

# On going Updates

**André Metelo** - metelo@gmail.com

**Christiano Braga** - cbraga@ic.uff.br

**Diego Brandão** - brandaodn@gmail.com

# Agenda

- Autômatos Híbridos
- Um Exemplo de Modelagem via Autômato Híbrido
- Implementação em Maude do Exemplo
- Execução do Exemplo

# Autômatos Híbridos

# O que é um Autômato Híbrido?

É um modelo dentro da teoria de autômatos que permite modelar, de forma precisa, sistemas em que processos computacionais interagem com processos físicos, e portanto analógicos por natureza.

Um autômato híbrido é uma "Finite State Machine" associado a um conjunto finito de variáveis que podem ser descritas através de equações diferenciais ordinárias - tipicamente variando no tempo.

# Formalmente Temos:

Definindo "*affine*" no contexto de um AH:

Dado as variáveis  $\{x_1, x_2, \dots, x_n\}$  um teste é dito affine se ele é da forma :  $a_1x_1 + a_2x_2 + \dots + a_nx_n \sim 0$ , e  $\sim$  pode ser:  $< \leq = > \geq$ .

Da mesma forma, uma atribuição é dita affine se ela é da forma:  $x_i = a_1x_1 + a_2x_2 + \dots + a_nx_n$

Sendo  $a_1, a_2, \dots, a_n$  constantes inteiras ou reais.

Um autômato híbrido AH, consiste de:

1. Um processo assíncrono P, que possui variáveis de estado contínuas que aparecem apenas em testes atribuições e atualizações "affine".
2. Um teste booleano, contínuo e invariante no tempo (CI), das variáveis de estado S que quando contínuas apenas aparecem em testes "affine".
3. Uma taxa de restrição (RC), que é um teste booleano "affine", que combina variáveis contínuas e suas derivadas assim como as variáveis discretas.

Note que as entradas, saídas, estados, inicialização e saídas do AH são as idênticas às do processo P.

Dessa forma, dado um estado  $s$ , e um intervalo de tempo  $\delta > 0$ , então:  $s \xrightarrow{\delta} s + \delta r$ , é uma ação de AH para um vetor  $r$ , composto de constantes  $r_x$  para cada variável contínua  $x$  se a RC é satisfeita para todas as variáveis  $x$ ; e se o estado  $s + tr$  satisfaz CI para qualquer  $t$  que satisfaz:  $0 < t < \delta$ .

# Alguns exemplos de CPS modelados por AH

- Termostato
- Cruise Control de Automóvel
- Piloto Automático de Avião

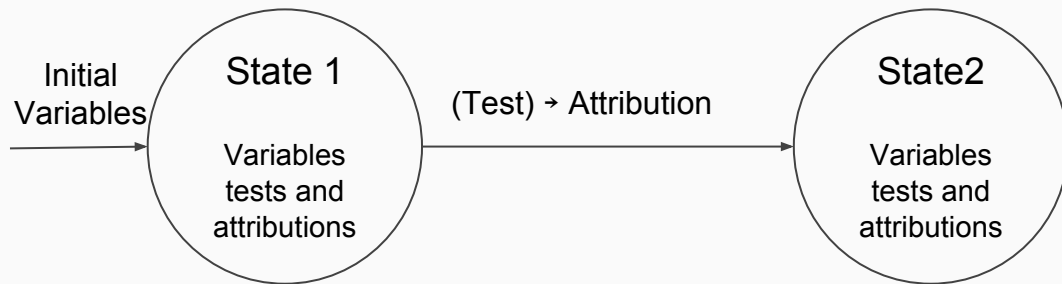
# Notação de um Autônomo Híbrido

Onde:

Cada círculo representa um estado definido no Autônomo

Uma seta sem conexão de origem a um estado representa um conjunto de entradas para o Autônomo

Uma seta conectando dois estados representa a transição de um estado para o outro caso o Teste (variável booleana) seja verdadeiro. Se a transição ocorrer então as variáveis listadas em Atribuições tem seus valores atualizados



# Notação de um Autônomo Híbrido (cont.)

- Tipicamente um estado tem um label significativo para o AH.
- Testes e atribuições são esperados de estar na forma "*affine*"
- Variáveis podem ser representadas pelo próprio label, ou sua derivada sendo denotada pela label com um ponto em cima
- Os Estados não precisam ser necessariamente representados por um círculo. Em geral é óbvio qual forma geométrica representa um estado



