CIS 721 - Real-Time Systems

Lecture 34 – Final Project Review

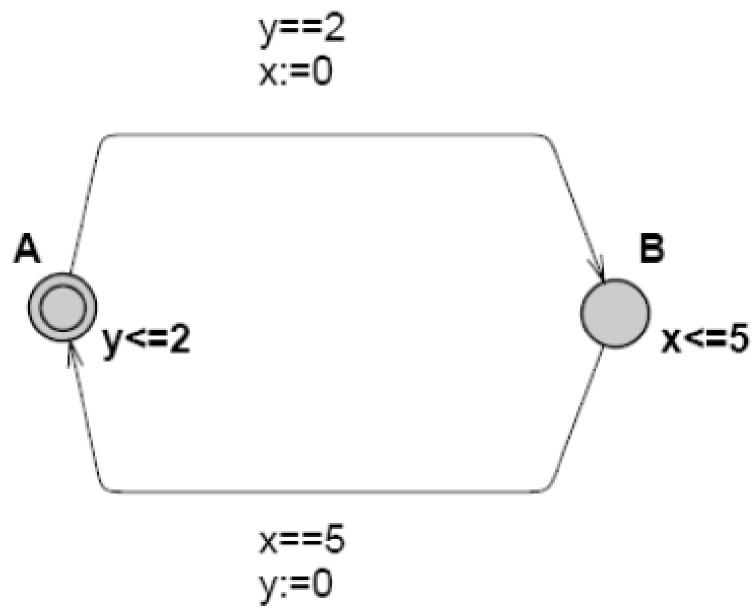
Mitch Neilsen

neilsen@ksu.edu

Outline

- Quiz #2: Wed., Dec. 2
  - Open book, open notes
- Final Project Presentations, Dec. 4, 7, 9, 11
  - 10-15 minutes with time for a few questions
  - Up to 4 presentations per day
  - Include real-time model + verificaction

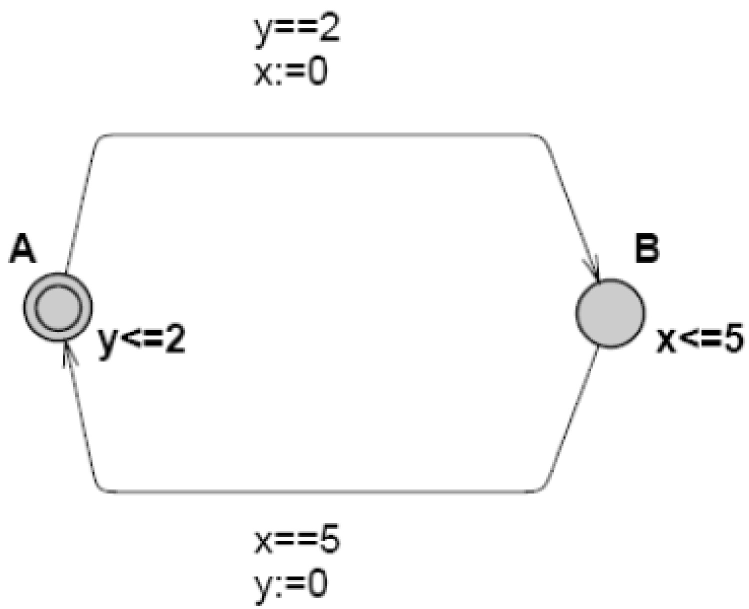
2. Consider the following timed automaton P:



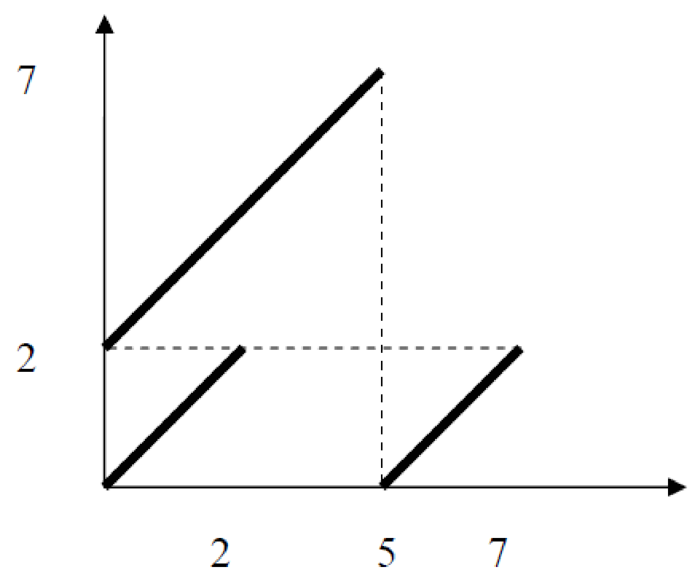
As this automaton has two clocks, x and y, the reachable state space with respect to the clocks can be viewed as a point in a two-dimensional Cartesian plane, one axis for clock x and one axis for clock y. A point (d, e), with non-negative d and e, can be used to denote that clock x equals d and clock y equals e.

(a) Determine the reachable state space for this automaton; e.g., draw a 2-d graph below:

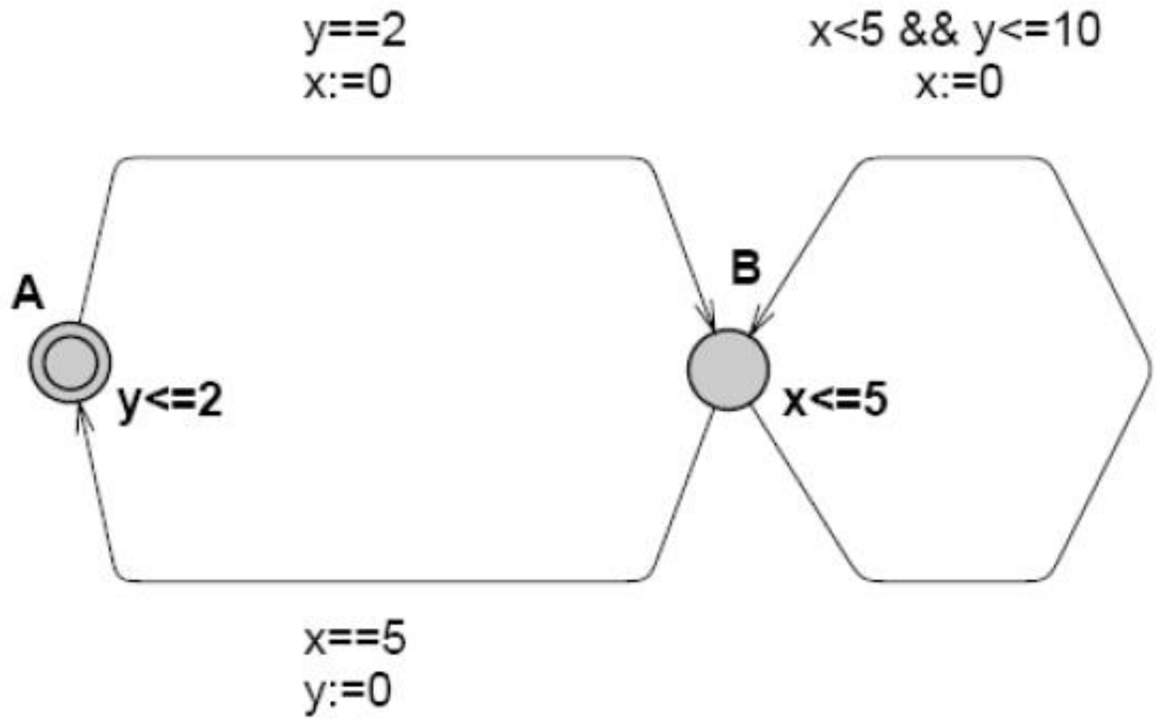
2. Consider the following timed automaton P:

