Heart modeling: Timed automata

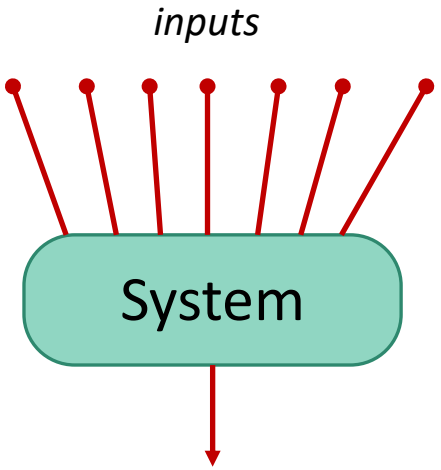
Lecture 12

Principles of Modeling for Cyber-Physical Systems

Instructor: Madhur Behl

Why model checking ?

Does



satisfy

Requirements

?

Does

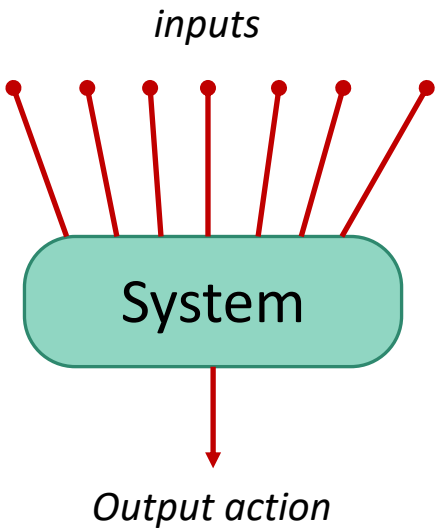
satisfy

Maintain at least LRI ventricle rate

Do not pace beyond maximum rate (URI)

…

?