# Exemplo de Autônomo Híbrido

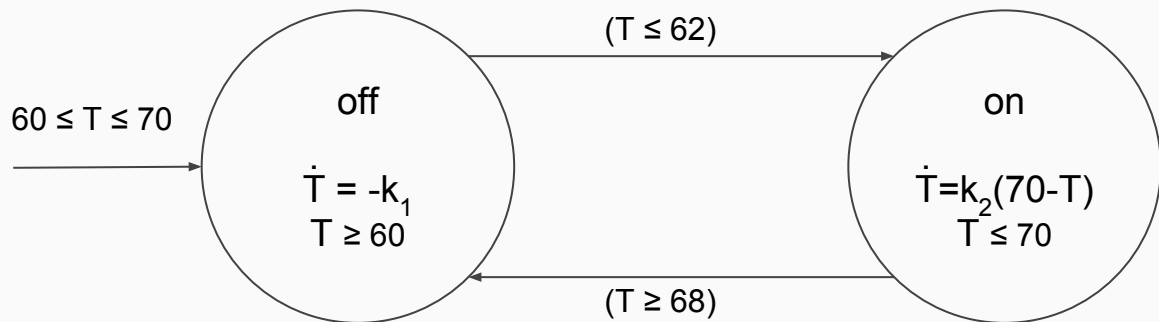
Termostato para manter Temperatura de uma sala entre 60F e 70F.

A entrada é um valor de  $T$  entre 60 e 70.

Enquanto o sistema está no estado "off" a sala perde temperatura a uma taxa constante.

Se a temperatura atinge 62, o sistema passa para o estado "on" onde a temperatura sobe via uma constante multiplicada pela diferença de 70 para  $T$ .

Se a temperatura atinge 68, o sistema passa para o estado "off" e o ciclo se repete



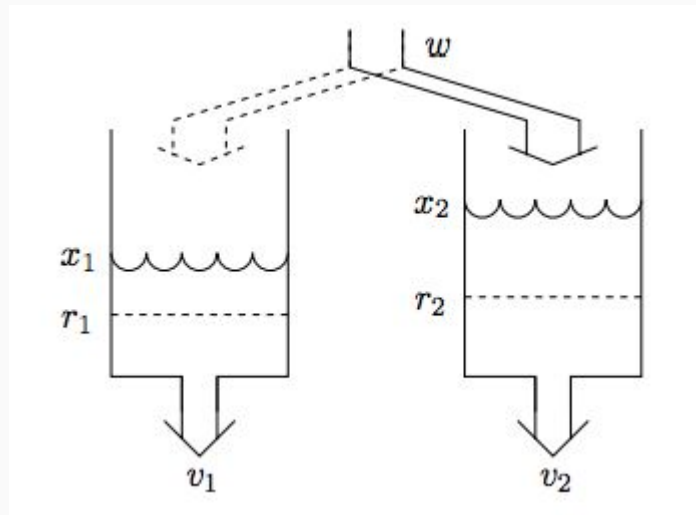
# Modelagem de um CPS via Autômato Híbrido

# Modelagem de Sistema de Controle de Volume de Água

- **Essa modelagem foi desenvolvida por Diego Brandão da CEFET/RJ**
- O autômato visa garantir que o volume de água em dois tanques, furados, acima de um volume mínimo através de uma mangueira que insere água no sistema.
- A mangueira pode ser movida de um tanque para outro instantaneamente.

# O Modelo

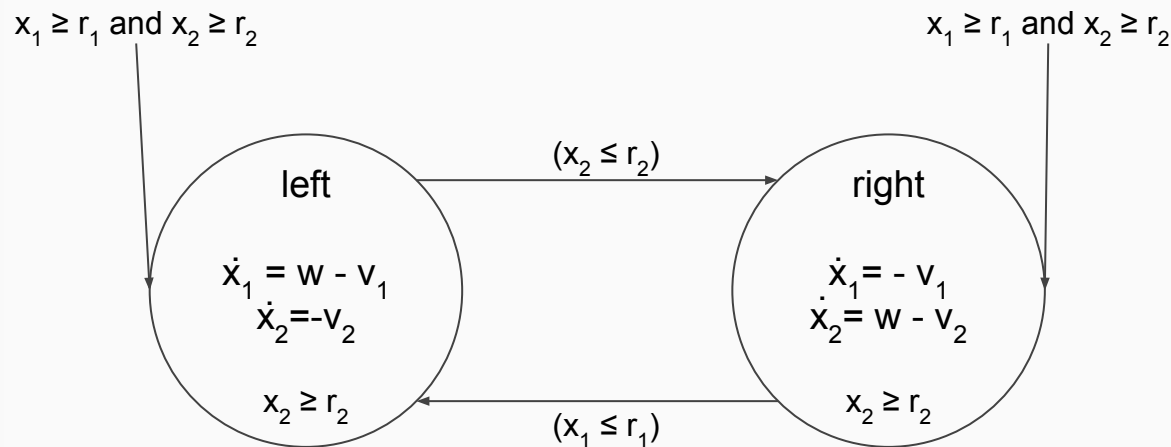
- Temos dois tanques vazando a uma taxa constante.
- Água é adicionada a uma taxa constante através de uma mangueira dedicada a um tanque em um dado momento no tempo.
- A qualquer instante a mangueira pode ser movida instantaneamente de um tanque para o outro