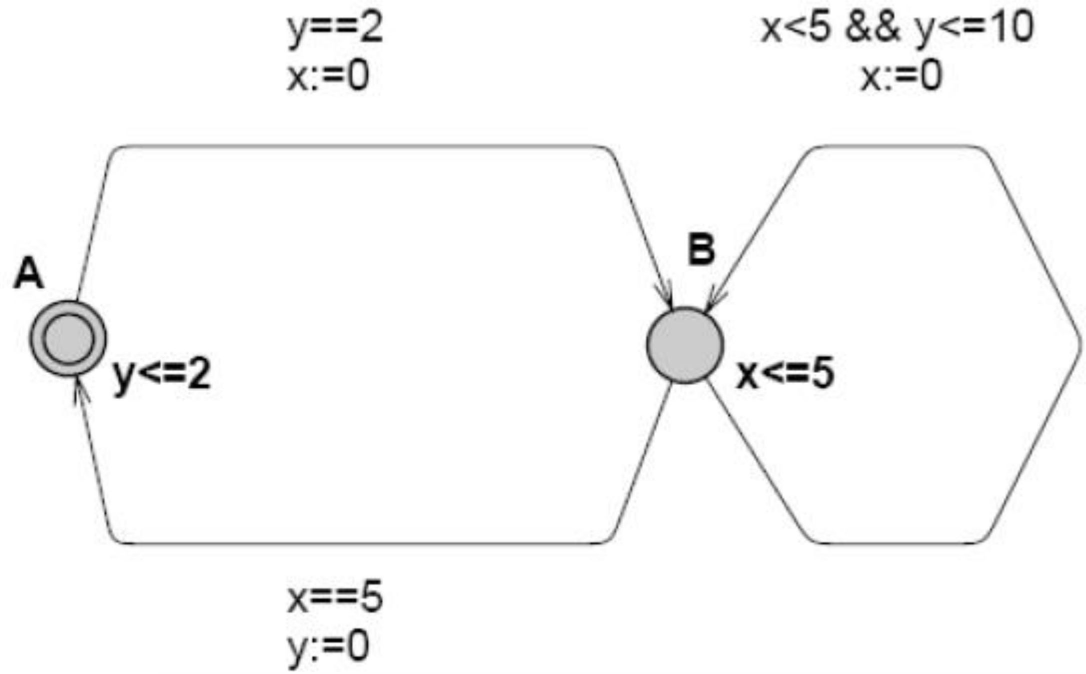


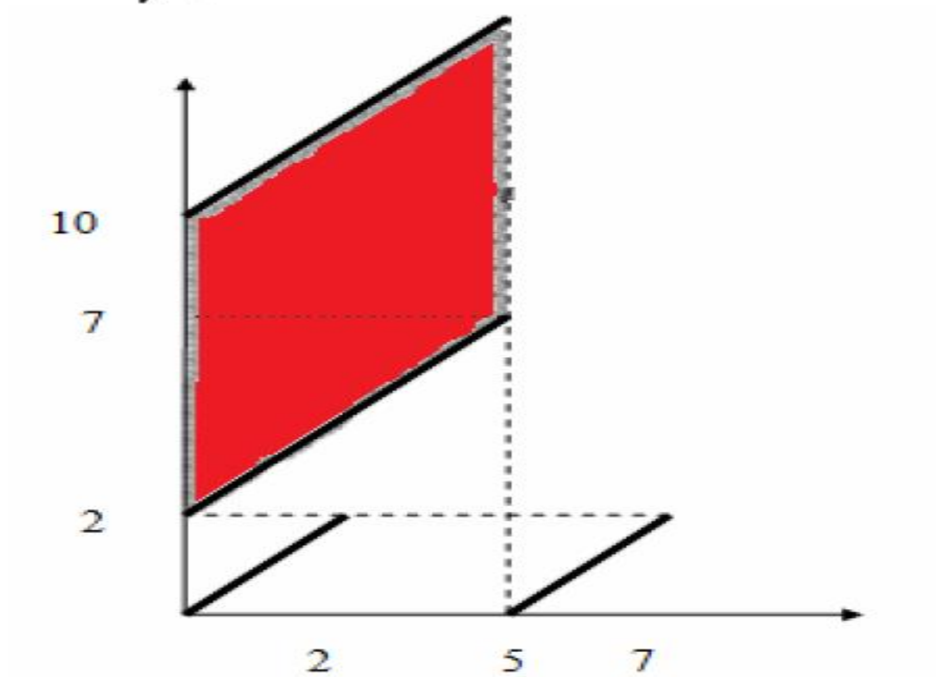
(a) Determine the reachable state space for this automaton; e.g., draw a 2-d graph below:

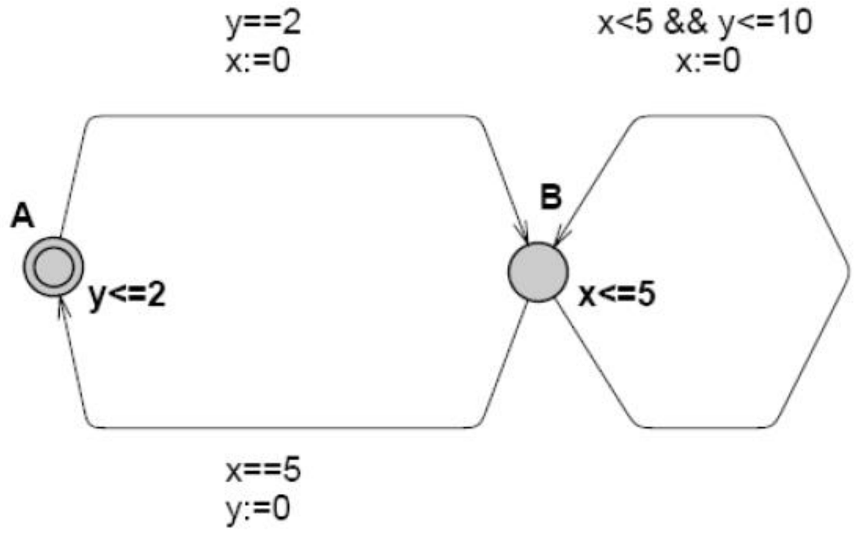


(b) Determine the reachable state space for this automaton for clocks x and y.



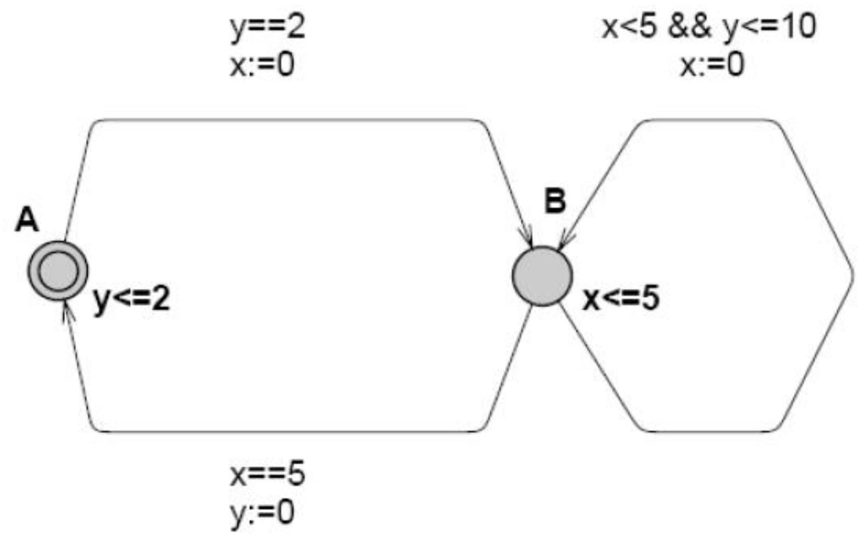

y==2
x:=0
x<5 && y<=10
x:=0
A
B
y<=2
x<=5
x==5
y:=0


10
7
2
2
5
7



c) Determine which of the following properties are satisfied for the automaton shown above in part (b):

