

# SPIN Case Study

## Automatic Greenhouse

Di Marco, Okwieka

University of Rome "La Sapienza"

July 15th, 2023

# Why SPIN?

NuSMV is better suited for large but regular systems like hardware

SPIN only has explicit LTL model checking

Promela has more language features (e.g. channels)

Promela has more similarities to classical programming (conditions, loops)

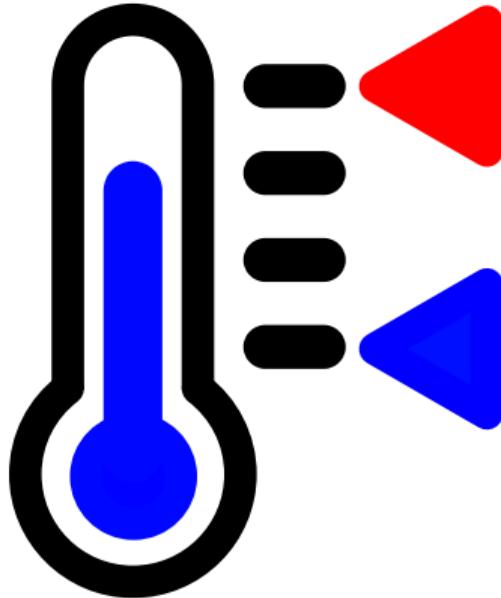
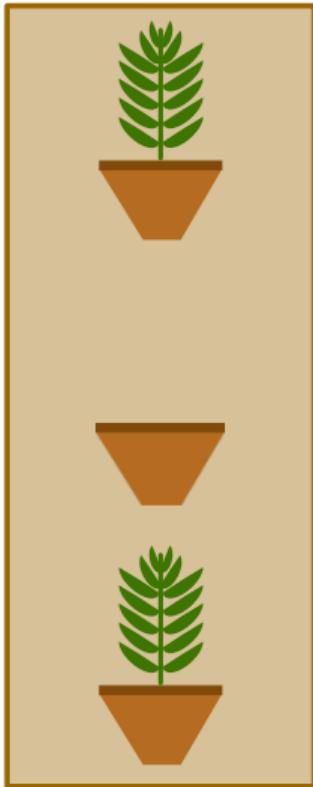
Promela has less focus on state transitions