# O Autônomo Híbrido

Onde:

- $x_1$  e  $x_2$  representam os volumes de água no tanque 1 e 2
- $r_1$  e  $r_2$  são os limites mínimo da água nos tanques 1 e 2
- $w$  o fluxo de água adicionado pela mangueira
- $v_1$  e  $v_2$  são o volume de água escoados dos tanques 1 e 2



# A Implementação do Autômato em Real Time Maude

# Sobre Real-Time Maude

- **Real-Time Maude é uma extensão de Full Maude que permite evolução no tempo de sistemas.**
- Full maude, por sua vez, é uma extensão de Maude.
- Já Maude, é uma linguagem/sistema que suporte lógica de reescrita e equações com uso em diversas áreas.
  - Não é do escopo dessa apresentação discutir a linguagem maude

# RT Maude Source para o Sistema de Reservatórios

```
(tmod RESERVOIR is
  protecting POSRAT-TIME-DOMAIN .*** Dense time domain

  sort Hose .
  ops left right : -> Hose [ctor] .
  ops w v1 v2 r1 r2 : -> NNegRat .
  op _'\, _'\, _ : Hose NNegRat NNegRat -> System [ctor] .

  vars x1 x2 : NNegRat .
  var R : Time .
  crl [moveright] : left,x1,x2 => right,x1,x2
                    if x2 <= r2 .
  crl [moveleft] : right,x1,x2 => left,x1,x2
                    if x1 <= r1 .

  crl [tick-right] :
    {right, x1, x2} =>
    {right, x1 - (v1 * R), x2 + ((w - v2) * R)} in time R
    if x1 > r1 [nonexec] .

  crl [tick-left] :
    {left, x1, x2} =>
    {left, x1 + ((w - v1) * R), x2 - (v2 * R)} in time R
    if x2 > r2 [nonexec] .

endtm)
```



# Executando o Código do Exemplo

# Verificação Formal: Search

```
Timed search in TEST
{right,30,30} =>* {S:System}
in time <= 4 and with mode default time increase 1 :
```

```
Solution 1
```

```
    S:System --> right,30,30 ; TIME_ELAPSED:Time --> 0
```

```
Solution 2
```

```
    S:System --> right,25,35 ; TIME_ELAPSED:Time --> 1
```

```
Solution 3
```

```
    S:System --> right,20,40 ; TIME_ELAPSED:Time --> 2
```

```
Solution 4
```

```
    S:System --> right,15,45 ; TIME_ELAPSED:Time --> 3
```

```
Solution 5
```

```
    S:System --> left,15,45 ; TIME_ELAPSED:Time --> 3
```

```
Solution 6
```

```
    S:System --> left,20,40 ; TIME_ELAPSED:Time --> 4
```

```
No more solutions
```

# Verificação

## Formal: Execução até um tempo X finito

```
Timed rewrite {right,30,30} in RESERVATORIO with mode  
default time increase 1 in time <= 12345
```

```
Result ClockedSystem :  
  {right,45,15} in time 12345
```

Update de 2017-10-04

# Implementação do *n-reservoir* em Maude

- A generalização do problema dos reservatórios foi Implementado
  - Permitia um número finito de reservatórios que é determinado durante a execução
- A solução não era elegante e limitada
  - Forçava reservatórios com mesma características físicas
  - Código confuso
    - As sorts não estavam bem definidas
    - Muitas recursões nas funções
- Utilizava a mesma representação do modelo de 2 reservatórios