- **`A[] not deadlock`**
- **`P.A --> P.B`**
- **`P.B --> P.A`**
- **`E<> (x==5 and y==7)`**
- **`E<> (x==5 and y==3)`**
- **`E<> (x==2 and y==12)`**



(c) Determine which of the following properties are satisfied for the automaton shown above in part (b):

- **A[] not deadlock (RED = satisfied)**
- **P.A --> P.B**
- P.B --> P.A
- **E<> (x==5 and y==7)**
- E<> (x==5 and y==3)
- **E<> (x==2 and y==12)**

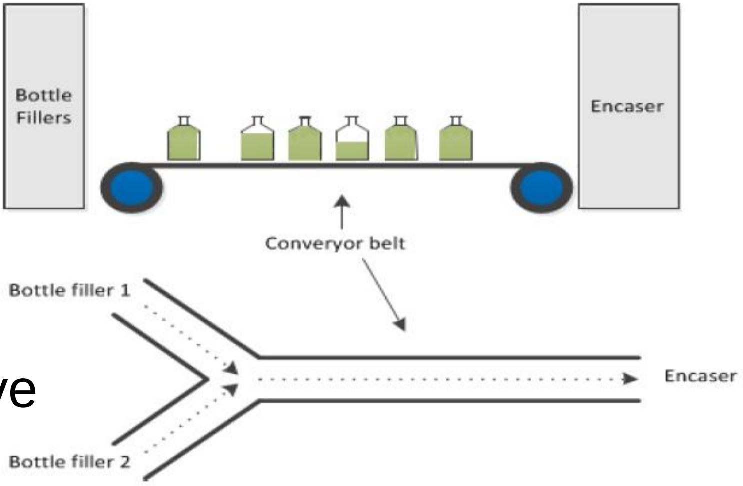
The only one that is counter-intuitive is why isn't P.B --> P.A satisfied? Clearly, if an automaton is in state P.B, then once the clock y is over 10, the x clock will advance to 5 and force the transition to P.A. The reason the "leads to" operator is not satisfied here is because we could have an INFINITE number of transitions in the right loop before taking the transition back to P.A.

Report Writing

- Technical Reports generally include the following sections in this order:
  - Abstract
  - Introduction
  - Literature Review
  - Methodology
  - Results
  - Discussion
  - Conclusion
- See "Writing Engineering Reports" by the Purdue Writing Lab for more details

Sample Projects

- Real-time LEGO Domino Layer – Brian Sweeney and Xiaolong (Daniel) Wang
  - Report
- Discovering key features of real-time Java – Karl Remarais
  - Report
- Development of a Mobile Sensor Platform for Phenotyping – Jed Barker
  - Report
  - Presentation
- Extending semaphores with non-preemptive critical sections (NPCS) in FreeRTOS
  - Carlos Salazar
  - Report



Development of a Mobile Sensor Platform For Phenotyping

Jed Barker

Background

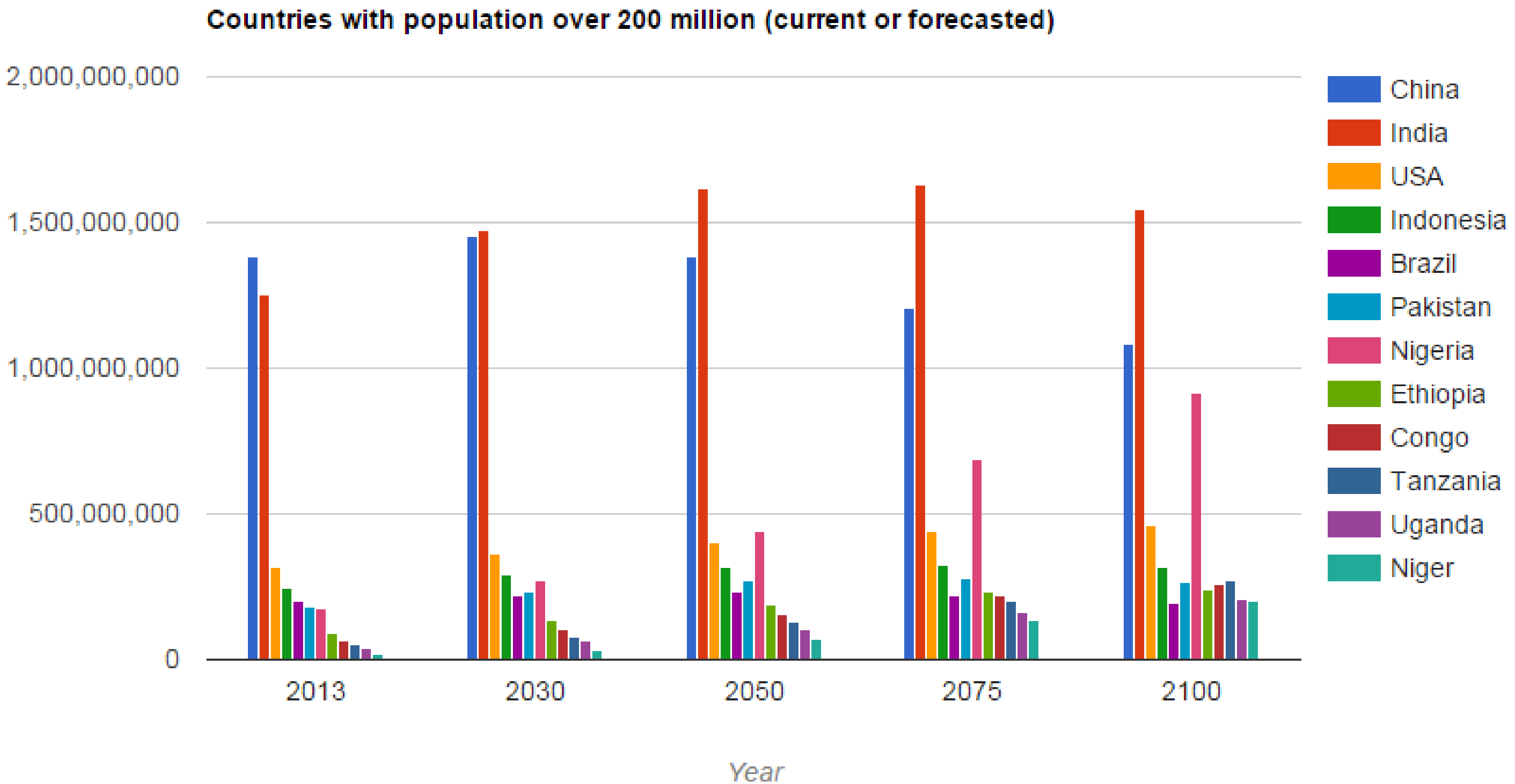
Objectives

Methods

UPPAAL

Conclusions

World Population



From http://www.worldometers.info/world-population/#growthrate

Background

7 Billion → 9 Billion

40 Years

Background

New Plant Cultivars

High Yield

Drought Tolerant

Background

New Plant Cultivars

To quickly and efficiently achieve this:

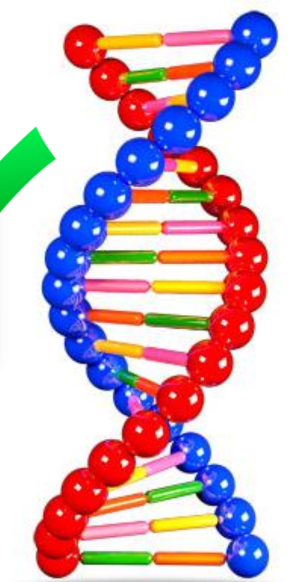
- Understand the genetic basis of complex traits
- Improve methods for predicting complex traits

