

Клетъчни автомати - упражнения

доц. П. Рашков (ИМИ-БАН)

Симулация на горски пожар

С помощта на елементарен клетъчен автомат ще симулираме разпространението на пожар при зададени начални условия.

Ландшафтът в началния момент е разнообразен и се състои от няколко дървета, които горят, както и от незасегнати участъци с дървета или празни площи.

Пожарът в естествени условия може да се влияе допълнително и от гръмотевици, които предизвикват допълнителни запалвания на дървета, но ние засега няма да разглеждаме този фактор.

Клетъчен автомат

Ако представим ландшафта като решетка, чиито клетки съдържат най-много по едно дърво, можем да използваме стохастичен клетъчен автомат за симулацията.

Клетъчният автомат описва промените в състоянието на клетките:

- | | | |
|---|---------|---------------------------------------|
| 0 | EMPTY | клетката е празна |
| 1 | TREE | клетката съдържа дърво, което не гори |
| 2 | BURNING | клетката съдържа горящо дърво |

Така правим симулация на пожар в ландшафт, който се състои от дървета и празно пространство.

Дърветата могат да горят в подължение на единица време (1 стъпка) и могат да подпалват съседни дървета с някаква вероятност.

Клетъчният автомат е стохастичен.

Начални условия - инициализация

Необходимо е да инициализираме тези клетки с началните условия.

Тъй като автоматът е стохастичен, необходимо е да теглим набор от случайни числа, за да определим състоянието на всяка клетка.

Начални условия - инициализация

задаване на размера на $n \times n$ -мрежата, описваща ландшафта ($n > 0$)

две вероятности за инициализиране на състоянието на клетъчния автомат

- ▶ `probTree`: показва дали в дадена клетка има дърво (%дървета в ландшафта)
- ▶ `probBurning`: показва дали дадено дърво гори (% горящи дървета при започване на симулацията)

Псевдокод за инициализация

Със следния псевдокод ще се върне началното състояние на решетката на КА.

Матрицата `forest` записва състоянието на гората в дадения момент.

Може да изобразите началното състояние с помощта на графична програма.

Псевдокод за инициализация

върща състоянието на автомата в началния момент

```
initForest(N, probTree, probBurning)
```

алгоритъм:

```
forest <- matrix(, nrow = N, ncol = N) //forest = n x  
n grid
```

Използваме генератор за случайни числа от равномерното разпределение в (0,1). Теглим две случайно избрани числа.

Псевдокод за инициализация

```
for every cell in forest, DO
  if (random number1 < probTree) // tree at site
    if (random number2 < probBuring) // tree is burning
      assign BURNING to cell
    else // tree is not burning
      assign TREE to cell
    else // no tree at site
      assign EMPTY to cell
return forest
```

Локално правило

На всяка стъпка от симулацията трябва да приложим локално правило за разпространение на пожара. Така ще определим състоянието на всяка клетка от КА за следващата стъпка. Състоянието на клетката в следващата стъпка ще зависи от нейната стойност и от стойностите на съседните ѝ клетки в околност на фон Нойман.

Ако за дадена клетка на позиция $site \in forest$ означим съседните клетки с латинските букви N, E, S, W за север, изток, юг и запад, аргументите на функцията `spread` за локалното правило ще включват $site, N, E, S, W$, както и вероятността клетката да не се запали, ако състоянието ѝ е TREE.

Локално правило

на всяка стъпка трябва да определим дали едно здраво дърво ще се запази, т.е. дали сред съседите му има поне едно горящо дърво в околност на фон Нойман (север N, изток E, юг S, запад W)

ако клетката е EMPTY, тя остава EMPTY

ако клетката е BURNING, на следващата стъпка става EMPTY

ако клетката е TREE, трябва да проверим дали сред съседните ѝ клетки има BURNING

Локално правило

Необходимо е да зададем вероятност `probImmune`, която показва колко е вероятно дървото да издържи на пожара и да не се запали от съседно дърво

Използваме генератор за случайни числа от равномерното разпределение в $(0,1)$. Теглим едно случайно избрано число и го сравняваме с `probImmune`.

локално правило

за всяка клетка от мрежата forest, трябва да определим какво става, ако в нея има дърво

```
spread(site,N,E,S,W,probImmune)
if (site = TREE) AND (N, E, S, or W = BURNING)
if (random number < probImmune)
    return TREE
else
    return BURNING
```

гранични условия

граничните условия са необходими, за да определим какво се случва по края на мрежата
за наше улеснение допускаме, че клетките по ръба са празни(EMPTY)

Симулация на КА

С помощта на функция `fire` извършваме симулация на автомата за даден брой стъпки по време.

Входните аргументи са стойностите на параметрите: размерът на решетката `n`, вероятностите `probTree`, `probBurning`, `probImmune` и броят на стъпките по времето `t`.

Функцията `fire` трябва да върне списък `grids`, включващ решетката с началните условия и решетките, получени в следващите `t` стъпки по локалното правило `spread`.

Симулация на КА

Обърнете внимание, че функцията `apply` трябва да приложи `spread` за всяка клетка `site` от текущата решетка `forest`!

След това състоянието на `forest` се записва към масива `grids`.

Състоянията на решетките от масива `grids` могат да бъдат изобразени графично.

fire - псевдокод

```
n <- size of the square grid
probTree <- вероятност полето да е заето от дърво
probBurning <- вероятност дървото да гори в началото
probImmune <- вероятност дърва да не се запази
t <- брой стъпките
spread <- правило за промяна на състояние на поле на всяка
стъпка
forest <- initForest(N, probTree, probBurning)
//списък на състоянията на КА в различните стъпки
grids <- array(, dim=c(t, N, N))
```

fire - псевдокод

```
REPEAT t times
```

```
    forest <- apply(forest, probImmune) //APPLY spread TO  
EACH SITE
```

```
    grids[i, , ] <- forest
```

```
RETURN grids
```

Задачи

Направете симулация на автомата със слените стойности на параметрите:

$n = 50$,
 $\text{probTree} = 0.8$,
 $\text{probBurning} = 0.0005$,
 $\text{probImmune} = 0.25$,
 $t = 50$

Задачи

Променете локалното правило, като допуснете, че запалено дърво изгаря за две стъпки по времето.

Какво трябва да се промени в променливите за състоянието на клетката и във функцията `spread`?