Szimuláció

Szimuláció

<u>Definíció</u>: A modell rendszerint bonyolult, részleteiben nem ismert rendszerek működésének megismerésére készített sematikus elképzelés, amelyből új összefüggésekre lehet következtetni, vagy amely alkalmas arra, hogy a rendszer jelenségei matematikailag leírhatók legyenek. A modell a valódi rendszereknek többnyire csak főbb tulajdonságait tükrözi, egyszerűsített formában.

MODELLEZÉS: A modell elkészítésének folyamata.

SZIMULÁCIÓ: A modell használatának folyamata.

Szimuláció

A modellalkotás első lépésében meg kell határozni a modellben szereplő objektumokat, amelyeket meg kell feleltetni a valós rendszer objektumainak (objektumai egy-egy osztályának).

Ez a megfeleltetés általában állapotaik megfeleltetését jelenti. Ahhoz ugyanis, hogy objektumokról külön-külön beszélhessünk, szükség van individuális létezésükre, amelyet állapotaik megadásával helyettesítünk.

Ezután következő feladat a rendszer állapotváltozását (az objektumok számának változását, állapotainak változását) leíró algoritmus elkészítése.

Szimuláció

A modelleket két nagy osztályba soroljuk.

Az egyikben a teljes jövőt előre kiszámítjuk (a kezdőállapotból), s azután csak a kiszámított jövő megjelenítése a feladat.

A másikban az aktuális állapotból csak a következő időegységbeli állapotot határozzuk meg, majd abból számítjuk a következőt ...

Amivel most foglalkozunk: sok elem mozog térben (síkban) egymástól függően, miközben állapotukat változtathatják.

Feladat: folyamatleírás – egy elemmel mi történhet?

Szimuláció

Megvalósítási lehetőség:

- időléptetés (minden időegységben történik mindenkivel)
  - elemenkénti vizsgálat
  - helyenkénti vizsgálat
- eseményléptetés (a következő esemény időpontjára lépünk és azt végrehajtjuk)

A párhuzamosság problémája: a valós világ párhuzamosságát a számítógép szekvenciális működésére kell átalakítani úgy, hogy az eseményeket a programbeli sorrend ne befolyásolja!

Feladat: folyamatleírás – egy elemmel mi történhet?

Szimuláció: havazás

Feladat: havazás

A szimulációs tér egy mátrix, ahova fentről hópelyhek lépnek be. A hópelyhek időegységenként egyet lépnek lefelé (egyszerre – azaz párhuzamosan). Ha alulra érnek, vagy már lent álló hópehely fölé érnek, akkor 3 jelenség történhet (az alábbi sorrendben):

- ha balra lefelé léphet, akkor oda lép;
- ha jobbra lefelé léphet, akkor oda lép;
- helyben marad.

Szimuláció: havazás

Megvalósítási lehetőségek:

- időléptetés: ez a természetes, a hópelyhek időegységenként lépnek egyet
    - elemenkénti vizsgálat: minden hópehelyre – mi történhet?
    - helyenkénti vizsgálat: minden helyre – mi történhet?
- eseményléptetés: lehetne esemény az, amikor a hópehely a végleges helyére kerül, de ezt nehéz előre kiszámolni.

Párhuzamosság megoldása: előbb léptetjük azt, aki a mozgásával másokat akadályozhat.

Szimuláció: havazás - hópelyhenként

Ábrázolás:

N,M – a tér mérete

DB – a hópelyhek száma

H(DB) – a hópelyhek sor- és oszlopkoordinátái

Párhuzamosság: ha a hópelyheket belépési idejük szerinti sorrendben vizsgáljuk, akkor az akadályozó előbb léphet, mint az akadályozott.

A belépés balról jobbra sorrendben történjen!

Szimuláció: havazás - hópelyhenként

```
Szimulációs lépés:
  Ciklus i=1-től DB-ig
    Ha H(i).sor<N akkor
      Ha szabad(H(i).sor+1,H(i).oszlop)
         akkor H(i).sor:=H(i).sor+1
      különben ha szabad(H(i).sor+1,H(i).oszlop-1)
         akkor H(i).sor:=H(i).sor+1
               H(i).oszlop:=H(i).oszlop-1
      különben ha szabad(H(i).sor+1,H(i).oszlop+1) és
                  szabad(H(i).sor+1,H(i).oszlop-1)
         akkor H(i).sor:=H(i).sor+1
               H(i).oszlop:=H(i).oszlop+1
  Ciklus vége
  Belépés az 1. sorba
Eljárás vége.
```

Kérdések: Kell 1-től?
Mekkora H tömb kell?