Background

New Plant Cultivars

- Understand the genetic basis of complex traits

Genetic Information



+

Phenotypic Information

Yield
Canopy Temp
Height
Spectral Reflectance

Background

Research Plots

Background

Objectives

Methods

UPPAAL

Conclusions

Objectives

- Assemble a sensor system/platform
- Develop software for collecting and georeferencing sensor data
- Verification of system timeliness in collecting and georeferencing sensor data
- Determine the effects of light intensity and ambient temperature on sensor readings

Objectives

- Develop an UPPAAL model of the phenotyping system

Background

Objectives

Methods

UPPAAL

Conclusions

Methods

Platform Requirements

- High Clearance Vehicle

Methods

Platform Requirements

- High Clearance Vehicle
- Three Sensor Rows

Methods

Platform Requirements

- High Clearance Vehicle
- Three Sensor Rows
- Two GPS units

Methods

Platform Requirements

- High Clearance Vehicle
- Three Sensor Rows
- Two GPS units
- GreenSeeker, Crop Circle, IRT, Ultrasonic Sensor

GreenSeeker

Crop Circle

IRT

Ultrasonic Sensor

Methods

Platform Requirements

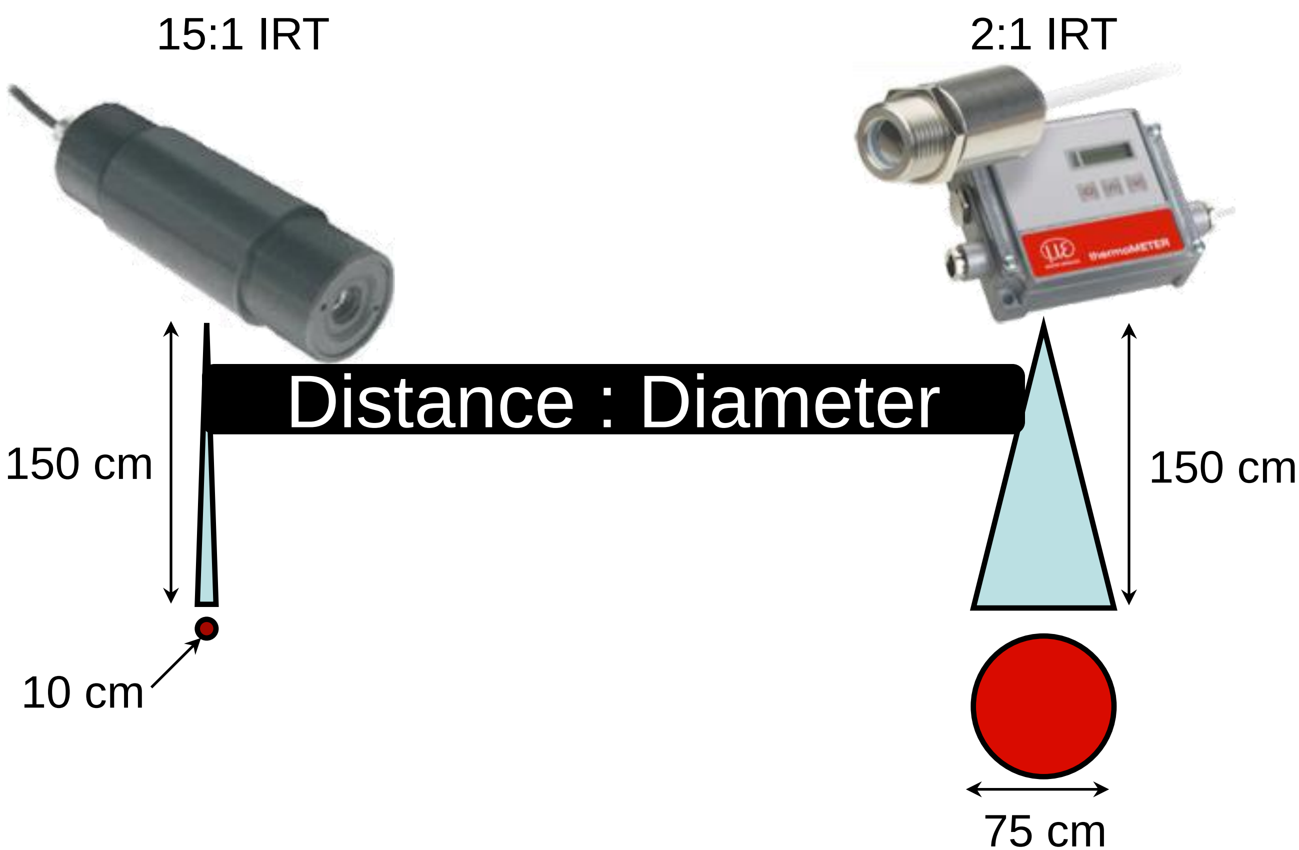
GreenSeeker

656 nm & 774 nm

Crop Circle

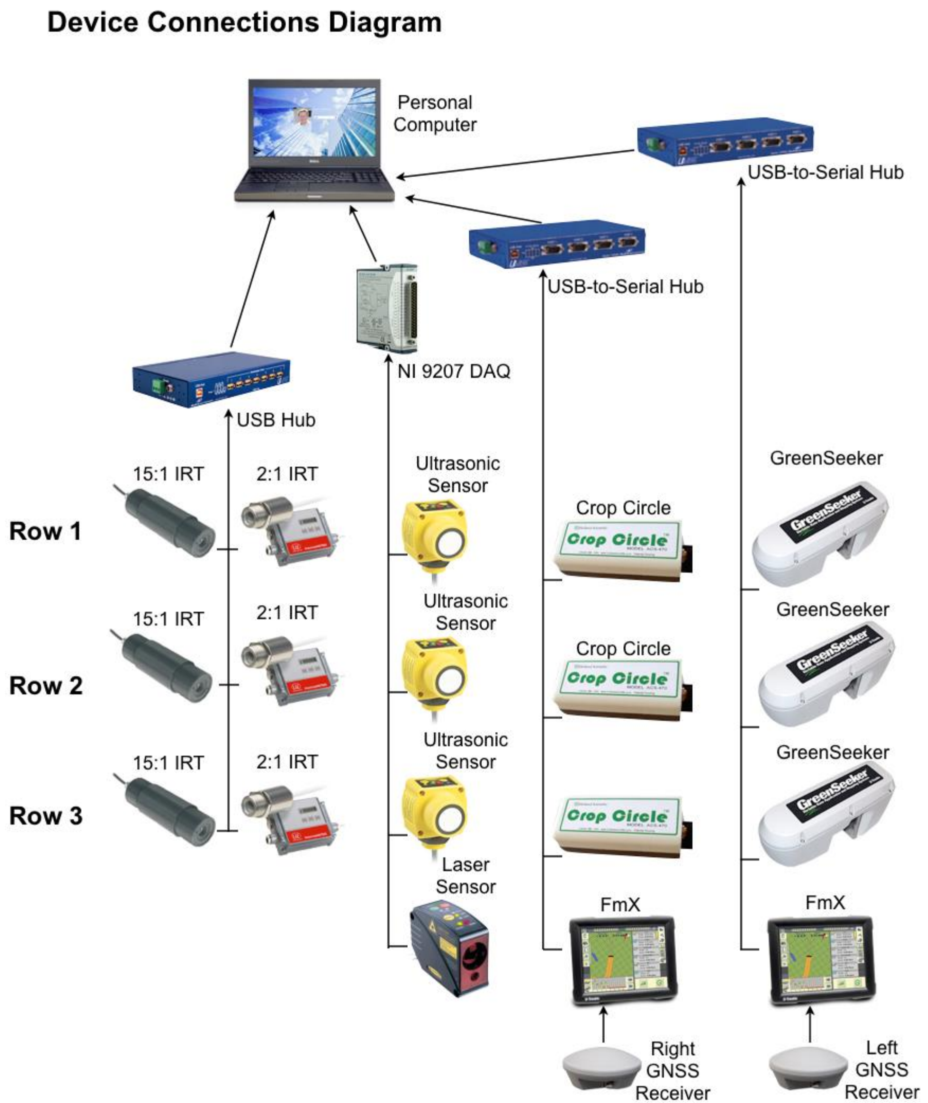
Channel 1 - 670 nm
Channel 2 - 760 nm
Channel 3 - 550 nm