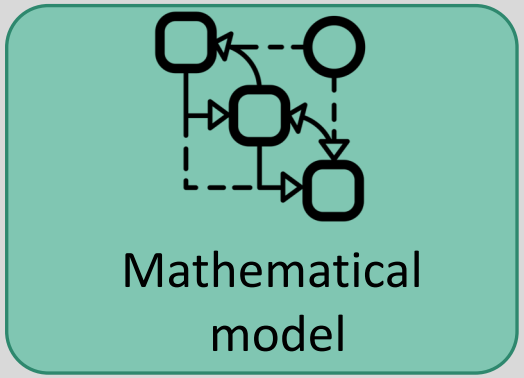


Does **satisfy** **?**

Requirements

Model Checking UPPAAL Model Checker

Does



Timed automata

satisfy

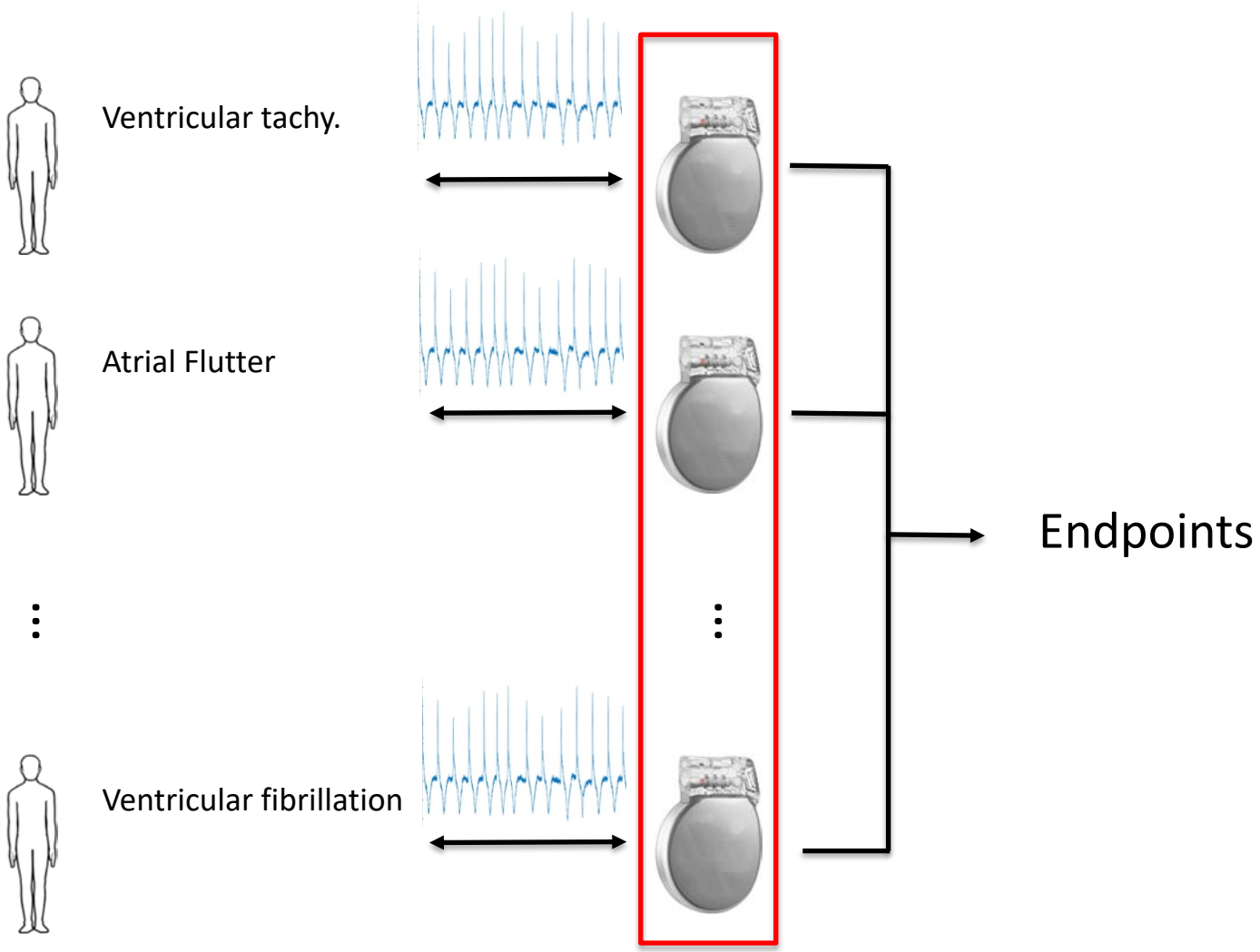
$$F\ p\ \wedge q$$

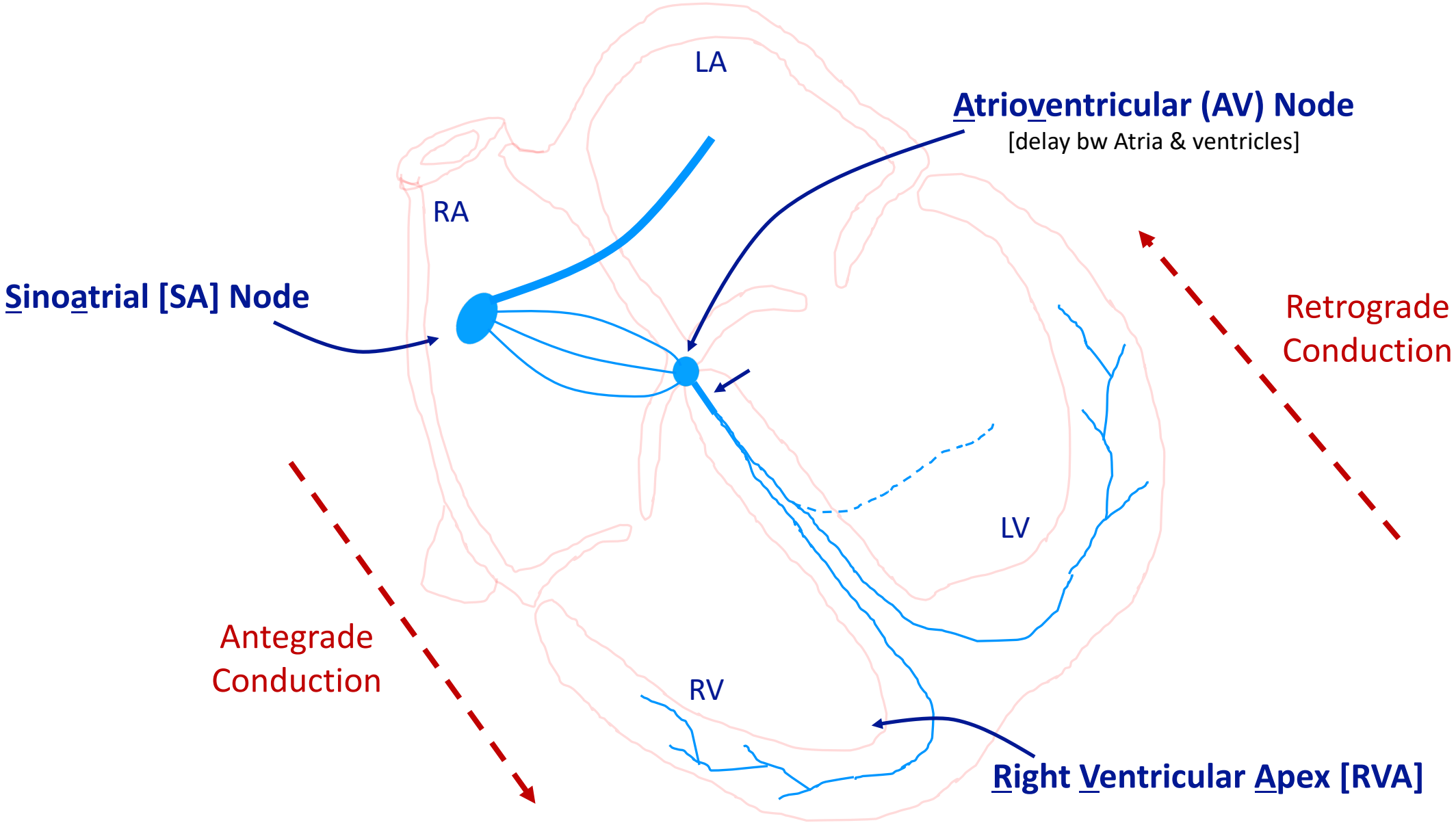
Formal notation

Linear Temporal Logic

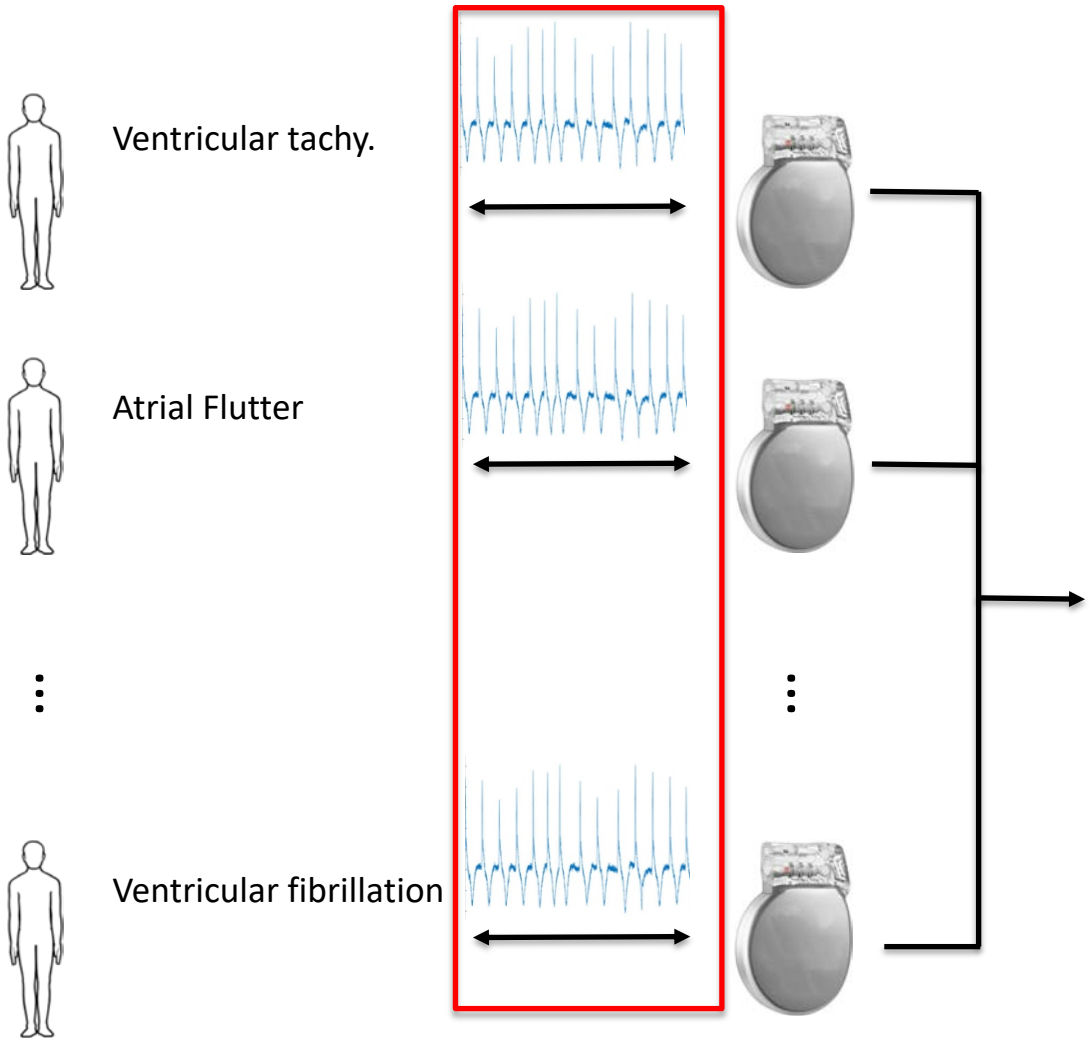
?

So far: Looked at a device algorithm



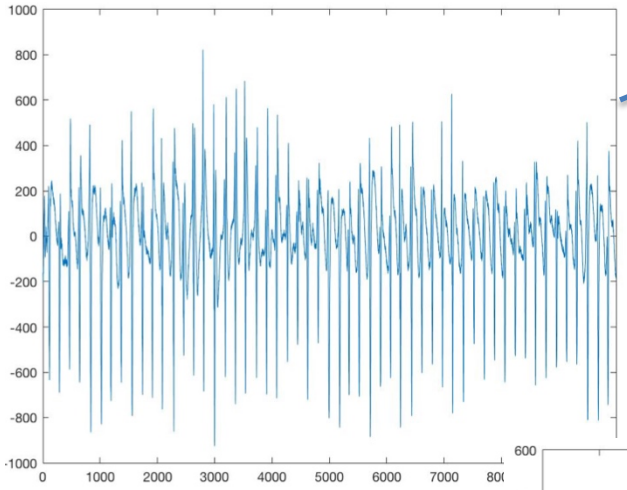

LA
Atrioventricular (AV) Node
[delay bw Atria & ventricles]
RA
Sinoatrial [SA] Node
Retrograde
Conduction
LV
Antegrade
Conduction
RV
Right Ventricular Apex [RVA]