# Christiano ao Resgate

- Nova versão com código utilizando as capacidade de Maude de melhor maneira
  - Uso "pattern match" do Maude para reduzir recursão
  - Nova codificação para o sistema de reservatórios
    - Características individuais para os reservatórios
    - `{ hose(10, 0) < 0 | thr: (15, 50), hth: 30, rte: 5 > < 1 | thr: (15, 50), hth: 30, rte: 5 > }`
  - Organização das sorts
- Inclui implementação de Model Checker para os reservatórios
- Disponível em: <http://mycioxp.gotdns.org:8081/OpenSource/Reservoir.git>

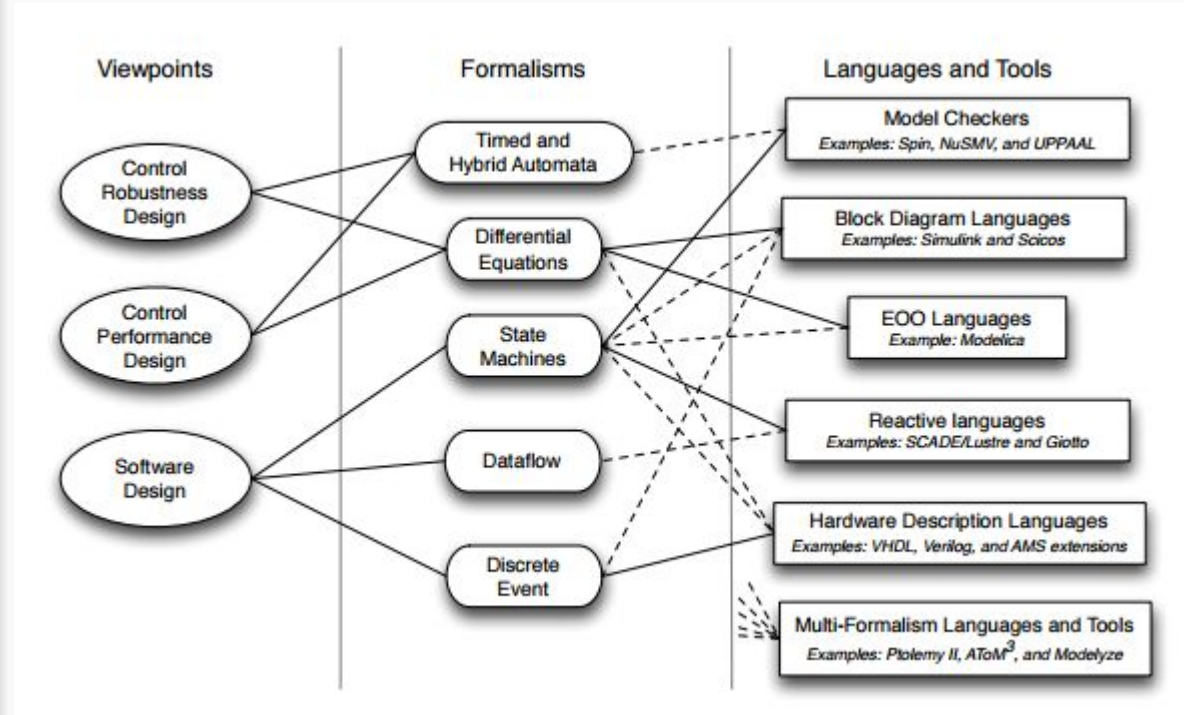
# Preparando um Artigo Sobre o Trabalho

- Apresentar o modelo como um tutorial de Maude para modelagem de CPS
  - Através de LHA + Maude
- Início do trabalho de identificar, qualificar e quantificar diferentes métodos de especificação, design e model check de CPS.
  - Idealmente servirá como base para um processo unificado para "projeto" de CPS

# Vista dos Métodos de "projeto" de CPS

Em: *Viewpoints, Formalisms, Languages, and Tools for Cyber-Physical Systems*,  
David Broman et al em 2012

Esse não é um grafo extensivo das opções





# Próximos Passos

- Finalizar o artigo
- Modificar o modelo do n-reservatórios para possuírmos uma mangueira com latência na mudança de um reservatório para outro.
- Identificar outros problemas para serem modelados com outros métodos
  - Water Boiler
  - Robo seguindo um path