Szimuláció: havazás - hópelyhenként

```
szabad(sor,oszlop):
  j:=1
  Ciklus amíg j≤DB és
              nem(sor=H(j).sor és oszlop=H(j).oszlop))
    j:=j+1
  Ciklus vége
  szabad:=j>DB
Eljárás vége.
```

```
Belépés az 1. sorba:
  Ciklus j=1-től M-ig
    Ha van belépés akkor DB:=DB+1
                         H(DB).sor:=1; H(DB).oszlop=j
  Ciklus vége
Eljárás vége.
```

Szimuláció: havazás - helyenként

Ábrázolás:

N,M – a tér mérete

T(N,M) – a szimulációs tér, belépés az első sorba

T(N+1,M) – az első alatti sor kitöltve álló hópelyhekkel, így a hópelyhek megállása egységesen kezelhető

T(i,j)=0, ha nincs ott hópehely; 1, ha van ott hópehely.

Párhuzamosság: ha a teret alulról felfelé haladva vizsgáljuk, akkor aki akadályozhat, azt előbb mozgatjuk, mint azt, akit akadályoz.

Szimuláció: havazás - helyenként

```
Szimulációs lépés:
  Ciklus i=N-1-től 1-ig -1-esével
    Lefelé lépés az i. sorból
    Balra lefelé lépés az i. sorból
    Jobbra lefelé lépés az i. sorból
  Ciklus vége
  Belépés az 1. sorba
Eljárás vége.
```

A mozgás sorrendje a szabályok sorrendjének felel meg.

Kérdés: mi a teendő, ha van olyan hely, ahova egyszerre jönnének balról és jobbról is?

Szimuláció: havazás - helyenként

```
Lefelé lépés az i. sorból
  Ciklus j=1-től M-ig
    Ha T(i,j)=1 és T(i+1,j)=0
       akkor T(i+1,j):=1; T(i,j):=0
  Ciklus vége
Eljárás vége.
Balra lefelé lépés az i. sorból
 Ciklus j=2-től M-ig
  Ha T(i,j)=1 és T(i+1,j-1)=0
     akkor T(i+1,j-1):=1; T(i,j):=0
 Ciklus vége
Eljárás vége.
```

Szimuláció: havazás - helyenként

```
Jobbra lefelé lépés az i. sorból
 Ciklus j=1-től M-1-ig
  Ha T(i,j)=1 és T(i+1,j+1)=0
     akkor T(i+1,j+1):=1; T(i,j):=0
 Ciklus vége
Eljárás vége.
```

```
Belépés az 1. sorba:
  Ciklus j=1-től M-ig
    Ha van belépés akkor T(1,j):=1
  Ciklus vége
Eljárás vége.
```

Közlekedés szimuláció

Egy utat középen egy gyalogosátkelő két szakaszra oszt, a zebrához közlekedési lámpát helyeztek. Az útszakaszokat négyzetes cellákra osztjuk. N cella van a lámpa előtt, 1 cella a zebra, újabb N cella van a lámpa mögött. A mozgás szabályai:

- egy autó egy időegység alatt egy cellával mozdulhat el

Pl:

X		X			X		X	

→

	X		X			X		X

- egy útszakaszon két autó között mindig kell lenni legalább 1 üres cellának (akkor is, ha sűrűbben érkeznének)

Közlekedés szimuláció

- a közlekedési lámpa periodikusan váltakozik piros és zöld között, piros lámpaállásnál autó nem léphet a zebrára.

Pl:

X			X		X		X	

→

	X		X			X		X

Készíts programot, amely megadja, hogy az egyes autók melyik időpillanatban jutnak ki az útszakasz végén!

Megoldási elv:

- tömbök helyenként
- sorok autónként

Szimuláció: közlekedés szimuláció

Lámpa nélküli eset – 1 időegység lejátszása:

```
Útszimuláció1:
 Ciklus j=N-1-től 1-ig -1-esével
   T(j+1):=T(j)
 Ciklus vége
 Ha véletlenszám<Be akkor T(1):=1 különben T(1):=0
Eljárás vége.
```

```
Útszimuláció2(idő):
 Ha nem üres(S) és idő-első(S)>N akkor Sorból(S,x)
 Ha véletlenszám<Be akkor Sorba(S,idő)
Eljárás vége.
```

Szimuláció: közlekedés szimuláció

Lámpás eset – 1 időegység lejátszása:

```
Útszimuláció1:
 Ciklus j=N-1-től L-ig -1-esével
   T(j+1):=T(j)
 Ciklus vége
Ha zöldlámpa akkor T(L):=T(L-1) különben T(L):=0
 Ciklus j=L-2-től 1-ig -1-esével
   Ha T(j+2)=0 és T(j)=1 akkor T(j+1):=1
                      különben T(j+1):=0
 Ciklus vége
 Ha véletlenszám<Be és T(2)=0 akkor T(1):=1
                           különben T(1):=0
Eljárás vége.
```