Need: a model of device inputs



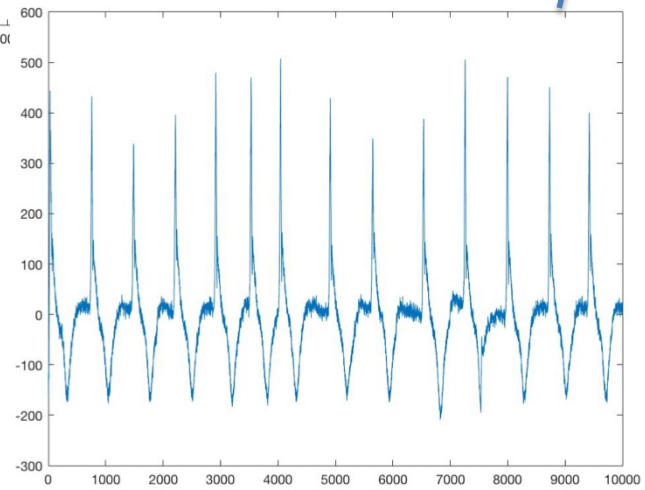
Need: a model of device inputs

Inputs to pacemaker are intra-cardiac electrograms



Atrial signal

Ventricular signal



Lets model the heart as a timed automaton

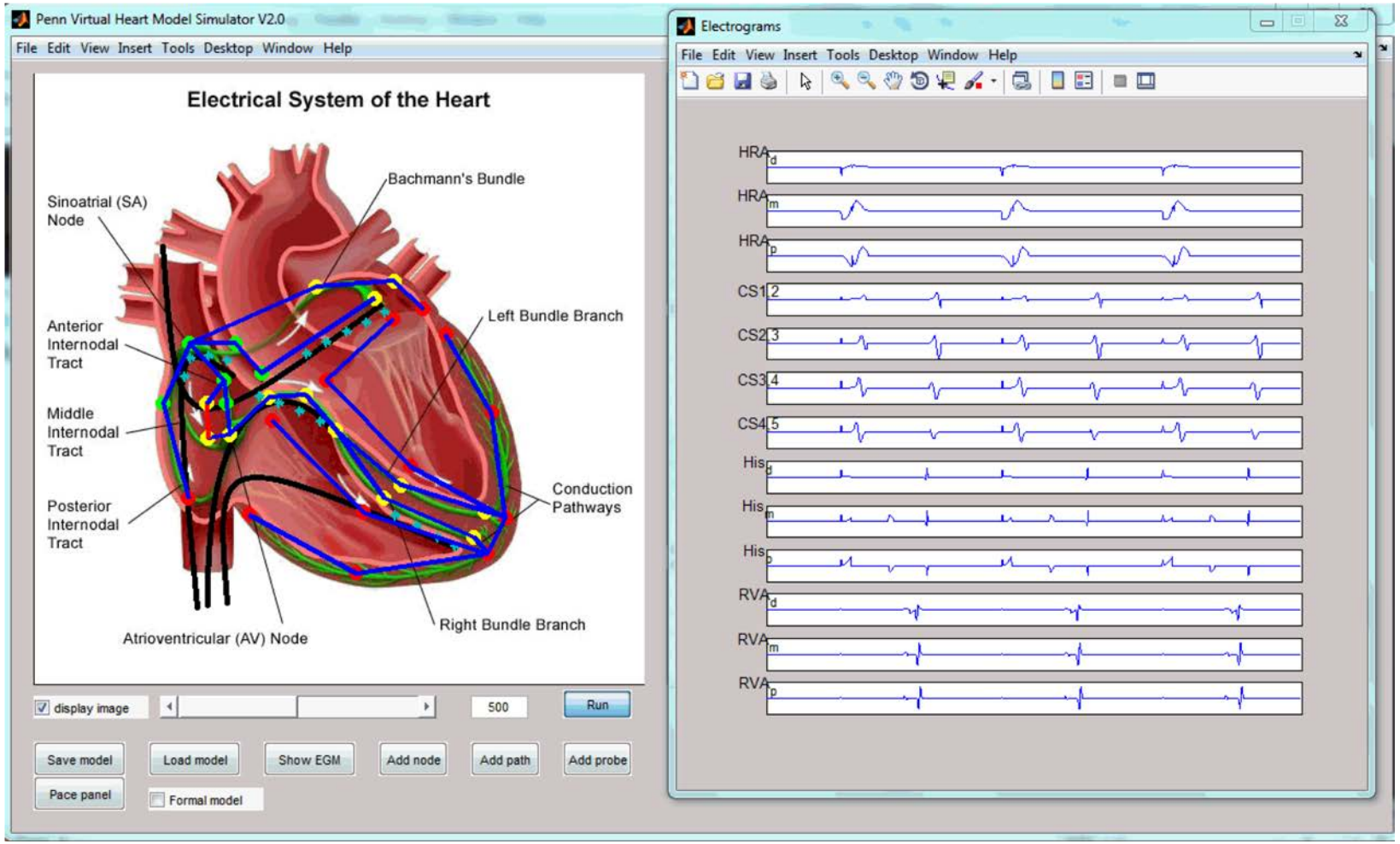
Its about time!

Modeling choices

- Molecular
- Cellular
- Tissue
- Whole organ

- Mechanical
- Electrical

What do we want from the heart model?



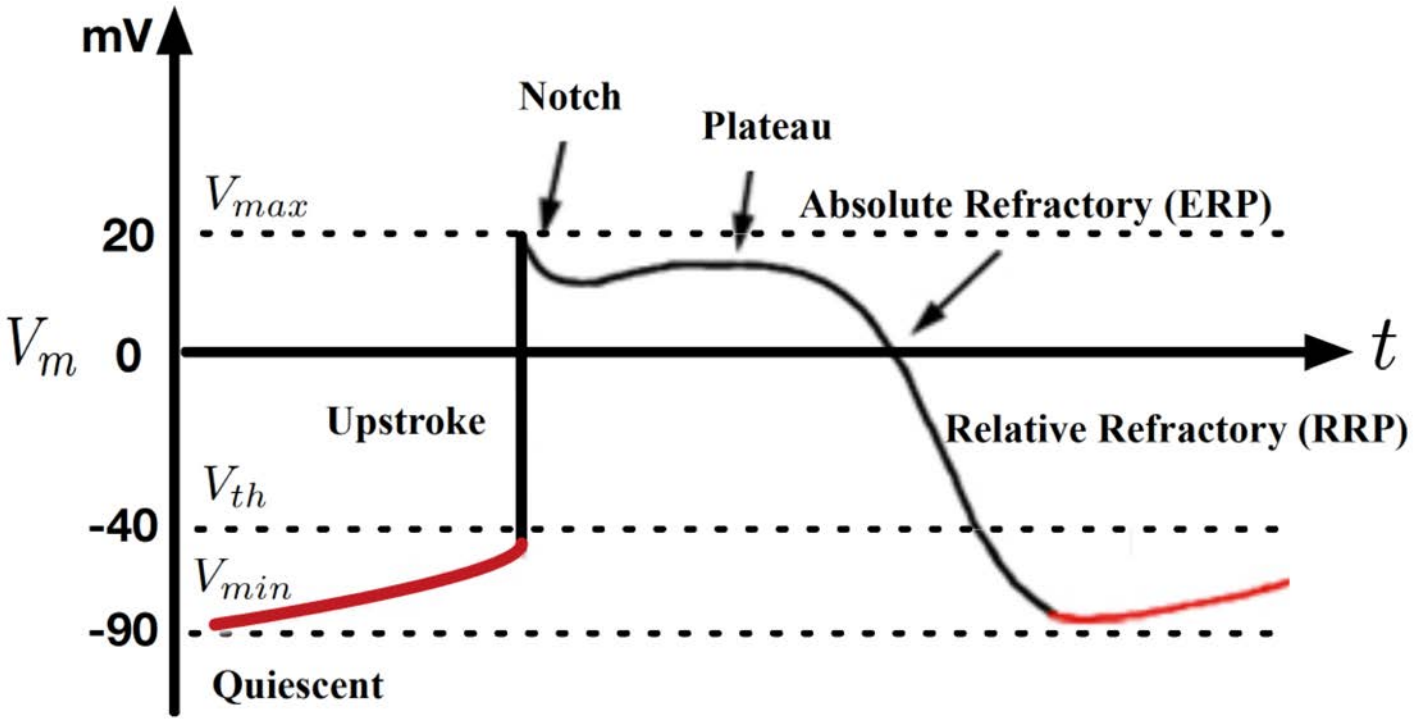
- Predict when it misses beats → model outputs are events in A and V.
- Receive pacing from device → inputs are *pacing* events.

