**Exercícios computacionais 2: Autômatos Celulares**

**Entregar um** .nb **via Moodle com as respostas**

1. Verifique a documentação da função CellularAutomaton (e o tutorial relativo a Cellular Automata), observando suas parametrizações possíveis. Em particular, entenda a equivalência entre as seguintes formas:

* *Boolean functions*

(e.g., rule 90 as a pure Boolean function: Xor[#1,#3]& ,

or simply as BooleanFunction[90,3])

* *Explicit replacements for neighbourhoods*   
  (e.g., rule 90: {{1,\_,1}→0, {1,\_,0}→1, {0,\_,1}→1, {0,\_,0}→0}
* *Single “algebraic” replacement rule*

(e.g., rule 90: {{x\_,\_,y\_} :> Mod[x+y,2]})

* *Explicit functions*

(e.g., rule 90 as the algebraic function{Mod[#[[1]]+#[[3]],2]&,{},1})

1. Entender o conceito de *Second-order rules*, definidas como:

o próximo estado s­i no tempo *t+1* é função não só de (s­i-1, s­i, s­i+1) no tempo *t*, mas também de (s­i-1, s­i, s­i+1) no tempo *t-1*.

Em seguida responda: quantas funções locais (regras) existem?

OBS: A definição está na documentação da função CellularAutomaton.

1. Entender o conceito de regras confinadas (*Captive*) e analisar sua ocorrência no espaço elementar (i.e., quantas e quais são as regras?).

OBS: A definição se encontra no CAMat.nb.

1. Entender o conceito de regras com um estado de espalhamento (*Spreading*) e analisar sua ocorrência nos espaços elementar e de raio 1.5.

OBS: A definição se encontra no CAMat.nb.