Platform Requirements



Methods

Hardware Connections

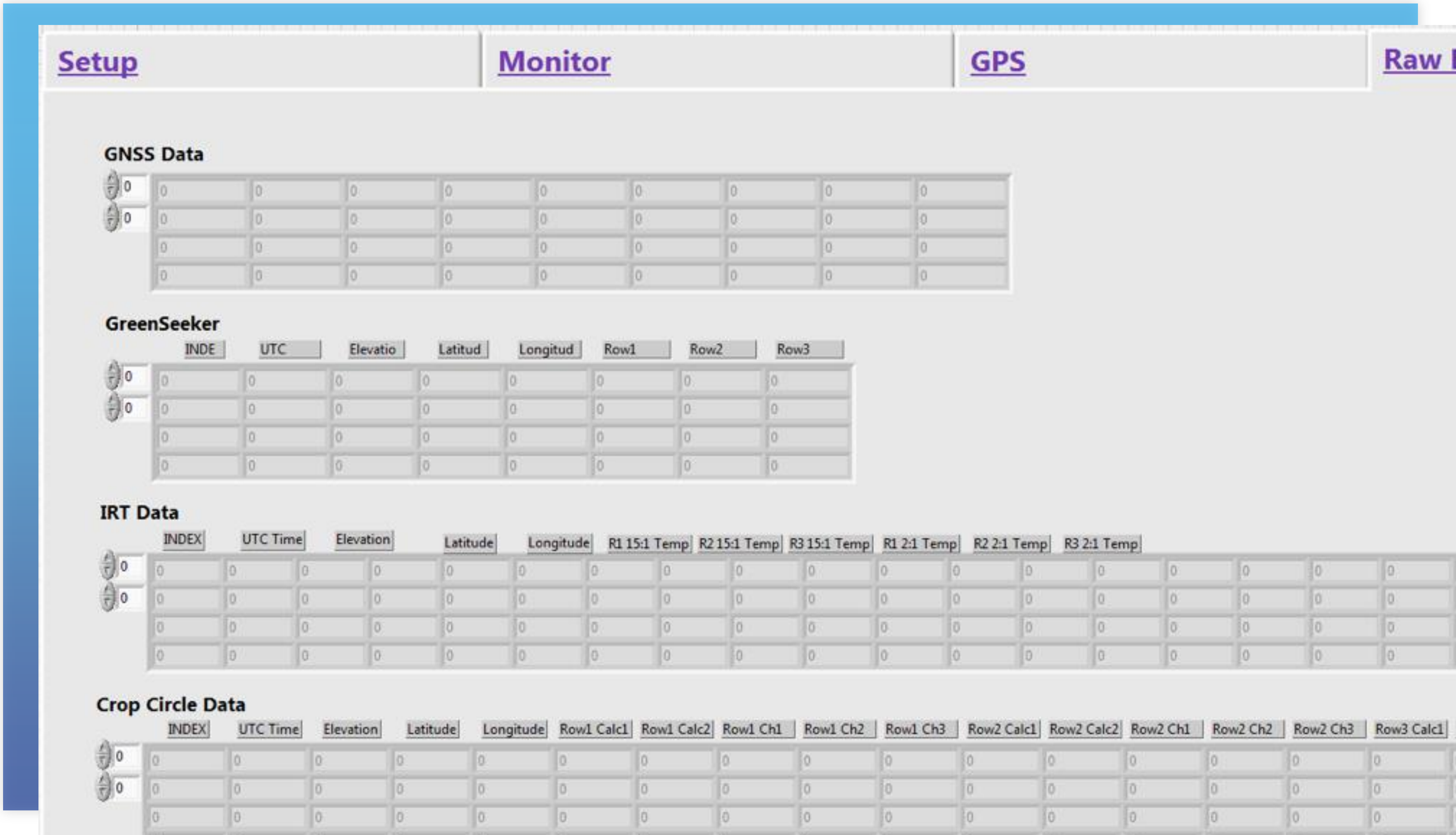


Methods

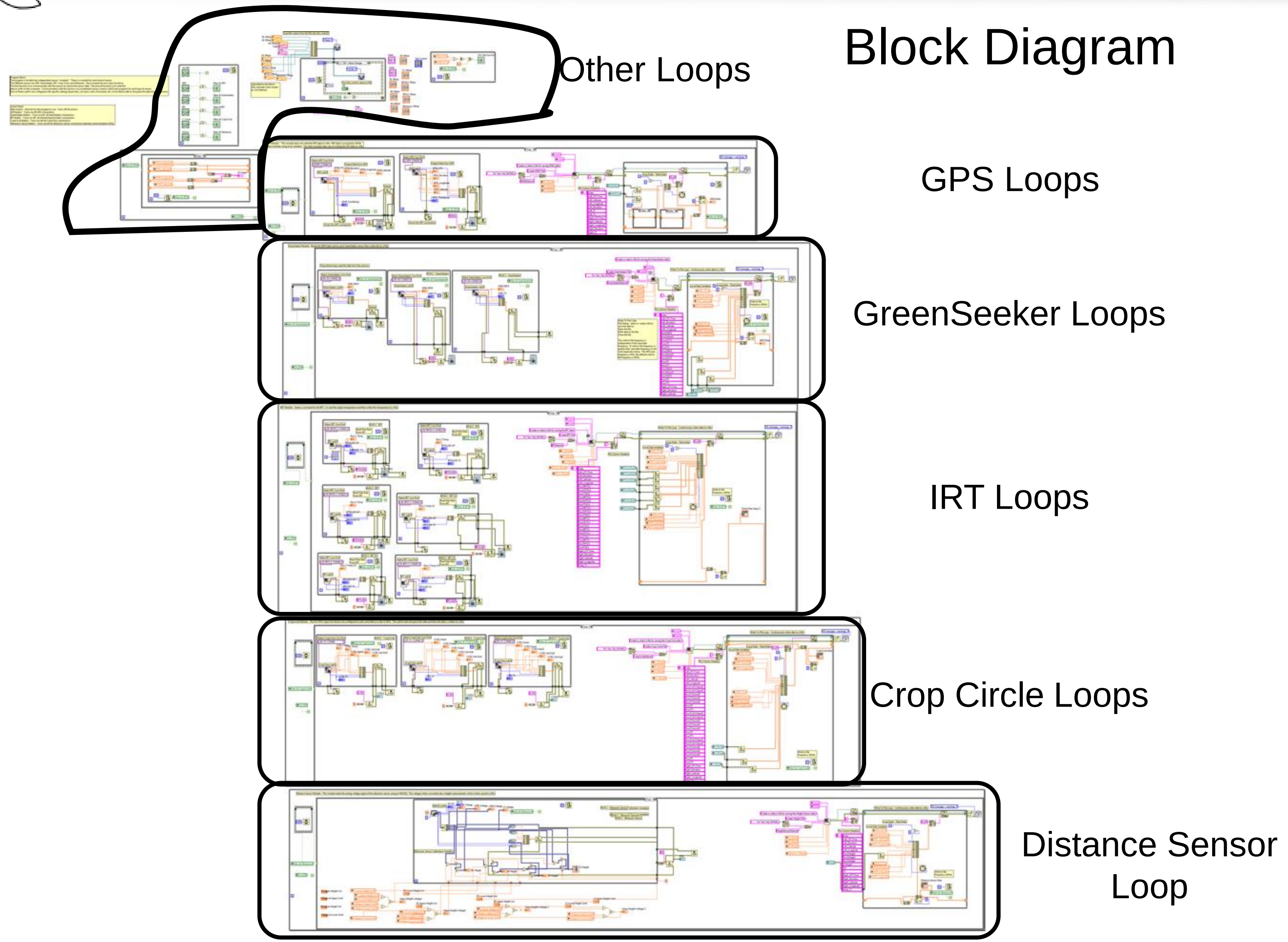
Phenotyping Software

NATIONAL INSTRUMENTS