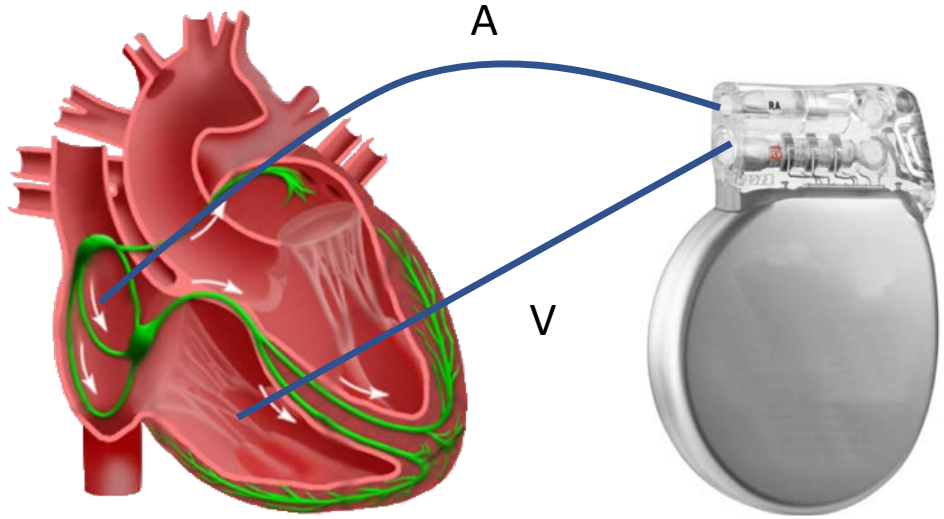
Modeling choices

- Molecular
- Cellular
- Tissue
- Whole organ

- Mechanical
- Electrical

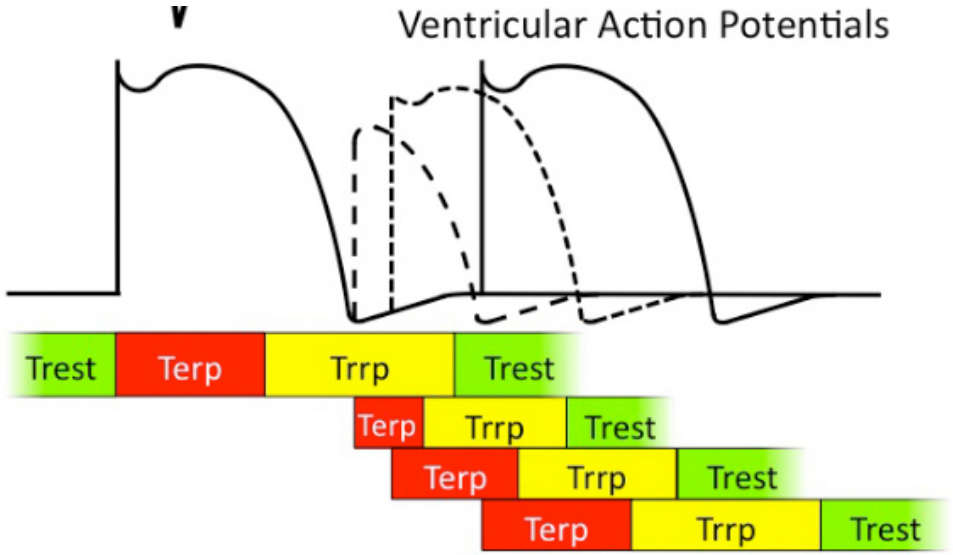
Action potential





Refractory periods

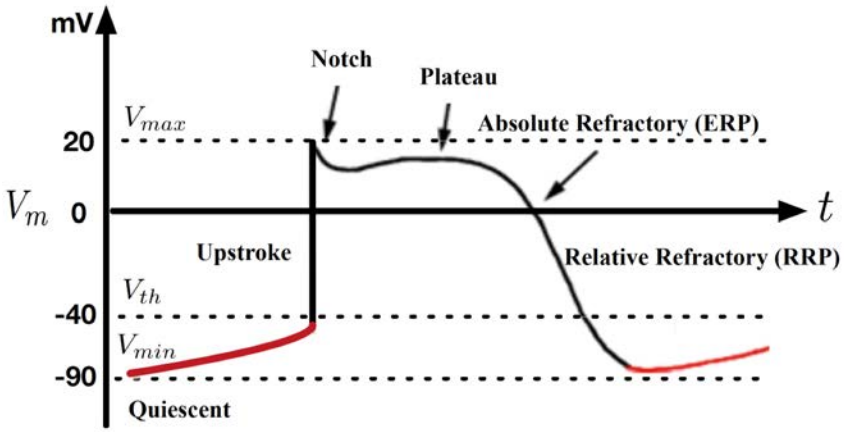
- Divide refractory period into time periods – Rest, ERP, and RRP
- Model refractory properties as timers.
- These time periods can be measured during EP study

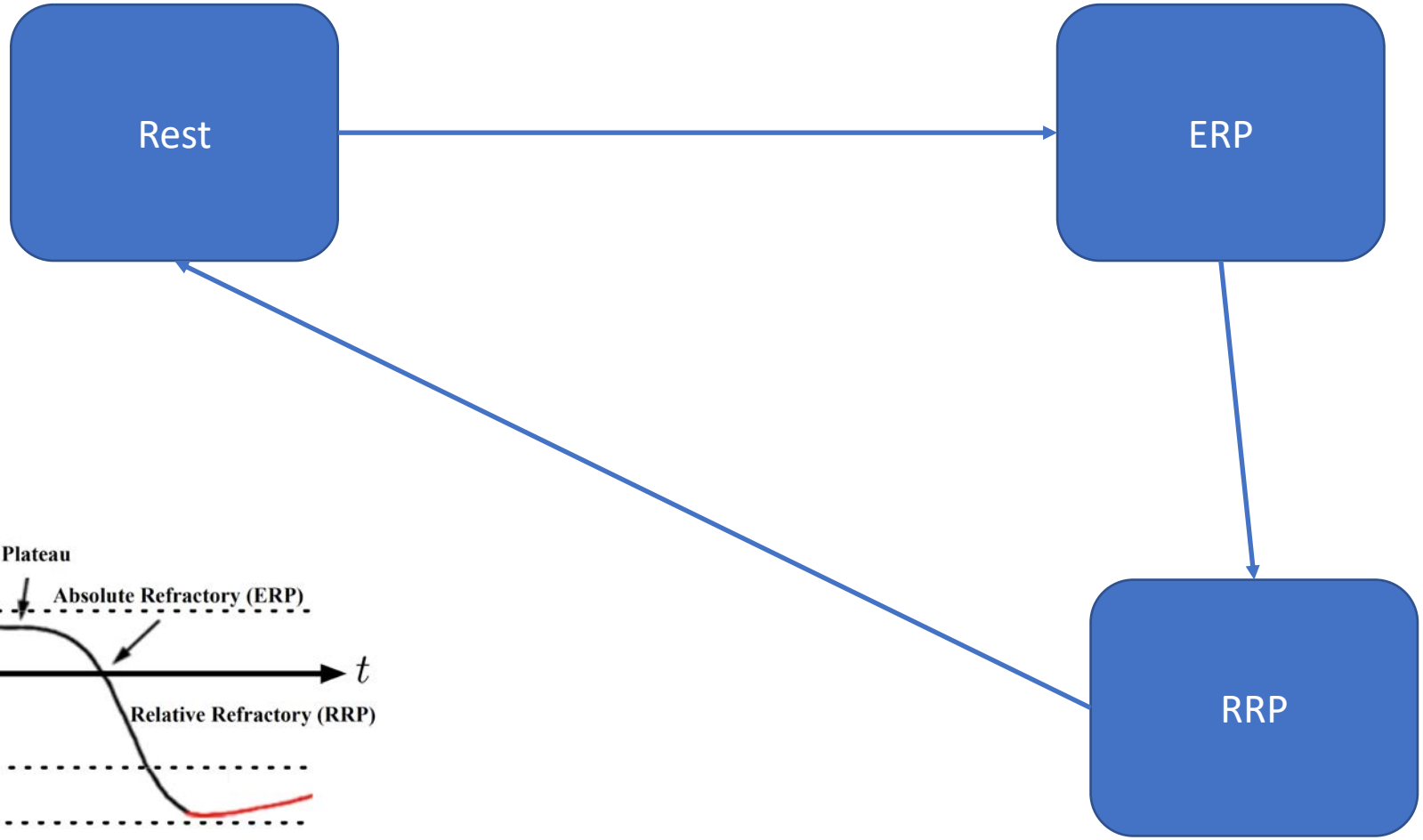


Excitable Cardiac cells

- A piece of heart tissue (myocardium) is made of many *excitable* cells
- Each cell has (roughly speaking) three phases: Rest, Effective Refractory Period (ERP), and the Relative Refractory Period (RRP).
- In Rest, the cell can be excited by electricity. It can also self-excite if enough time elapses
- After being excited (*depolarized)*, it enters a refractory phase during which it cannot be excited a second time. This is the Effective Refractory Period
- After the ERP, it enters a second refractory phase during which it can be excited, but with greater difficulty and with a smaller reaction. This is the Relative Refractory Period
- After the RRP, it returns to Rest and cycle repeats.

One cell


mV
Notch
Plateau
V_{max}
20
Absolute Refractory (ERP)
V_m
0
t
Upstroke
Relative Refractory (RRP)
V_{th}
-40
V_{min}
-90
Quiescent