Update de 2017-10-26

# Agenda

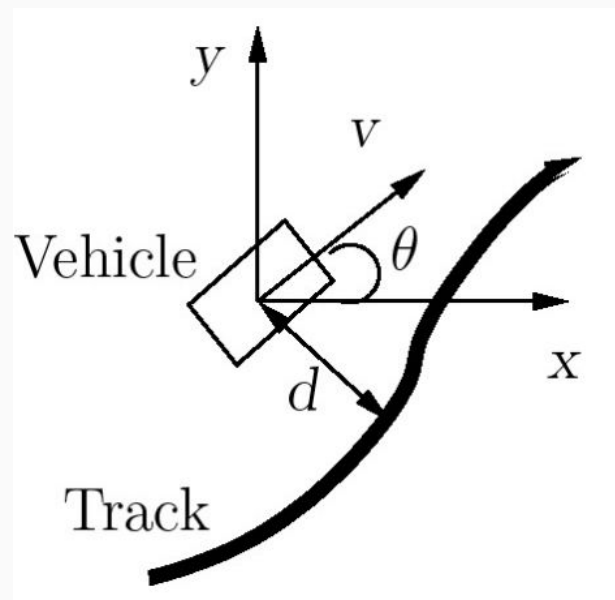
- Automated Guided Vehicle como um CPS
- Extensão do modelo de 2-reservatórios com mangueira com latência na mudança de um reservatório para outro.

# Modelagem de Automated Guided Vehicle

- **Essa modelagem foi apresentada por Alur em Principles of Cyber Physical Systems**
- O autômato visa garantir que um carrinho nunca se distancie mais que uma distância pré-determinada de um caminho.
  - O carrinho não conhece o caminho
- Não é escopo do modelo a tecnologia usado pelo carrinho para identificar a distância ao caminho.
  - É assumido q é possível medir a distância do ponto atual para o caminho

# O Modelo

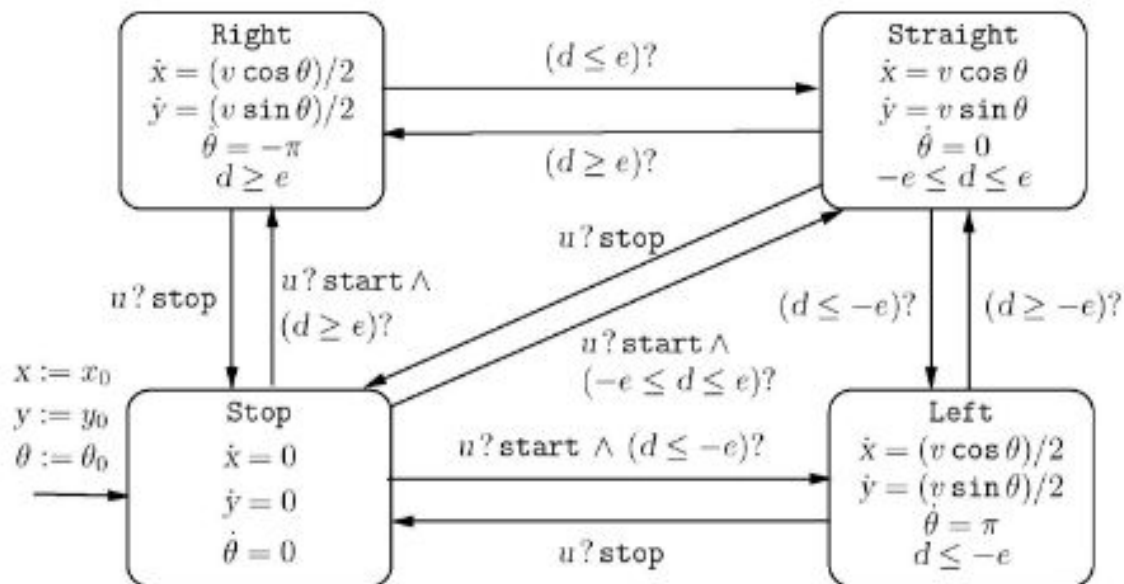
- O carro é tratado como um corpo rígido em movimento num plano
- O carro pode mover em linha reta com velocidade  $v$  ou rodar um ângulo  $\omega$  e mover  $v/2$ .
- O par  $(x,y)$  representa a posição do carrinho em relação ao eixo de referência que também é usado pelo caminho definido para o carro seguir.  $\theta$  representa o ângulo de inclinação do carrinho em relação ao eixo  $X$ .
- O tempo para executar a rotação e mudança de velocidade é instantâneo.