LabVIEW

Phenotyping Software



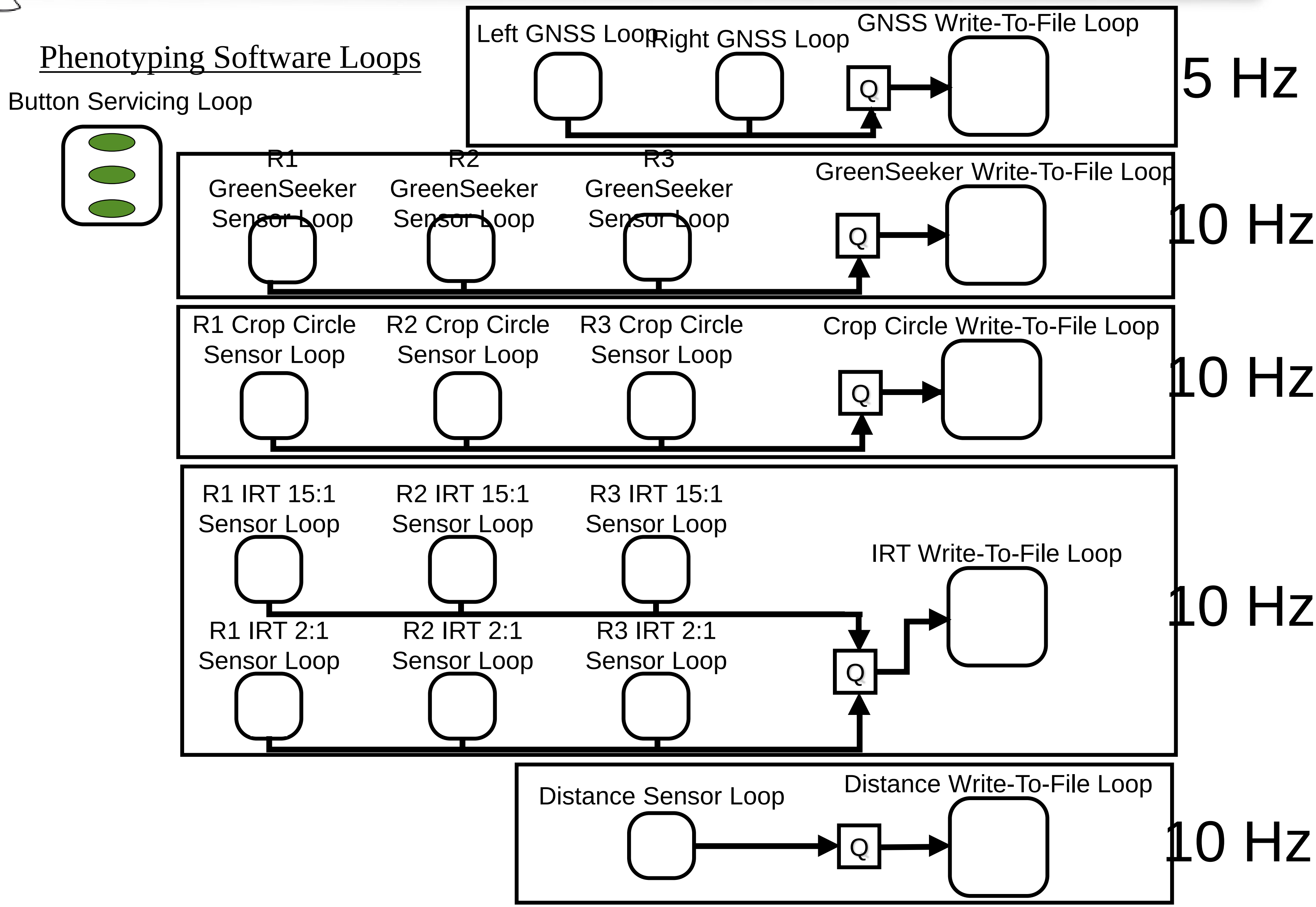
Methods



Block Diagram
Other Loops
GPS Loops
GreenSeeker Loops
IRT Loops
Crop Circle Loops
Distance Sensor Loop

