



Lift megvalósítása UPPAAL rendszerben

FÜLÖP EDVÁRD – GX8DFG

Rendszer működése

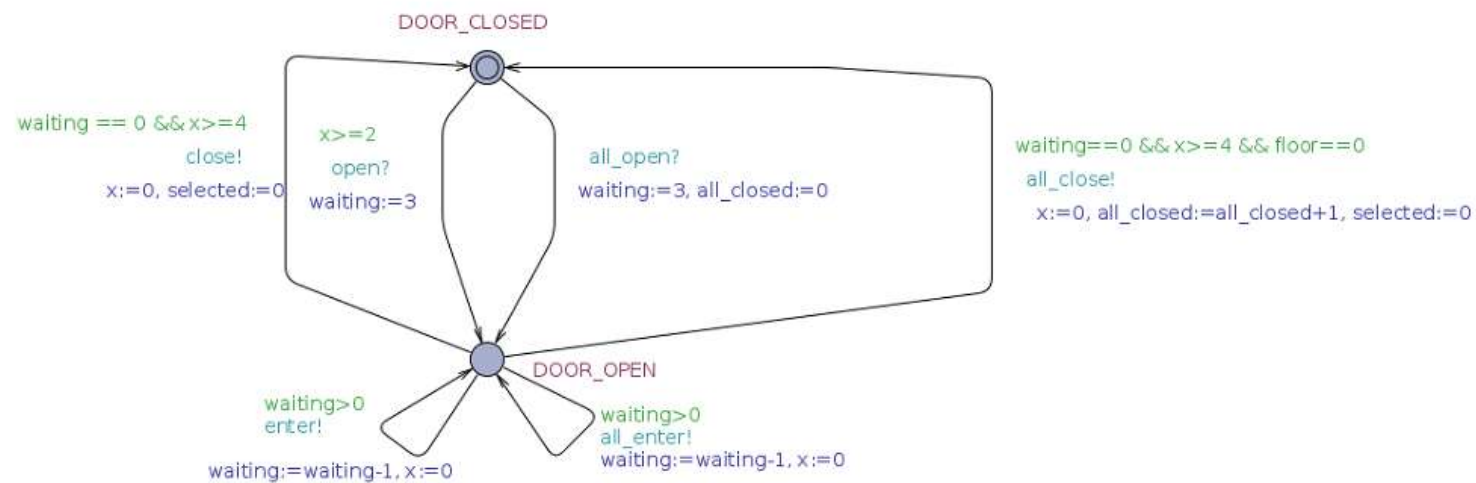
Az UPPAAL rendszer egy liftet szimulál. A lift 3 emelet között mozog. A liftnek két végében található ajtó.

- ▶ Földszint (mind a két ajtó kinyílik)
- ▶ Első emelet (csak az egyik ajtó nyílik ki)
- ▶ Második emelet (csak a másik ajtó nyílik ki)