# O Autônomo Híbrido

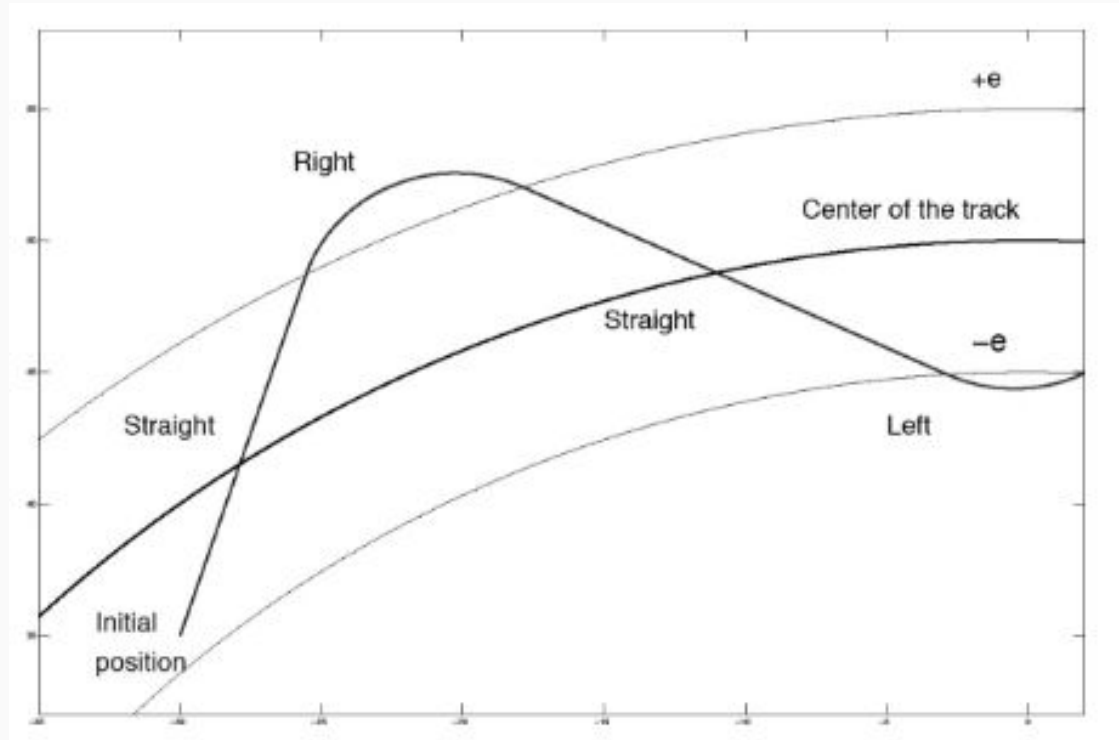
Onde:

- $(x,y)$  - posição do carro no plano
- $\theta$  - ângulo do carrinho e o eixo X.
- $v$  velocidade linear do carrinho
- $d$  - distância do carrinho para o caminho
- $e$  - distância máxima desejada da pista.
- $u$  - estado do motor do carrinho (start/stop)



# Caminho Esperado

- Dado um path, e os parametros do carrinho, é esperado um comportamento do seguinte tipo para o modelo:
- Note que no pior caso, podemos ter  $d = e + v$



# Implementação em RT-Maude

- E um autômato mais complexo que o 2-reservatórios:
  - 4 estados
  - 10 transições
  - Mais variáveis para controlar.
- Versão 0.1 feita, ainda precisa ser modelada e model-checked.
  - Em processo de execução



# Extensão do modelo de 2 reservatórios

- **O objetivo é modelar o problema de 2 reservatórios com uma mangueira que não move instantaneamente.**
  - A mangueira passa a ter mais estados.
  - identificar um sistema bem formado (com a vazão de entrada = a soma das vazões de saída) fica mais complicado devido a períodos em que água não entra no sistema.
- Criamos 2 novos estados para a mangueira:
  - *movingleft*
  - *movingRight*
  - Esses estados acontecem instantaneamente. Daí duas novas regras temporais transicionam para *left* e *right* e fazem a drenagem dos reservatórios.

# Próximos Passos

- Finalizar o artigo
- Modificar o modelo do n-reservatórios para possuírmos uma mangueira com latência na mudança de um reservatório para outro.
- Completar o check model do Carrinho.
- Identificar outros problemas para serem modelados com outros métodos
  - Water Boiler

# Obrigado!

Perguntas?