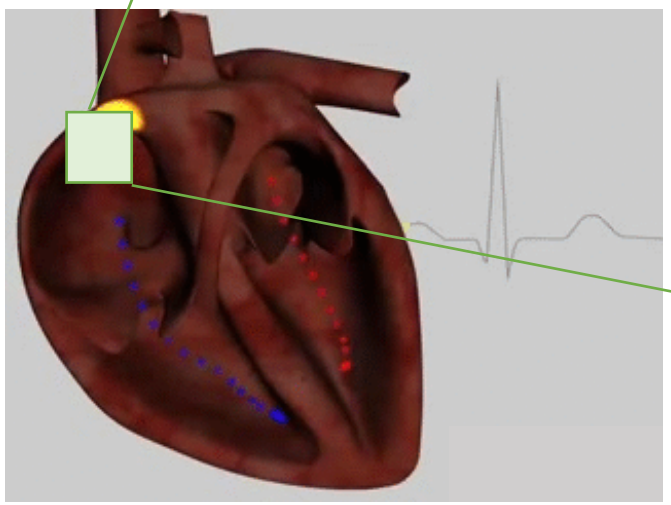

Rest
ERP
RRP

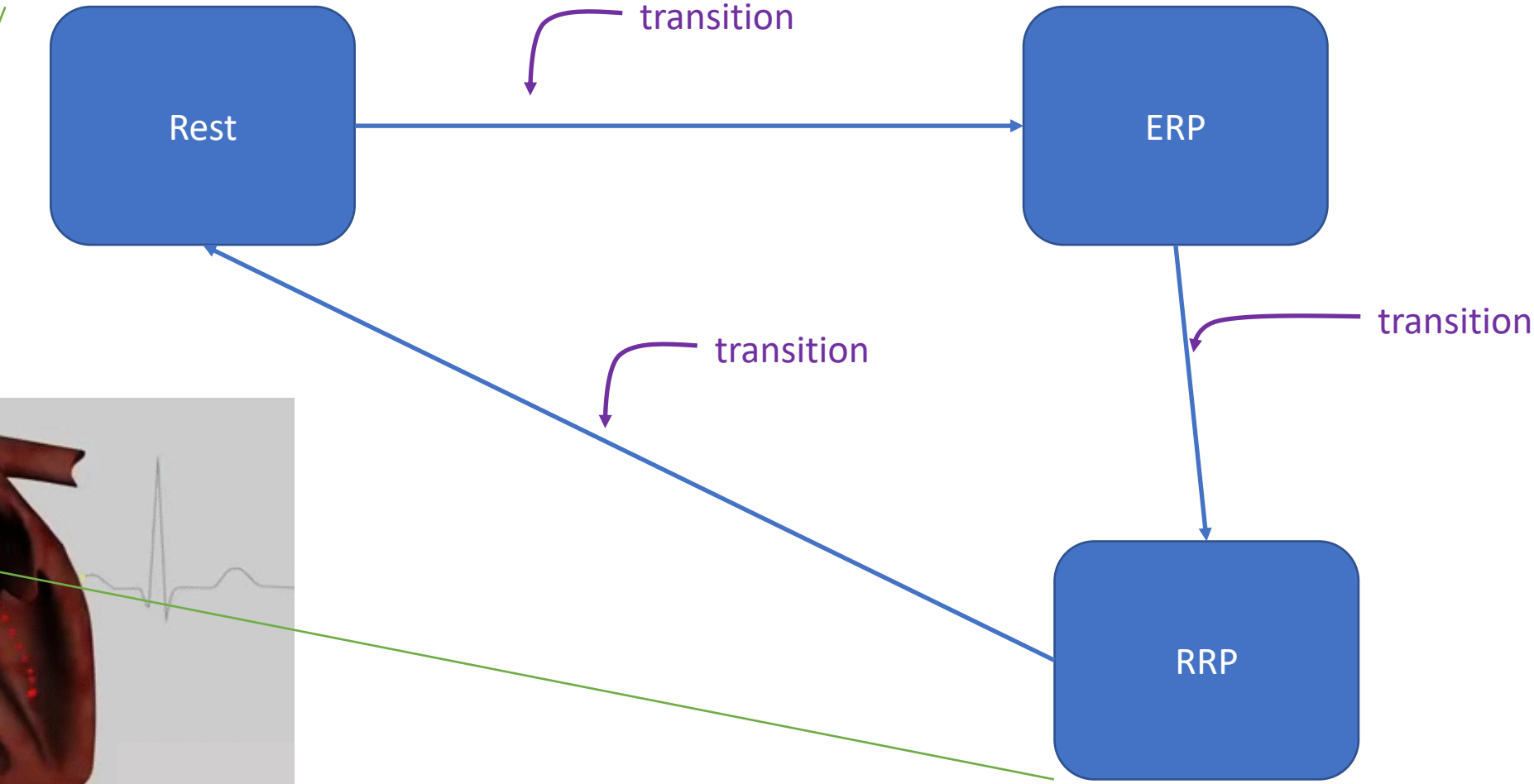
Cardiac tissue : Functional Syncytium

We can model an entire piece of cardiac tissue as having the same three phases as the cells that make it up: Rest, ERP and RRP.

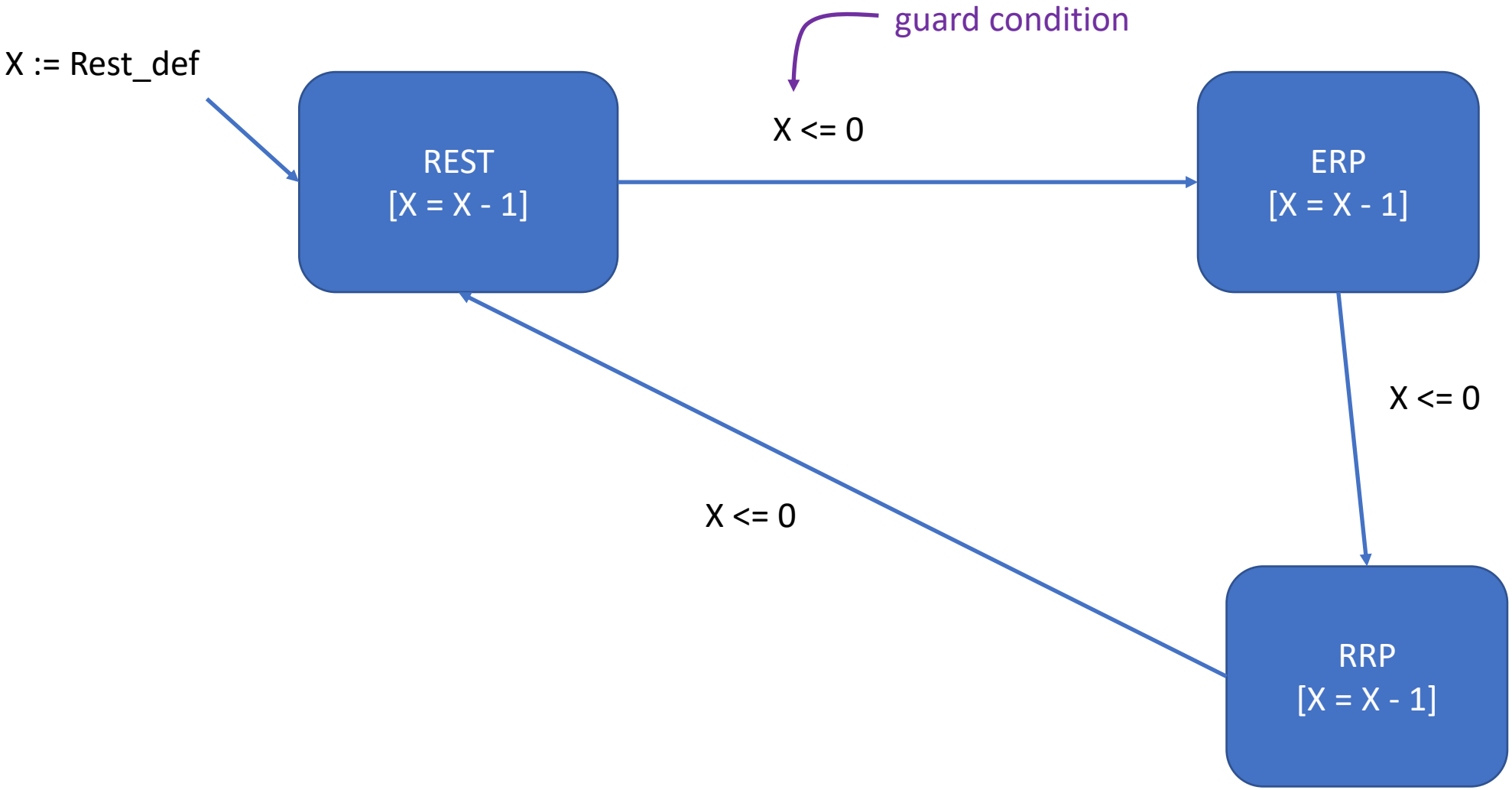
Homegenization:
even though some quantities are time- or space-varying, it might be OK, to first order, to model them as constants across time (and space)