A megvalósításhoz az alábbi két template-t hoztam létre

- ▶ Door
- ▶ Elevator

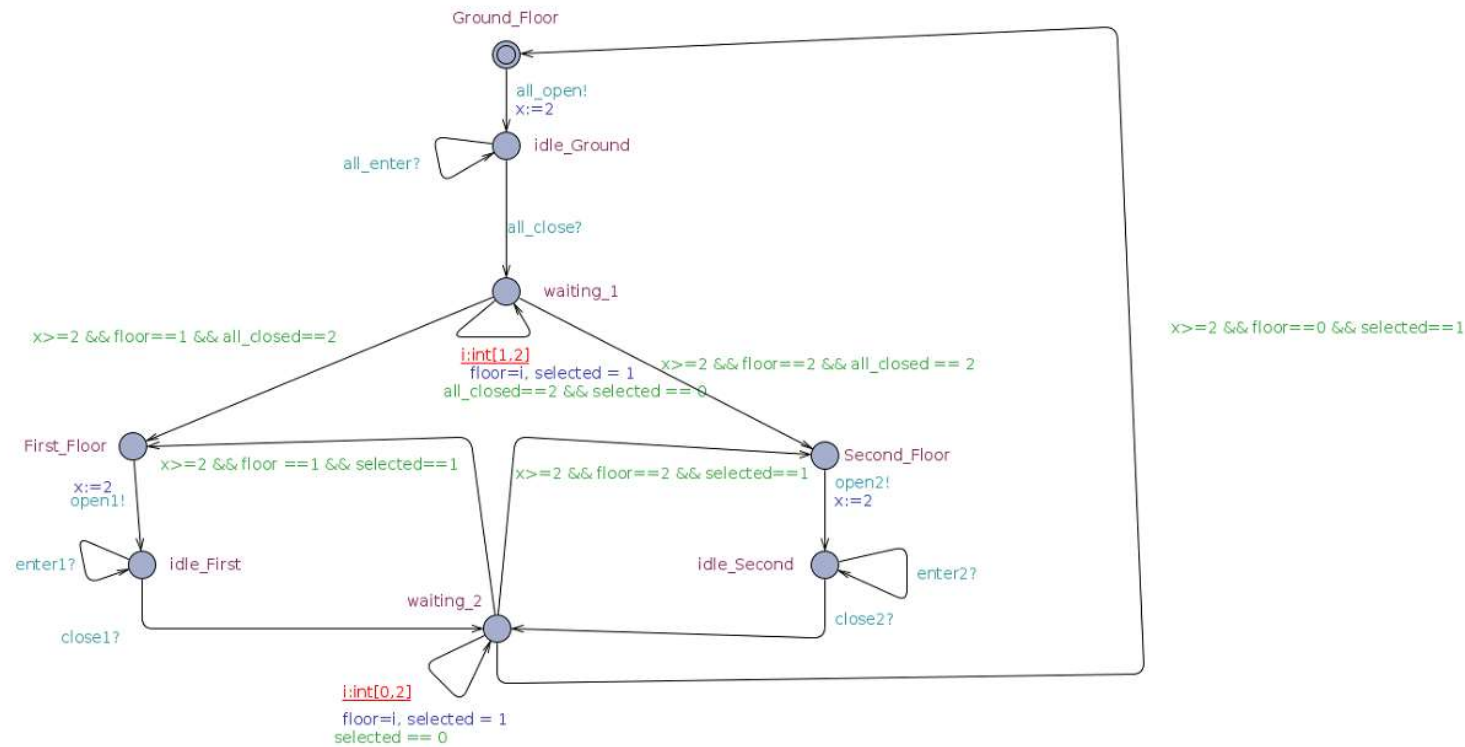
Door Template I.



Door Template II.

- ▶ Alaphelyzetben az ajtó zárva van
- ▶ A lift amikor a Földszinten található mind a két ajtót kinyítja. Ekkor kiküld a lift egy boardcast üzenetet `all_open` paranccsal
- ▶ Mind a két ajtó kinyílik. A `waiting` globális változó 3-as értéket kap. Ez a várakozó személyek számát szemlélteti.
- ▶ `All_closed` változó 0 (`false`) értéket kap. Ezt a változót fogjuk felhasználni a két ajtó földszinten történi bezárásának ellenőrzésére.
- ▶ Az ajtó kinyílt és megkezdődhet a személyek beszállása. (`all_enter, enter`)
- ▶ A beszálláskor elkezdjük csökkenteni a `waiting` változót. (várakozók száma)
- ▶ Az `X` globális változó értéke 0 lesz. A lift órája
- ▶ Attól függően hogy a lift melyik szinten van, végrehajtnak az alábbi szinkronizációk. (`all_close, close`)
- ▶ A amikor a lift a földszinten van az `all_close` fut le. Itt 3db őrfeltétel található. `Waiting == 0` (Mindenki beszált a liftbe), `x >= 4` a lift órája több mint 4, `floor == 0` a földszinten vagyunk
- ▶ Az feltételek mellett az alábbi `update`-ek futnak le. A lift órája 0-as értéket kap, az `all_closed` változót 1-el növeljük. Ez szemlélteti hogy a két ajtó között mennyi van becsukva. Illetve a `selected` változót 0 értéket kap. (Nincs még kiválasztva szint)