Szimuláció: közlekedés szimuláció

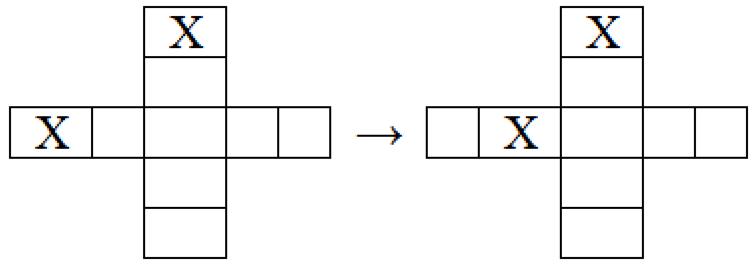
Lámpás eset – 1 időegység lejátszása:

```
Útszimuláció2(idő):
 Ha nem üres(S2) és idő-első(S2)>N-L akkor Sorból(S2,x)
 Ha nem üres(S1) és idő-első(S1)>L és elemszám(S2)<(N-L)/2
    akkor Sorból(S1,x); Sorba(S2,idő)
 Ha véletlenszám<Be és elemszám(S1)<L/2 akkor Sorba(idő)
Eljárás vége.
```

Közlekedés szimuláció

További feladatok:

- a kereszteződésben jobbkéz-szabály van: ha a kereszteződés előtti cellákba egyszerre lépne 2 autó, akkor a balról jövő léphet, a felülről jövő nem (kereszteződés a K pozíción)



Szimuláció: közlekedés szimuláció - helyenként

```
Útszimuláció1:
 Ciklus j=N-1-től K-1-ig -1-esével
   TV(j+1):=TV(j)
 Ciklus vége
 Ha TF(K-1)=1 akkor TV(K-1):=0 különben TV(K-1):=TV(K-2)
 Ciklus j=K-3-tól 1-ig -1-esével
   Ha TV(j+2)=1 akkor TV(j+1):=0 különben TV(j+1):=TV(j)
 Ciklus vége
 Ha véletlenszám<Be és TV(2)=0 akkor TV(1):=1
                              különben TV(1):=0
 … {függőleges út}
Eljárás vége.
```

Szimuláció: közlekedés szimuláció - sorokkal

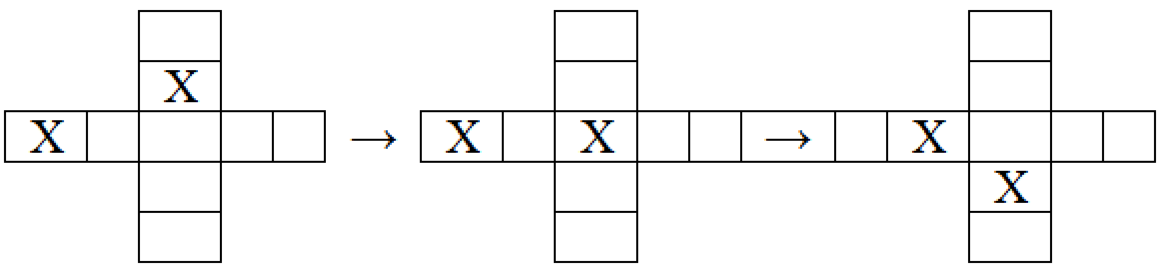
```
Útszimuláció2(idő):
 Ha nem üres(SV2) és idő-első(SV2)>N-K akkor Sorból(SV2)
 Ha nem üres(SF2) és idő-első(SF2)>N-K akkor Sorból(SF2)
 Ha nem üres(SV1) és idő-első(SV1)>K-1 és ker=0
    akkor Sorból(SV1); Sorba(SV2,idő); ker:=1
    különben ker:=0
 Ha véletlenszám<Be és hossz(SV1)<K/2 és
    utolsó(SV1)<idő-1 akkor Sorba(idő)
 … {függőleges út}
Eljárás vége.
```

4 sor kell a 4 útszakaszra, továbbá a kereszteződés kezelése.

Közlekedés szimuláció

További feladatok:

- ha a felülről jövő a kereszteződésbe lép, a balról jövőnek tartania kell az 1 cella távolságot.



Játéktábla

Egy játéktáblán a 0. időegységben L bábu van. Mindegyiket elindítjuk valamerre. Egy időegység alatt mindegyik a neki megfelelő irányba mozdul el, a tábla szélén mozgás irányukat az ellenkezőre változtatják. Lehetséges, hogy előbb-utóbb két bábu összeütközik: ugyanarra a helyre lépnének vagy átlépnének egymáson.

Készíts programot, amely megadja, hogy K időegységen belül mikor ütközik legelőször két bábu!

Szimulációs nyelv (GPSS)

Egy esztergapad működését kell modelleznünk, előtte a munkadarabok sorbaállnak, érkezés 10±3, feldolgozás 9±4 percig tart. Milyen az eszterga kihasználtsága, átlagos sorhossz, termékek keletkezési üteme,...?

```
SIMULATE
GENERATE 10,3
QUEUE SOR
SEIZE ESZTERGA
DEPART SOR
ADVANCE 9,4
RELEASE ESZTERGA
TERMINATE 1
START 1000
END
```

Szimuláció