Cardiac tissue: one *node automaton*

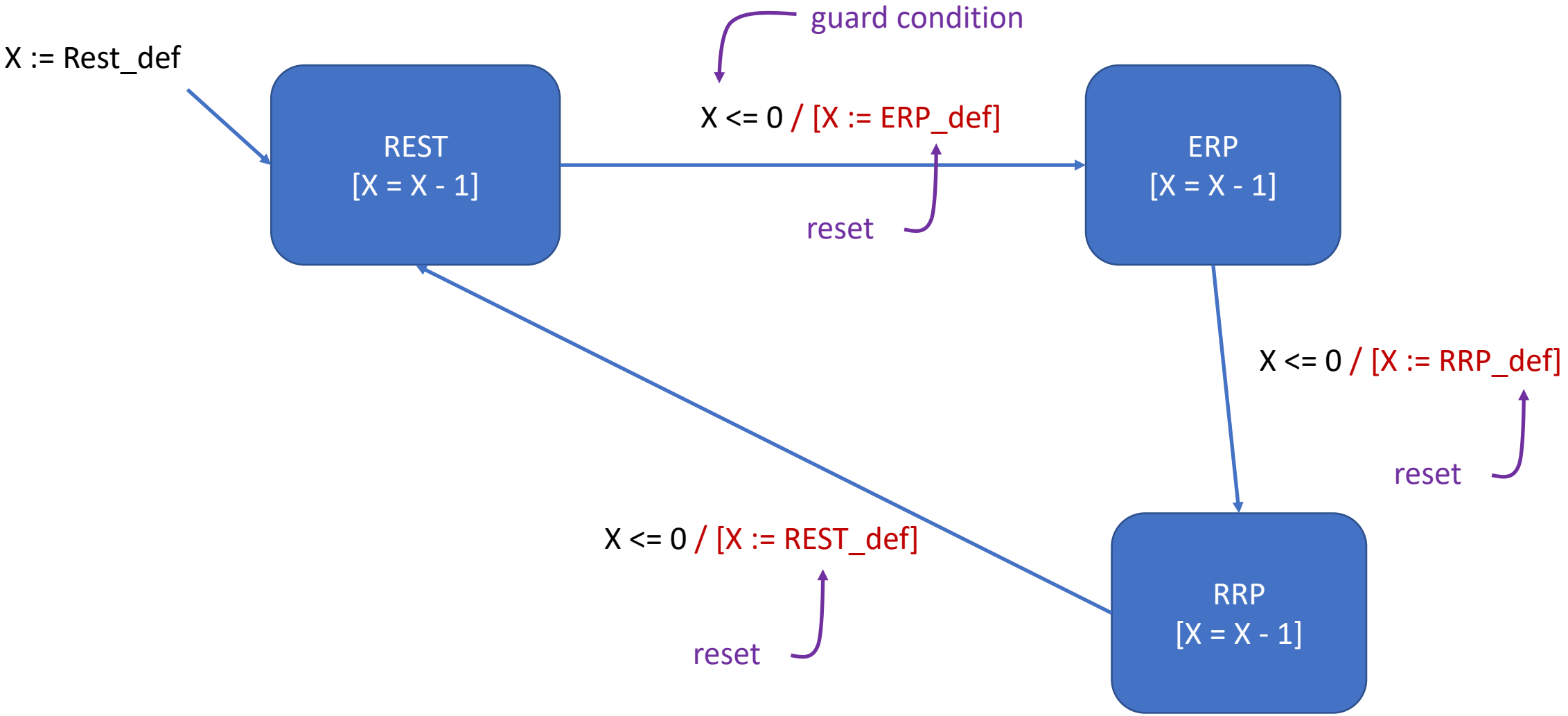




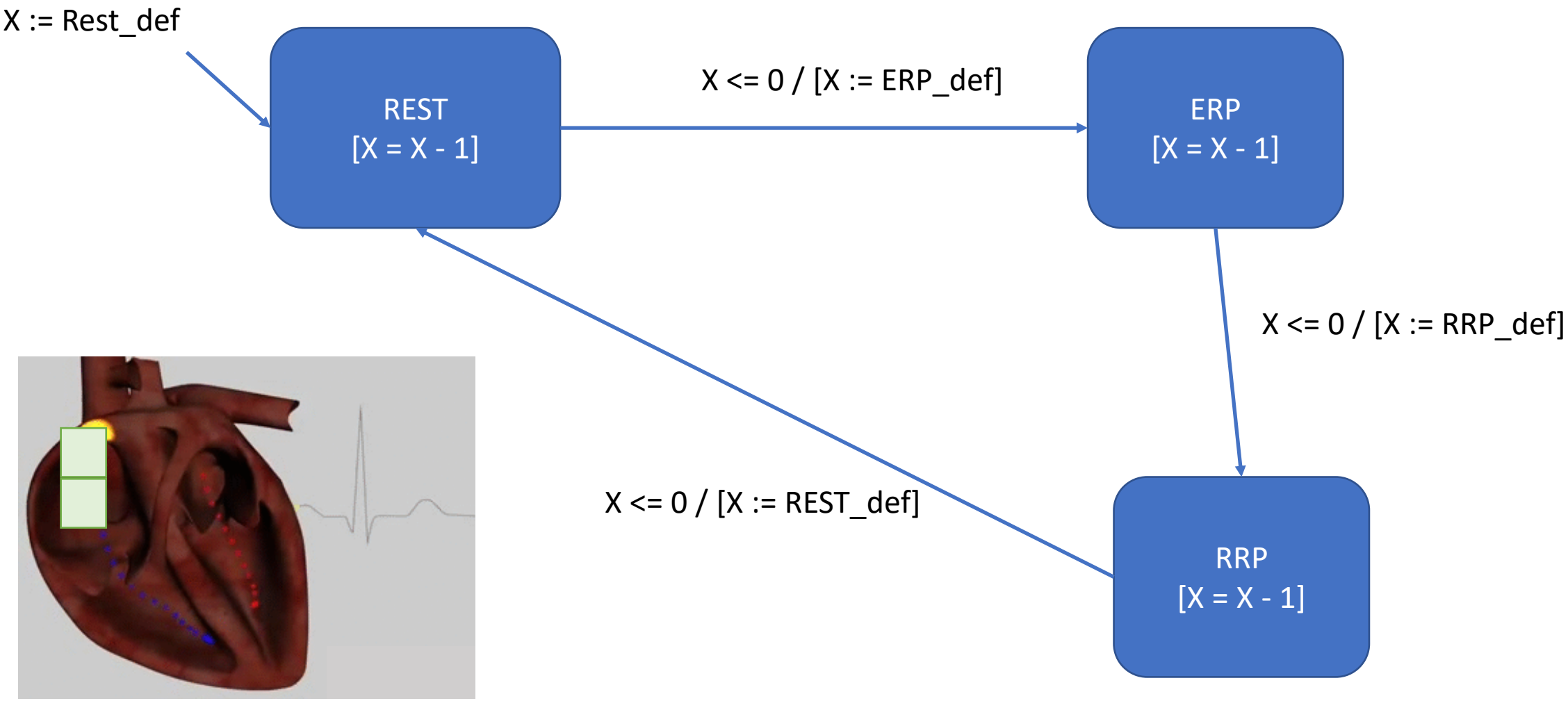
Node automaton with clocks



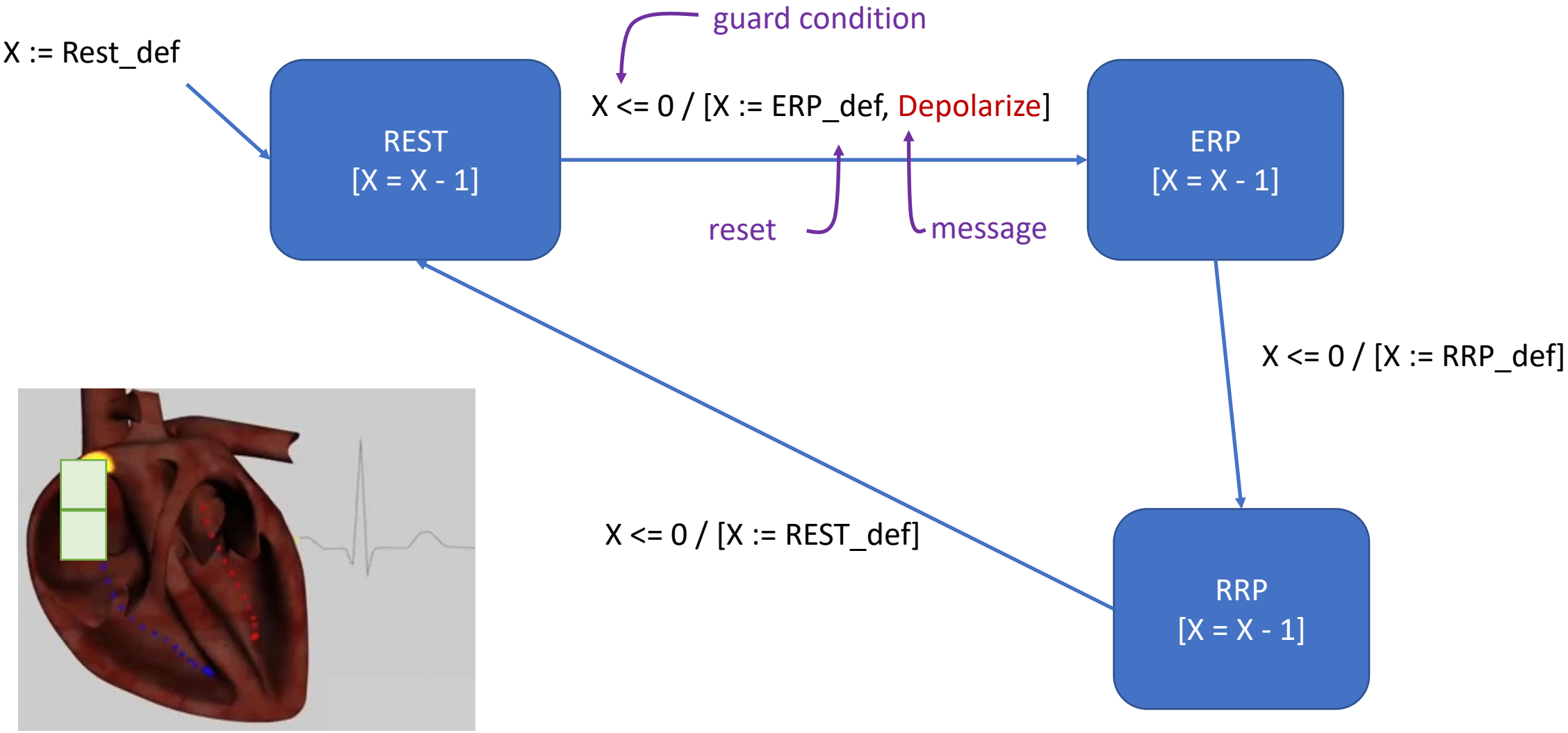
Node automaton with clocks and resets



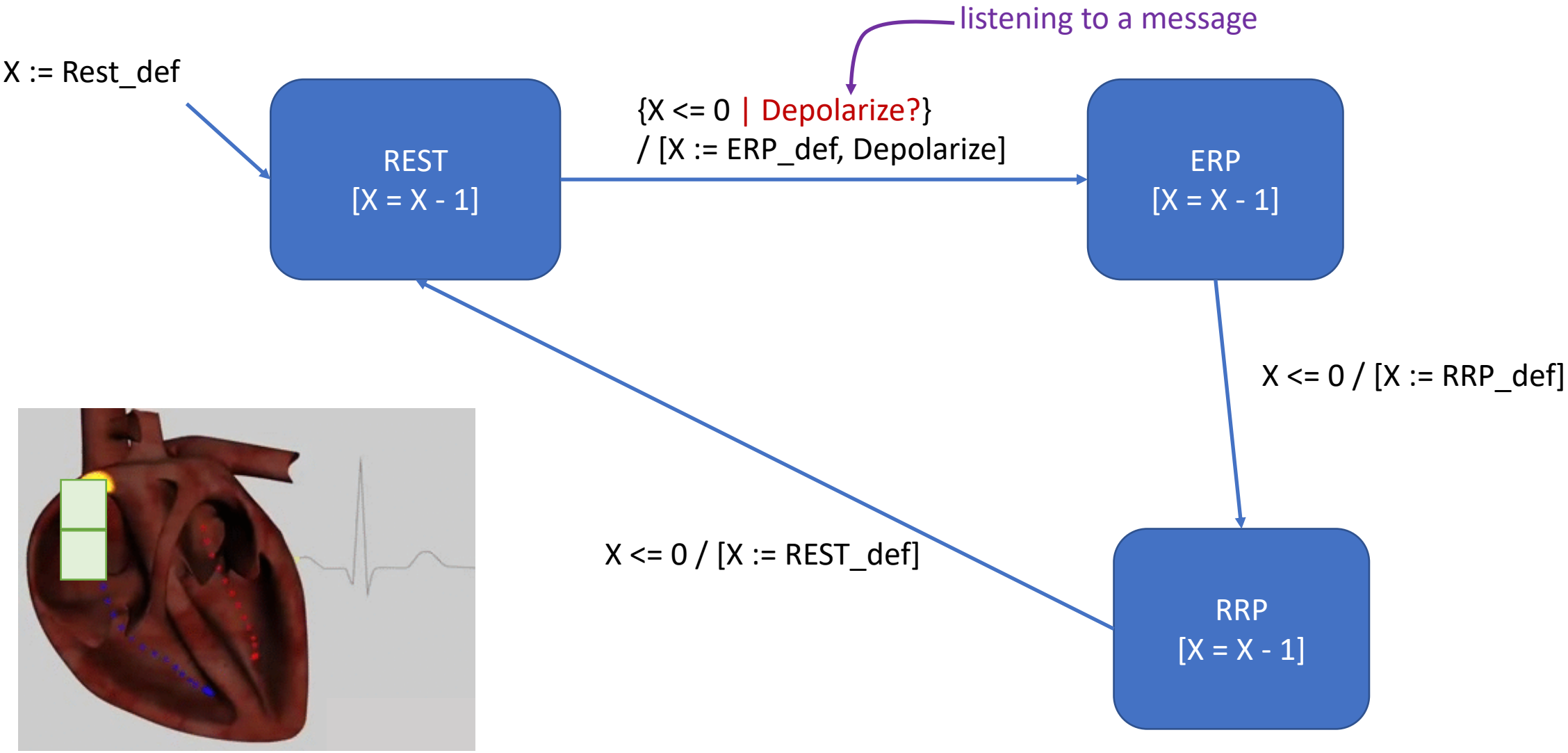
Node automaton with clocks, resets and actions



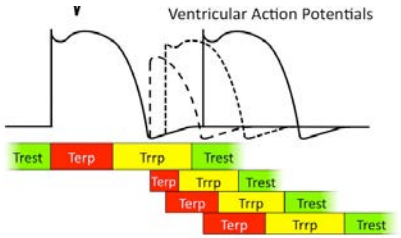
Node automaton with clocks, resets and actions