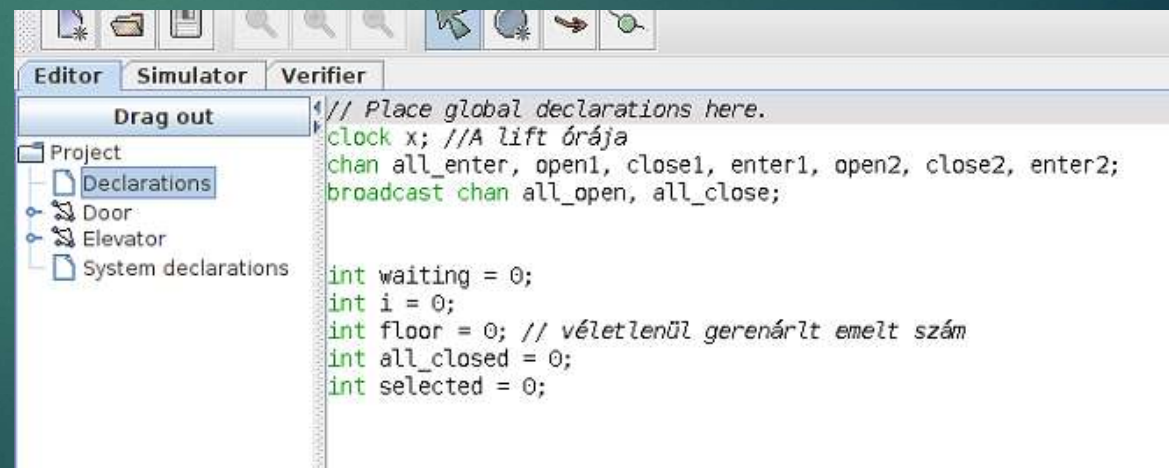
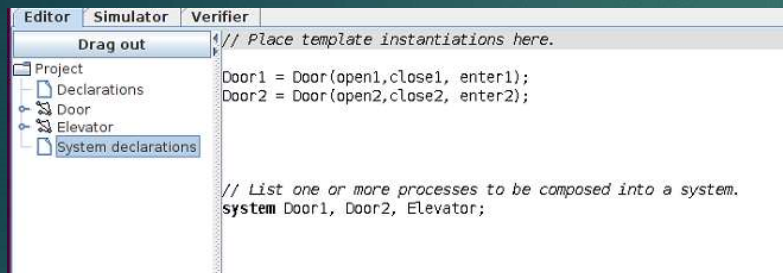
Elevator Template I.



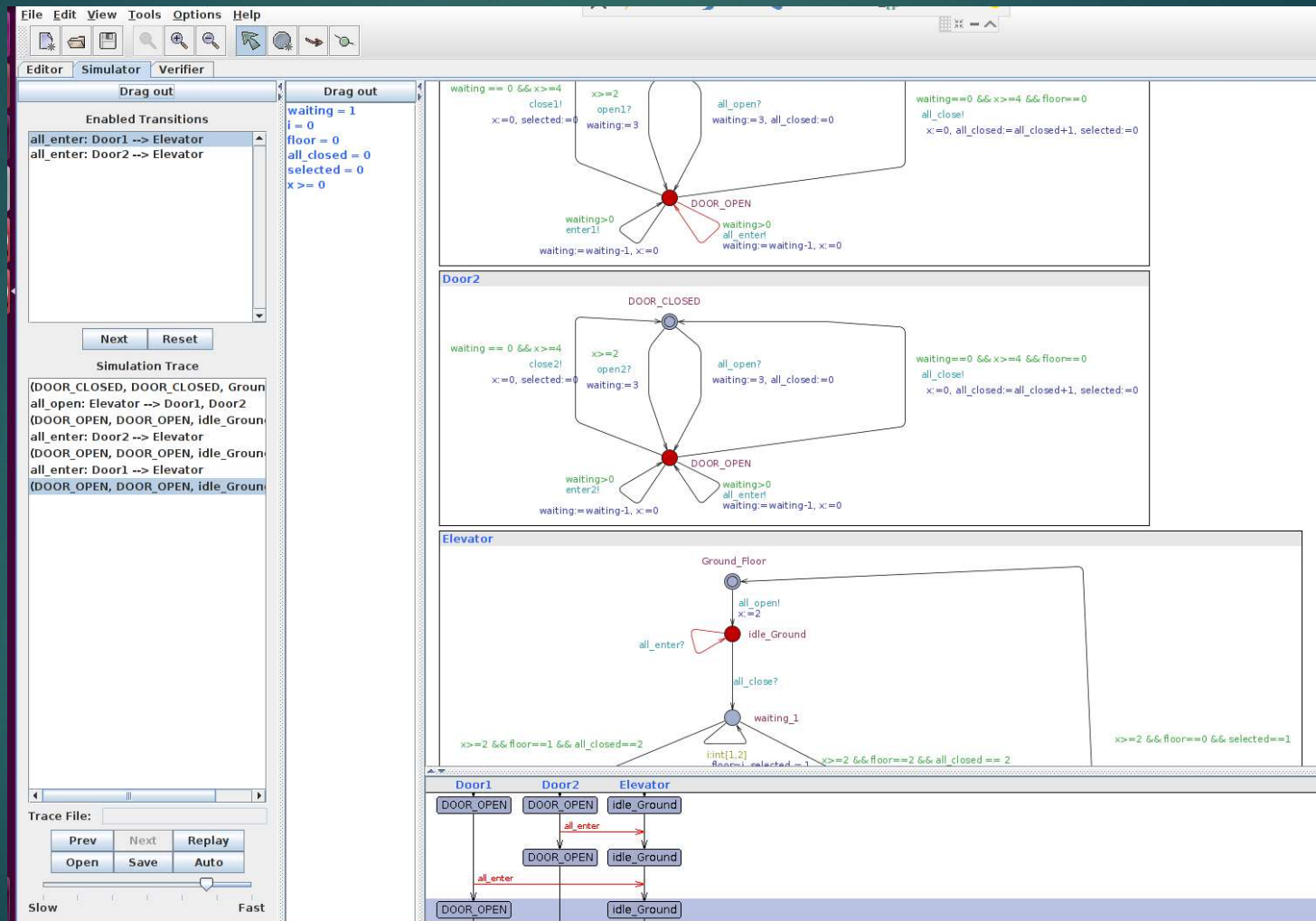
Elevator Template II.

