse(direction);
      }
         func when (b bool, c chan bool) chan bool {
                 if b {return c;}
                 else {nil;}
```

variables d'acceptation

Entrée (direct°)	train circule et sens de la voie identique à celui de circulat° du train.
sortir	

```
bool traincircule;
sens sensvoie;

Entree (direction) {
        si train circule ∧ sens voie != direction alors
                                    attendre ( acces) ;

func voie unique ( entréeEO chan bool , entreeEO chan bool, sortir chan bool)
        traincircule = false ;
        sens_courant = sensEO;
        for {
          select {
             case <- if condition { entreeOE
                        }else { entreeOE } :
               sens-courant := sensOE ;
               tran circule := true;
             case <- when (traincircule == false , entreeOE) :
               sens -courant := SensOE ;
               traincircule := true;
            case <- sortir :
                 train circule : = false ;
           }
      }
}
   func train (entree chan bool, sortir chan bool )
                   entree <- true;
                   sortir <- true ;
   }
   go train (entreeEO, sortir) ;
```

```
func   voie_unique (entreEO chan bool, entreOE chan bool,
                    sortir chan bool) {
    sens_courant := sensEO;
    traincircule := false;

    for {
        select {
                case <- when (sens_courant == sensOE || nb_trains == 0,
                              entreEO) :

                    sens_courant := sensOE ;
                    nbtrains ++ ;
                                            sens du train
                                          est identique au sens
                                                de la voie
              case <- when (sens_courant == sensOE || nbtrains == 0,
                                            and nbtrains(N)
                            entreEO) :
                    sens_courant := sensEO ;
                    nbtrains ++ ;
              case <- sortir :
                    nbtrain -- ;
              }
        }
}
```