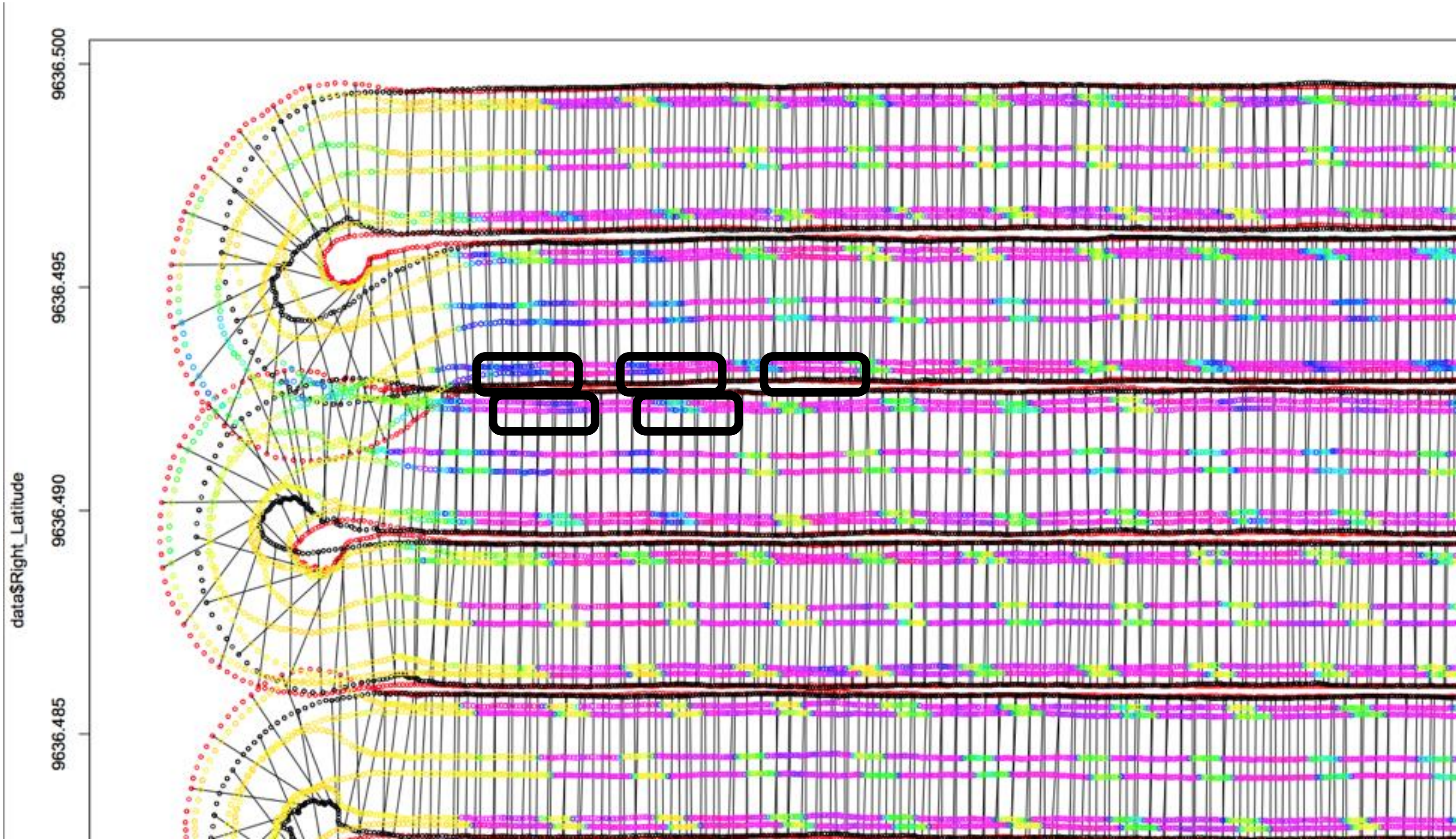
Methods

Phenotyping Software Loops



Methods

Data Files



Background

Objectives

Methods

UPPAAL

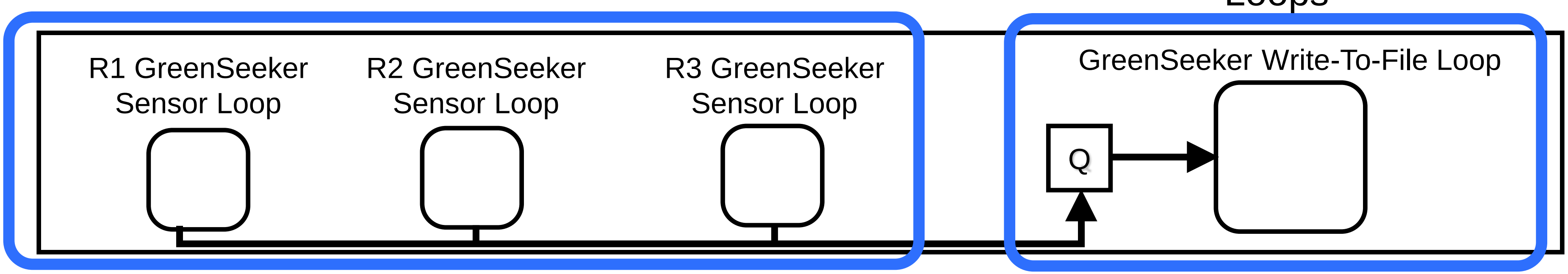
Conclusions

UPPAAL

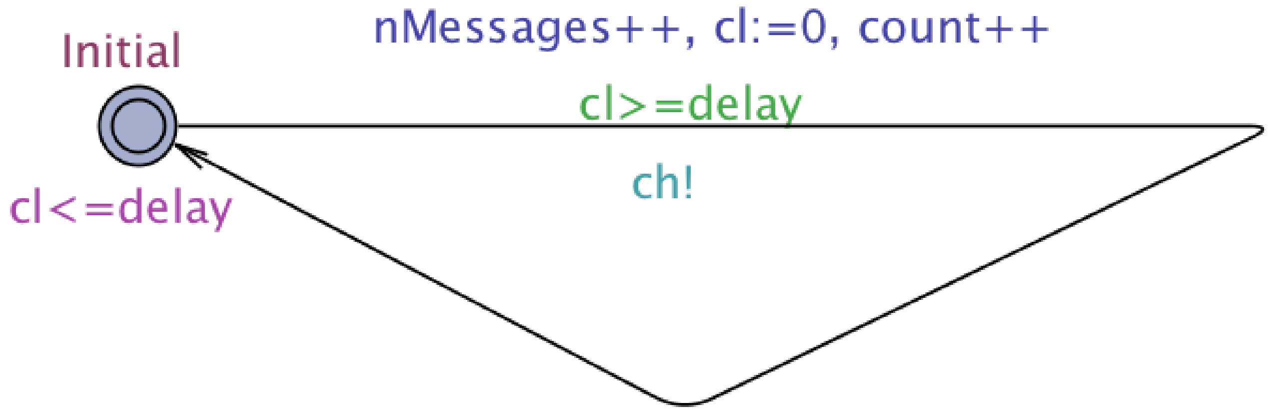
Sensors

Sensor Read Loops

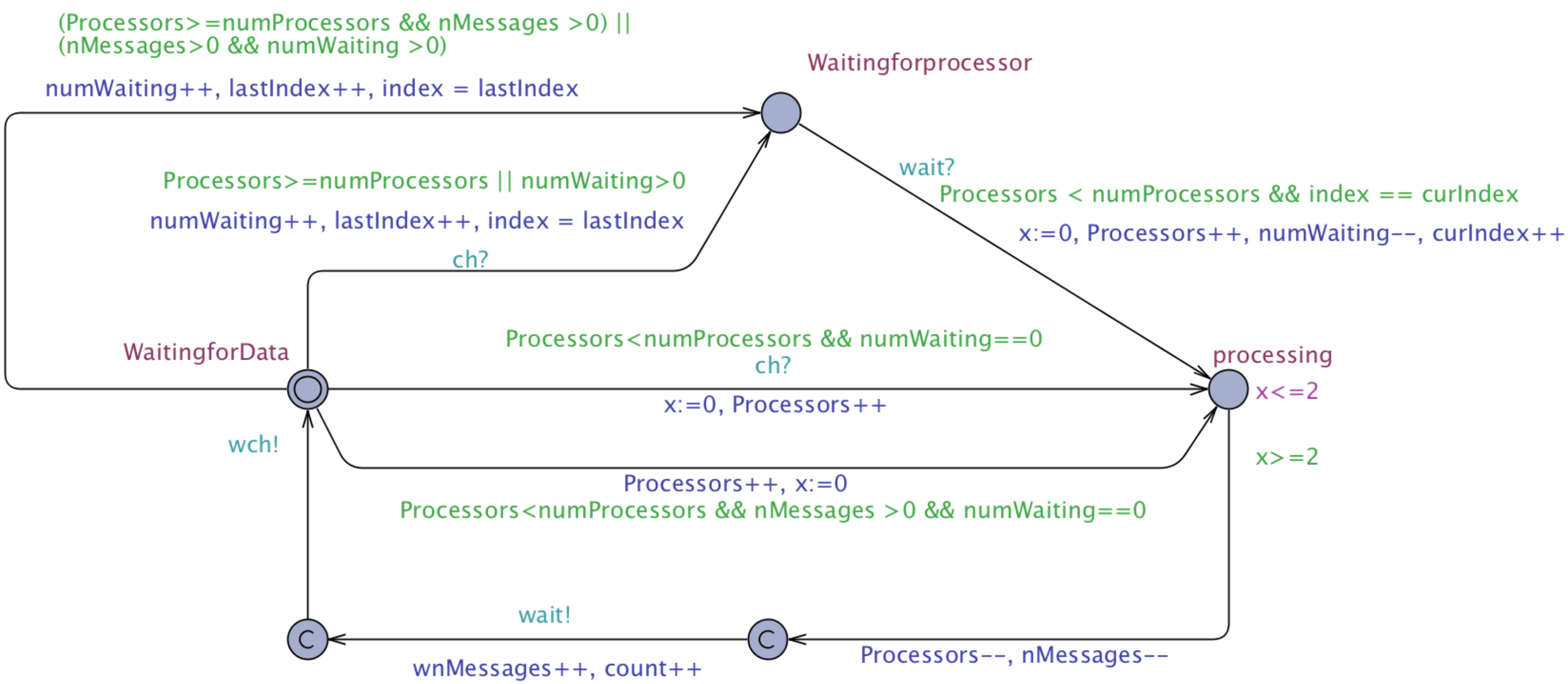
Write To File Loops



Sensor

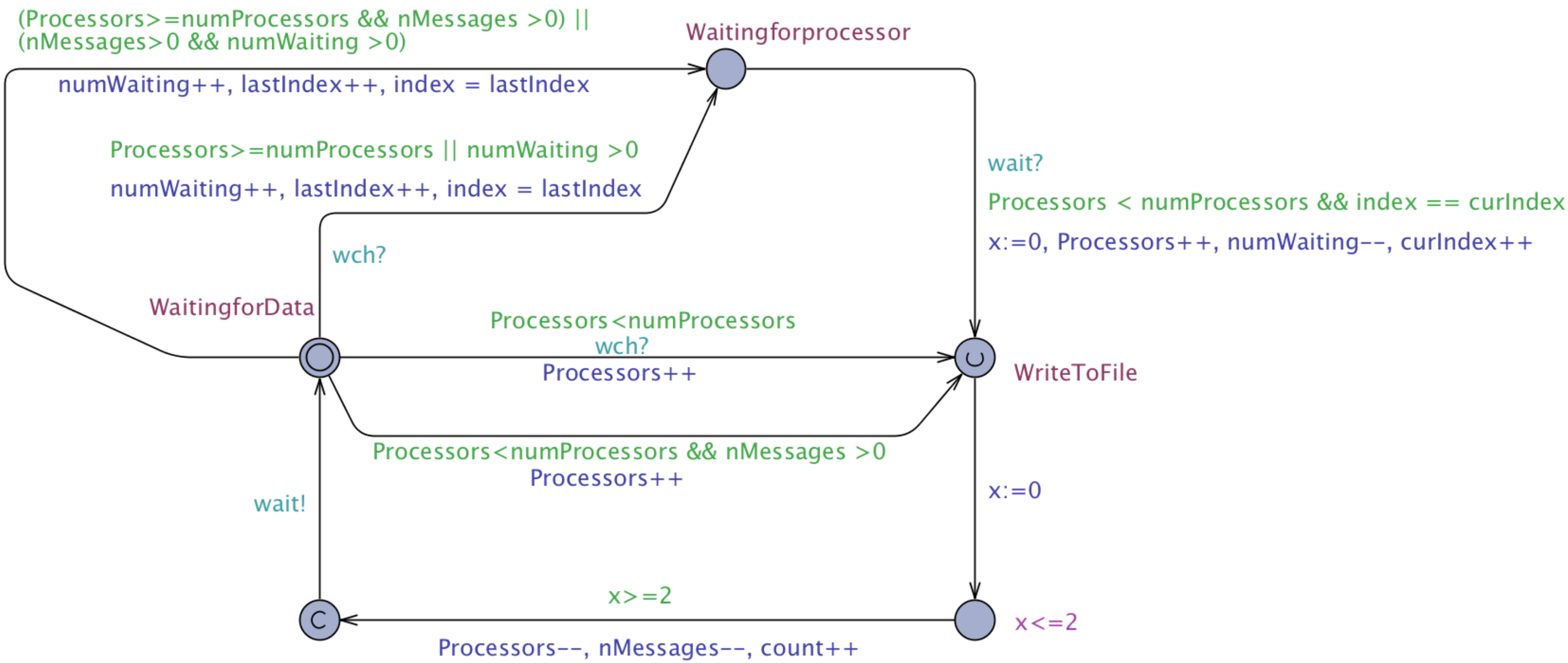


Sensor Read


(Processors>=numProcessors && nMessages >0) ||
(nMessages>0 && numWaiting >0)
numWaiting++, lastIndex++, index = lastIndex
Waitingforprocessor
wait?
Processors < numProcessors && index == curIndex
x:=0, Processors++, numWaiting--, curIndex++
Processors>=numProcessors || numWaiting>0
numWaiting++, lastIndex++, index = lastIndex
ch?
WaitingforData
Processors<numProcessors && numWaiting==0
ch?
x:=0, Processors++
processing
x<=2
wch!
x>=2
Processors++, x:=0