Node automaton with clocks, resets and actions receiving an action (message)



Node automaton with clocks, resets and actions receiving an action (message)



X := Rest_def

REST
[X = X - 1]

{X <= 0 | Depolarize?}
/ [X := ERP_def, Depolarize]

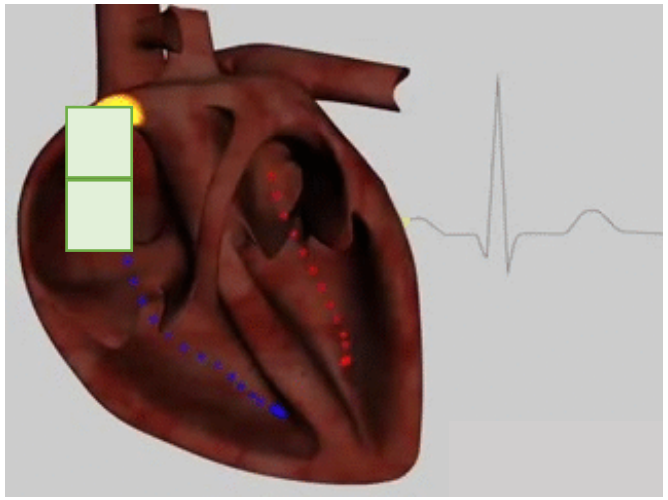
ERP
[X = X - 1]

{Depolarize?}
/ [X := ERP_def, Depolarize]

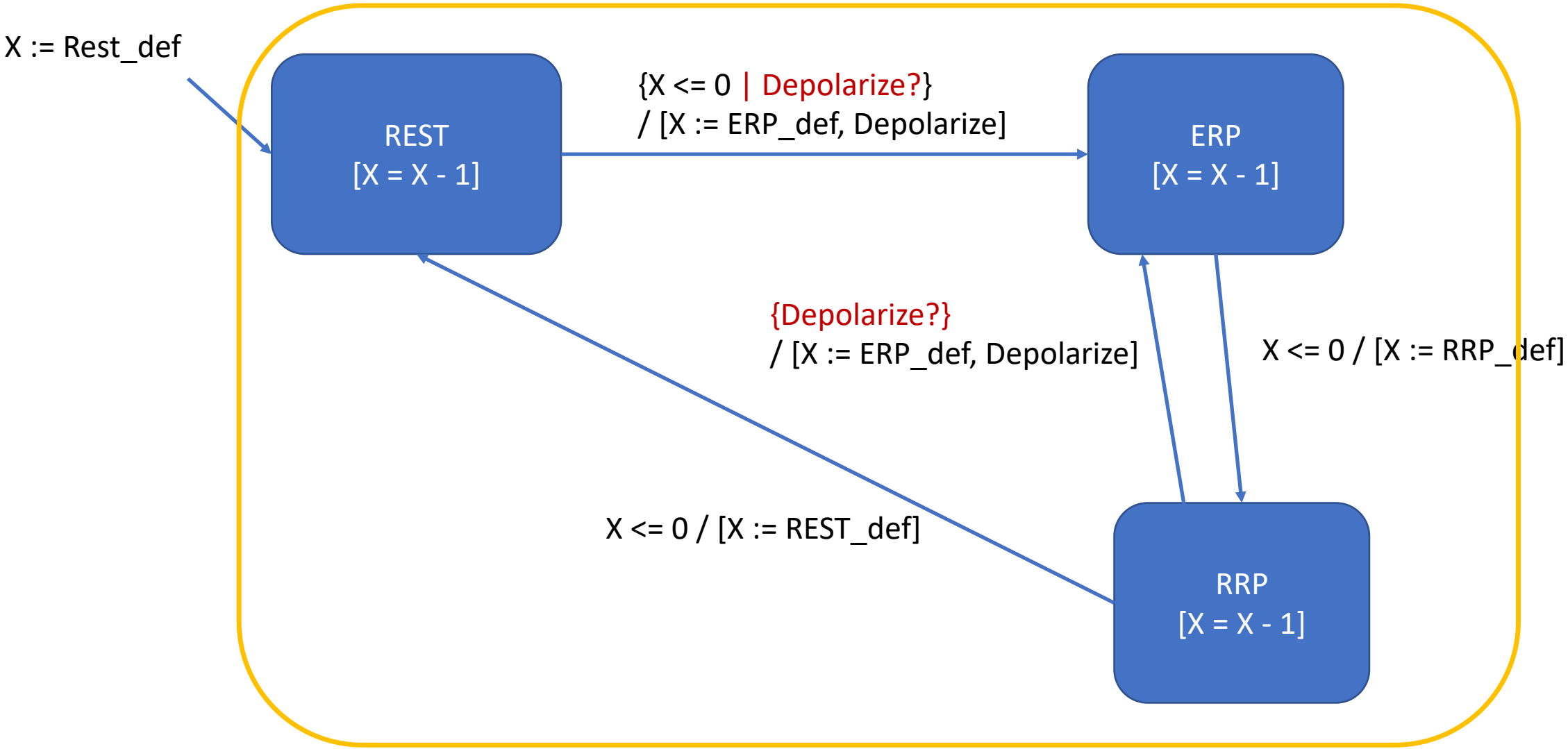
X <= 0 / [X := RRP_def]