- ▶ Alaphelyzetben a lift a földszinten van.
- ▶ Kiküldi a broadcast `all_open` szinkronizációs parancsot. Erre mind a két ajtó kinyílik.
- ▶ A lift órája 2-es értéket kap. Ennyi időt vesz igénybe az ajtó nyitás.
- ▶ Megtörténik a beszállás, majd az ajtók becsukása.
- ▶ Ezután a `waiting_1` vár hogy mind a két ajtó becsukodjon. `all_closed==2` és nincs még kiválasztva szint `selected ==0`
- ▶ `i:int[1,2]` segítségével generálunk véletlen számot 1-2 között. Ezzel szemléltetjük hogy a bent lévő utasok választottak egy szintet.
- ▶ A `selected` értéke 1 lesz. Tehát választottak szintet.
- ▶ Annak függvényébe hogy milyen szintet választotak tovább halad a lift.
- ▶ Az adott szint adott ajtóbecsukódása után ismét várakozó állapotba kerül a lift. Várja hogy szintet válasszanak.
- ▶ Lehetséges hogy ugyan azt a szintet választják ki ami az utolsó volt. Ekkor újra kinyílódik az ajtó és megtörténik a ki-be szállás.
- ▶ Amikor viszont a földszint hívódik meg mindeképpen kinyílik a két ajtó mert nem lehet pontosan tudni hogy az adott utasok melyik oldalon akarnak kiszállni.

Globális és rendszer deklarációk



Szimulációból egy részlet



Verifier

- ▶ $E \leftrightarrow (\text{Door1.DOOR_OPEN and Door2.DOOR_OPEN})$
 - ▶ Igaz, ha a rendszer valamikor elérhet egy olyan állapotot, amelyben Door1 és Door2 is DOOR_OPEN állapotba van
- ▶ $E \leftrightarrow \text{Door1.DOOR_OPEN and } x \geq 4$
- ▶ $A[] \text{ Elevator.First_Floor imply } x \geq 2$
 - ▶ Invariáns tulajdonság, valahányszor az Elevator First_Floor állapotba van, mindannyiszor x nagyobb vagy egyenlő mint 2
- ▶ $A[]$ not deadlock

The screenshot displays the UPPAAL Verifier interface. The 'Overview' tab is active, showing the following properties:

- $E \leftrightarrow (\text{Door1.DOOR_OPEN and Door2.DOOR_OPEN})$
- $E \leftrightarrow \text{Door1.DOOR_OPEN and } x \geq 4$
- $A[] \text{ Elevator.First_Floor imply } x \geq 2$
- $A[]$ not deadlock

On the right side of the Overview tab, there are three green circles indicating successful verification and a button labeled 'Check'. Below the Overview tab, the 'Query' tab shows the same $E \leftrightarrow (\text{Door1.DOOR_OPEN and Door2.DOOR_OPEN})$ query. The 'Comment' tab contains the text: 'Igaz, ha a rendszer valamikor elérhet egy olyan állapotot, amelyben Door1 és Door2 is DOOR_OPEN állapotba van'. The 'Status' tab at the bottom provides a detailed log of the verification process, including connection status and the final result for each property: 'Property is satisfied' for all three and 'Property is satisfied' for the deadlock check.