Processors<numProcessors && nMessages >0 && numWaiting==0
wait!
wnMessages++, count++
Processors--, nMessages--

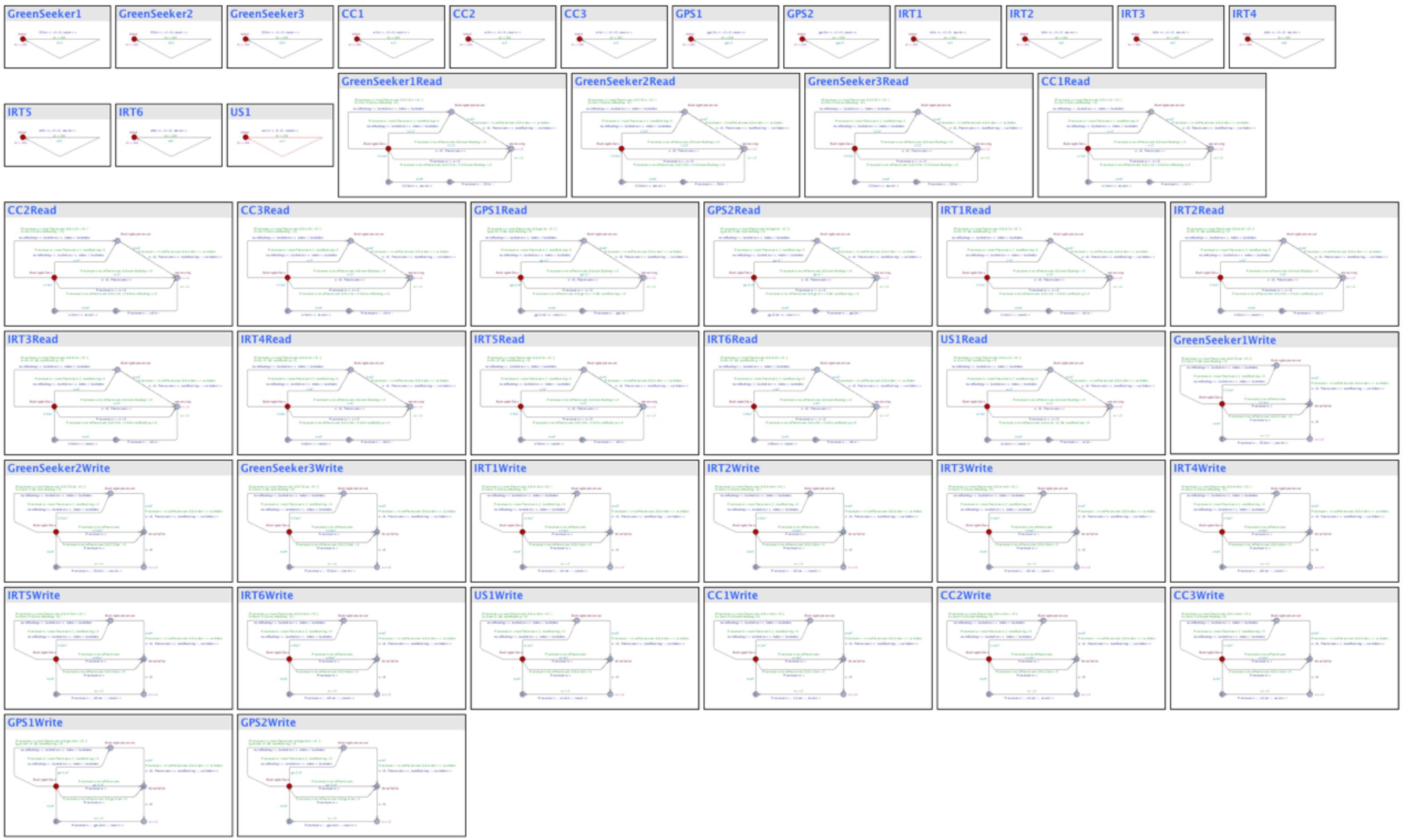
UPPAAL

Write To File



UPPAAL

Complete Model



Model Constraints

- Number of Processors
- Sensor Send Frequency
- Processing Time

UPPAAL

Model Verification

- Data Buffering/Accumulation
- Timeliness of Data Processing
- Effects of Additional Components

Background

Objectives

Methods

Results

Conclusions

Sample Projects

- Real-time LEGO Domino Layer – Brian Sweeney and Xiaolong (Daniel) Wang
  - Report
- Discovering key features of real-time Java – Karl Remarais
  - Report
- Development of a Mobile Sensor Platform for Phenotyping – Jed Barker
  - Report
  - Presentation
- Extending semaphores with non-preemptive critical sections (NPCS) in FreeRTOS - Carlos Salazar
  - Report