# Greenhouse Model

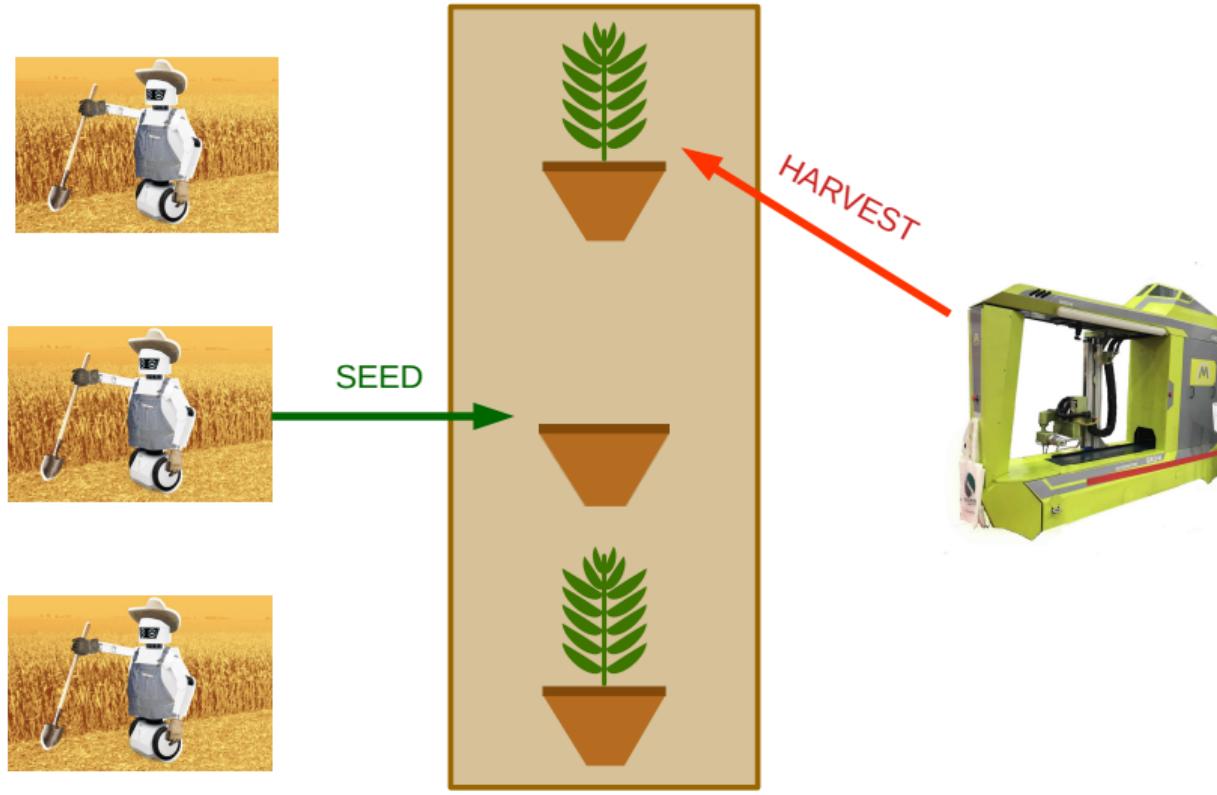
# Global overview: time and weather



# Plant Beds and Temperature



# Seeder and Harvester Bots



# SPIN Model

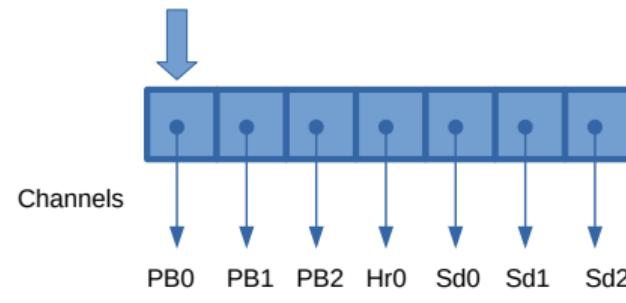
# Global Clock

Need a global time, all processes must be in lockstep

Global array of unbuffered channels, one per process

Clock process cycles through and sends TIME messages

Reading the message and the process's action are atomic



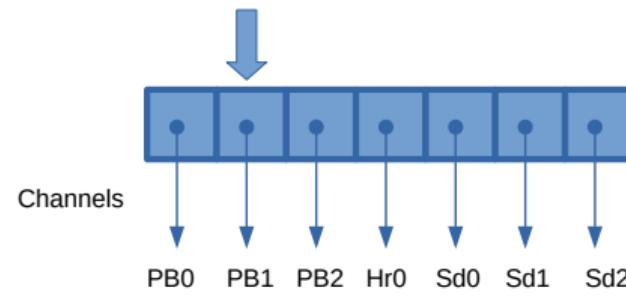
# Global Clock

Need a global time, all processes must be in lockstep

Global array of unbuffered channels, one per process

Clock process cycles through and sends TIME messages

Reading the message and the process's action are atomic



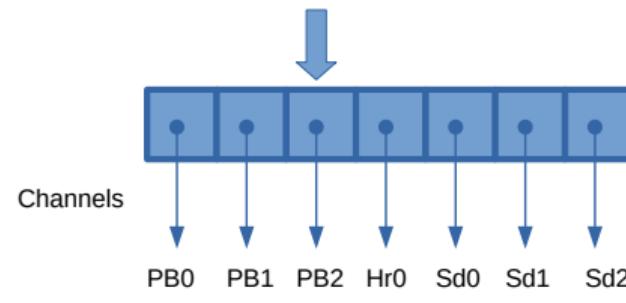
# Global Clock

Need a global time, all processes must be in lockstep

Global array of unbuffered channels, one per process

Clock process cycles through and sends TIME messages

Reading the message and the process's action are atomic



# Global Weather

Process with an internal **weather duration** decrementing each timestep

When this counter reaches 0, randomly choose (within parameters)

- **weather duration**
- constant **weather effect** on temperature (`weather_temp_rate`)

Currently only cold weather, as the greenhouse cannot counter heat (no cooling)

# Plant Bed

Process that keeps the age of each plant. Can be empty (AGE\_NO\_PLANT).

The plant is grown each timestep if it has not **expired**.

If it is too **hot** or too **cold**, the plant expires immediately.

The plant state is published every timestep to a global array (plant sensors).

The plant bed checks whether it received a **plant action** message (SEED, HARVEST) and changes its state accordingly.

# Harvester

Has a number of **Plant Beds** assigned to it based on index.

Reads its plant beds' sensors to find the age.

Policy: Harvests the plant with the **highest age** that is also **mature**.

Can only harvest **one plant per timestep**.



# Seeder

Has a number of **Plant Beds** assigned to it by index.

Finds first empty plant bed based on plant sensor array.

Can plant **one seed per timestep**.



# Heat Manager

Process that updates the greenhouse's **temperature** each timestep.

The temperature change depends on weather and whether the heater is turned on.

The heat manager also turns the **on** or **off** upon reaching safety thresholds.

