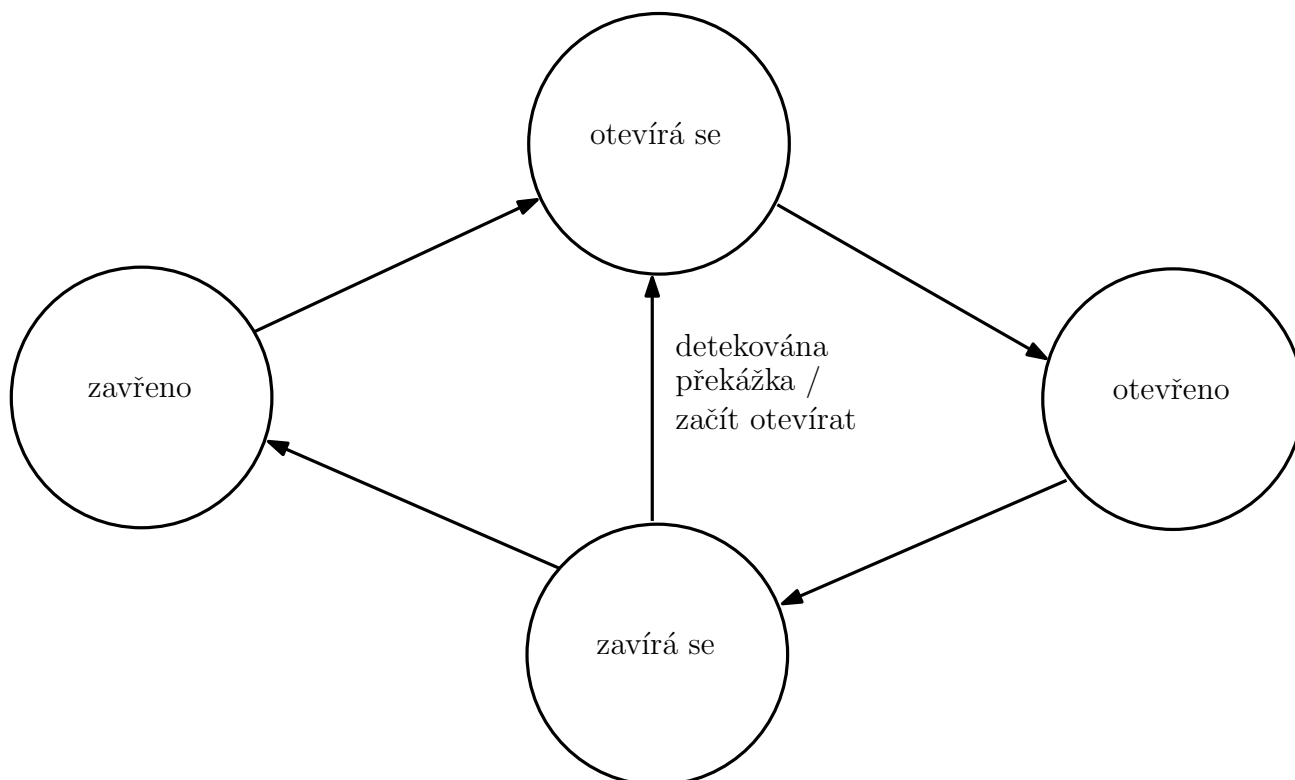


## 1 Problém 2

Abychom do automatu pro ovládání dveří výtahu zahrnuli i senzor překážky, stačí přidat jen jediný přechod do Obrázku 11 z časopisu:



Obrázek 1: Automat dveří

Teď se musíme zamyslet nad tím, jak ten stroj pozná, že se před ním nachází překážka. Záleží též na tom, kde je ten senzor umístěn. Zde budu předpokládat, že se senzor nachází na těch pohybujících se částí dveří tak, že budou schopny změřit díru mezi oběma částmi dveří. Tím pádem pokud senzor bude vracet hodnoty od 0 do 1, pak by v případě, kdy překážka nebrání v zavírání, budou hodnoty otevřenosti dveří a hodnoty senzoru stejné (samozřejmě v rozmezí odchylky). Proto pro detekci překážky musí platit:

$$|\text{senzor} - \text{dveře}| > \text{odchylka}$$

Protože jsme v automatu přidali jen jeden přechod, pro implementaci v kódu 6 z časopisu nám stačí tento kód jen mírně upravit (předpokládám, že v simulátoru je implementována metoda `e.getDoorSensor(id: str, floor: int) -> float`):

```

1
2 # ...
3
4 def jeZdePrekazka(e: elevators.Simulator, id: str, floor: int) -> bool:
5     return abs(
6         e.getDoorSensor(id, floor) - e.getDoorsPosition(id, floor)
7     ) > GD.PRECISION
8
9 def prechodovaFunkce(e: elevators.Simulator) -> None:
10    # ...
11
12    # dole se zavirajicimi se dvermi
13    elif GD.posState == GD.POS_DOWN and GD.doorState == 2:
14        if jeZdePrekazka(e, 'A', GD.POS_DOWN):
15            GD.doorState = 1
16        elif zavirejDvere(e, 'A', GD.POS_DOWN):
17            GD.doorState = 0
18            GD.posState = 11

```

```
19         e.speedUp('A')
20         # nahore se zavirajicimi se dvermi
21     elif GD.posState == GD.POS_UP and GD.doorState == 2:
22         if jeZdePrekazka(e, 'A', GD.POS_UP):
23             GD.doorState = 1
24         elif zavirejDvere(e, 'A', GD.POS_UP):
25             GD.doorState = 0
26             GD.posState = 12
27             e.speedDown('A')
28
29     # ...
```