# Monitor and Properties

# Expired

- At every time step the monitor checks the **number of expired plants harvested**.
- If ( $n_{\text{expired\_harvested}} > \text{LIMIT\_EXPIRED}$ ) the property has been violated.

# Temperature

- At every time step the monitor checks the **temperature**.
- If  $\text{temp} \in (\text{FREEZING\_TEMP}, \text{BURNING\_TEMP})$  the property is satisfied.

# Rate

- At every time step the **rate** of expired plants harvested.
- If  $\left( \frac{n\_mature\_harvested + n\_expired\_harvested}{n\_expired\_harvested} > RATE\_EXPIRED \right)$  the property is satisfied.

# Timed Properties

- Every mentioned property has a *timed* variant.
- The property is checked only for the first TIME\_LIMIT steps.
- After that point, it is set as satisfied.

# Fairness

- Most violations happened because of continuous harsh temperature.
- SPIN only supports **weak fairness**.
- We can enforce fairness **inside the LTL formula**.
- Every mentioned property has a *fair* variant.
  - The property is checked only for fair execution.
  - The weather can't always be the *worst* and it can't always last for the *longest time*.

# Results

# Lightweight Simulation

*Simulation time!*



# Heavyweight Simulation

The screenshot shows the iSpin 6.5.2 interface with a session titled "greenhouse.pml". The main window displays a state transition graph and a log of simulation events. The left sidebar contains icons for activities like Activities, Edit, View, Simulate / Replay, Verification, Swarm Run, Help, Save Session, Restore Session, and Out.

**Session Information:**  
22 Jun 20:08 Spin Version 6.5.2 -- 30 May 2023 :: iSpin Version 1.1.5 -- 28 May 2021

**Configuration:**

- Safety:** safety (checkbox checked), Invalid endstates (deadlock) (checkbox checked), assertion violations (checkbox checked), + rx/xs assertions (checkbox unchecked).
- Storage Mode:** exhaustive (checkbox checked), + minimized automata (slow) (checkbox unchecked), + collapse compression (checkbox checked), hash-compact (checkbox checked), bilstate/supertrace (checkbox unchecked).
- Search Mode:** depth-first search (checkbox checked), + partial order reduction (checkbox unchecked), + collapse compression (checkbox checked), bounded context switching with bound: 0 (checkbox checked), + iterative search for short trail (checkbox unchecked), breadth-first search (checkbox unchecked), + partial order reduction (checkbox checked), report unreachable code (checkbox checked).
- Show Error Trapping Options:** Save Result In: pan.out
- Remove Advanced Parameters:**
  - Physical Memory Available (in Mbytes): 4000 (checkbox checked), explain (checkbox checked)
  - Estimated State Space Size (states x 10<sup>3</sup>): 1000 (checkbox checked), explain (checkbox checked)
  - Maximum Search Depth (steps): 1000000 (checkbox checked), explain (checkbox checked)
  - Nr of hash-functions in Bilstate mode: 3 (checkbox checked), explain (checkbox checked)
  - Size for Minimized Automaton: 10000 (checkbox checked), explain (checkbox checked)
  - Extra Verifier Generation Options: (checkbox checked), explain (checkbox checked)
  - Extra Compile-Time Directives: DNFAIR=40 -DVECTORS (checkbox checked), explain (checkbox checked)
  - Extra Run-Time Options: (checkbox checked), explain (checkbox checked)

**Log Output:**

```
Depth: 999999 States: 3e+06 Transitions: 3.03e+06 Memory: 3694.7421= 24.8 H= 1e+05
pan: reached -DMEMLIMIT bound
4.1942e+09 bytes used
400000 bytes more needed
4.1943e+09 bytes limit
Hint: To increase memory resources with
-DCOLLAPSE #collapse, +compression, or
-DMA=2404 #better/slower compression, or
-DHC #hash-compaction, approximation
-DBITSTATE #supertrace, approximation

(Spin Version 6.5.2 -- 30 May 2023)
Warning: Search not completed

Full statespace search for:
never claim + (t_exp)
assertion violations + (if within scope of claim)
acceptance cycles + (states enabled)
invalid end states (disabled by never claim)

State-vector 2404 byte, depth reached 999999, errors: 0
32608171 states, stored
32608171 states, matched
32603200 states, stored+matched)
11259767 atomic steps
hash conflicts: 13802 resolved

Stats on memory usage (in Megabytes):
7562.930 memory used for states (stored) (State-vector + overhead)
3827.789 actual memory usage for states (compression: 50.61%)
state-vector as stored = 1203 bytes + 28 byte overhead
128.000 memory used for hash table (-w24)
53.406 memory used for DFS stack (-m1000000)
9.272 memory lost to fragmentation
3999.918 total actual memory usage

pan: elapsed time 30.6 seconds
No errors found -- did you verify all claims?
```

*Fin.*