X <= 0 / [X := REST_def]

RRP
[X = X - 1]

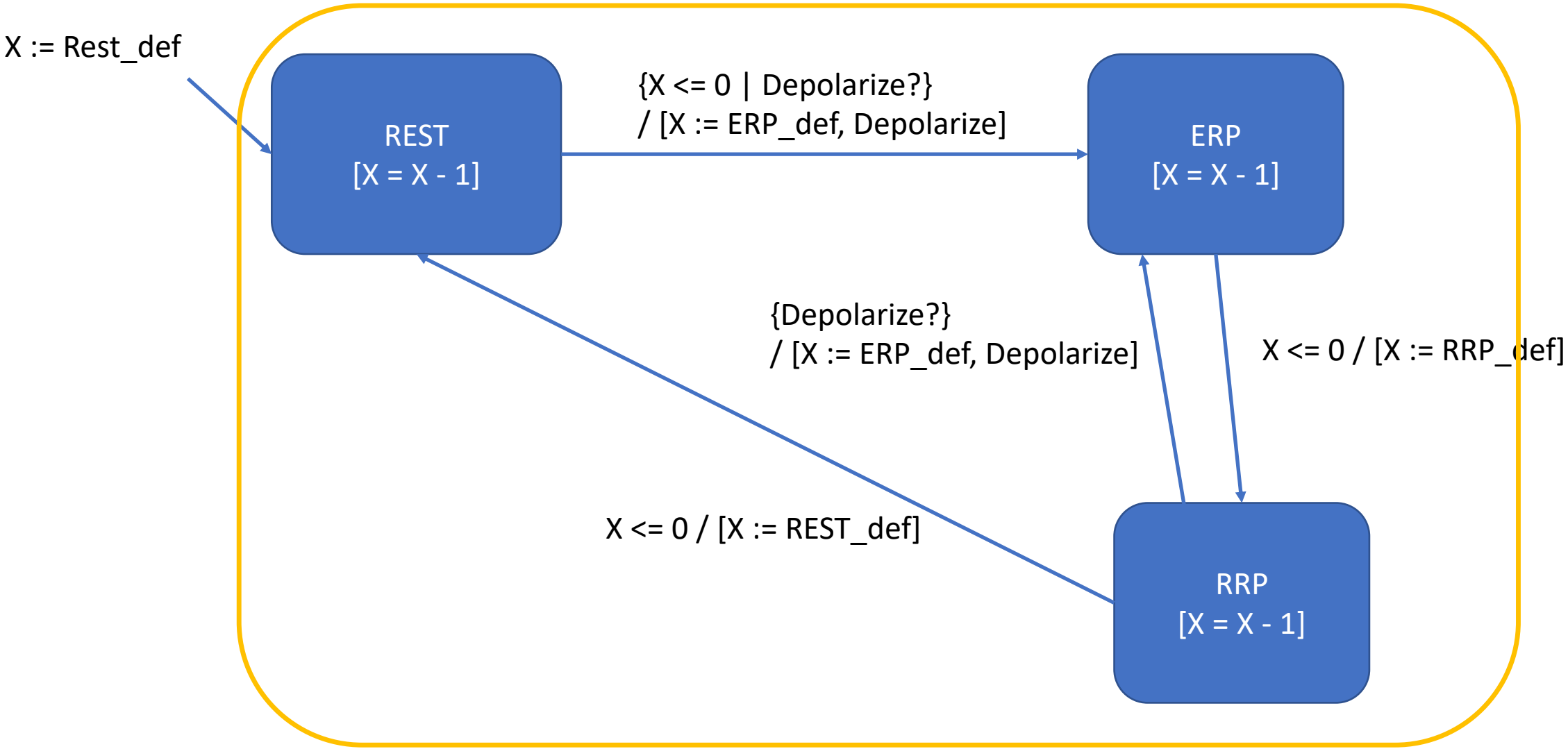


Node automaton with clocks, resets and actions

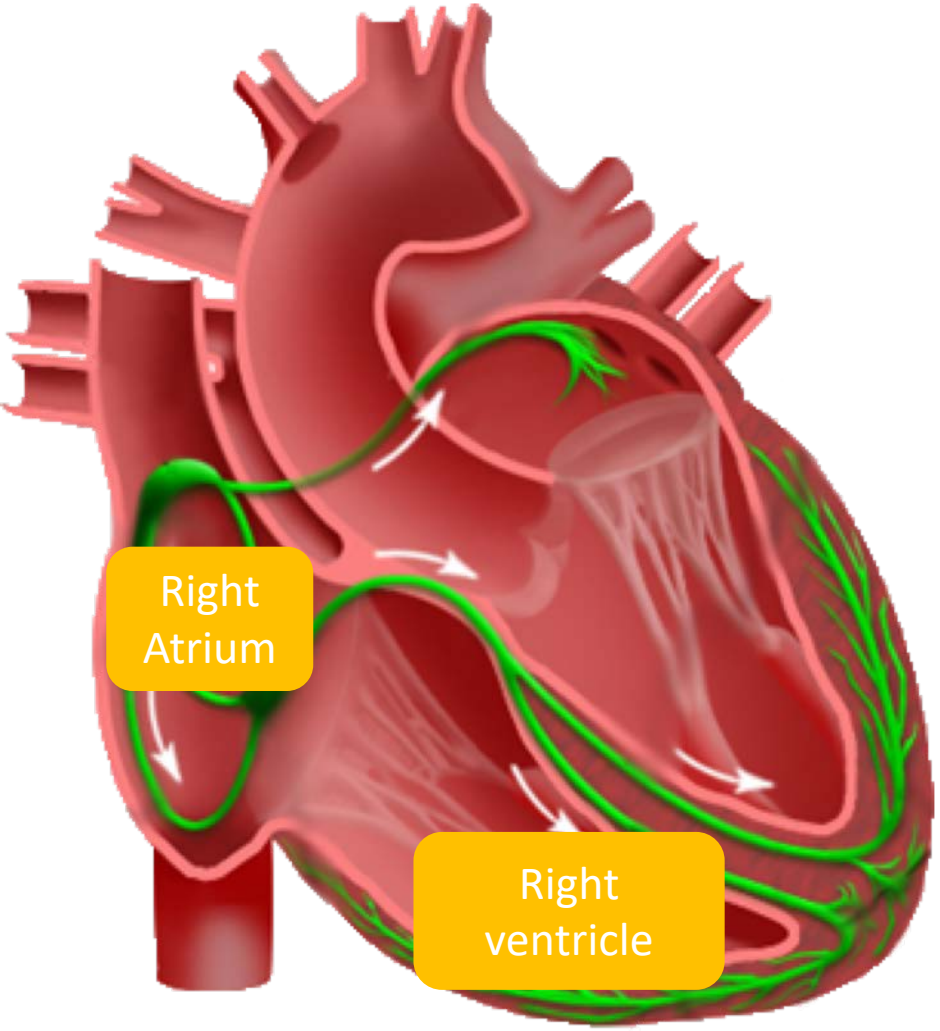


Node automaton with clocks, resets and actions

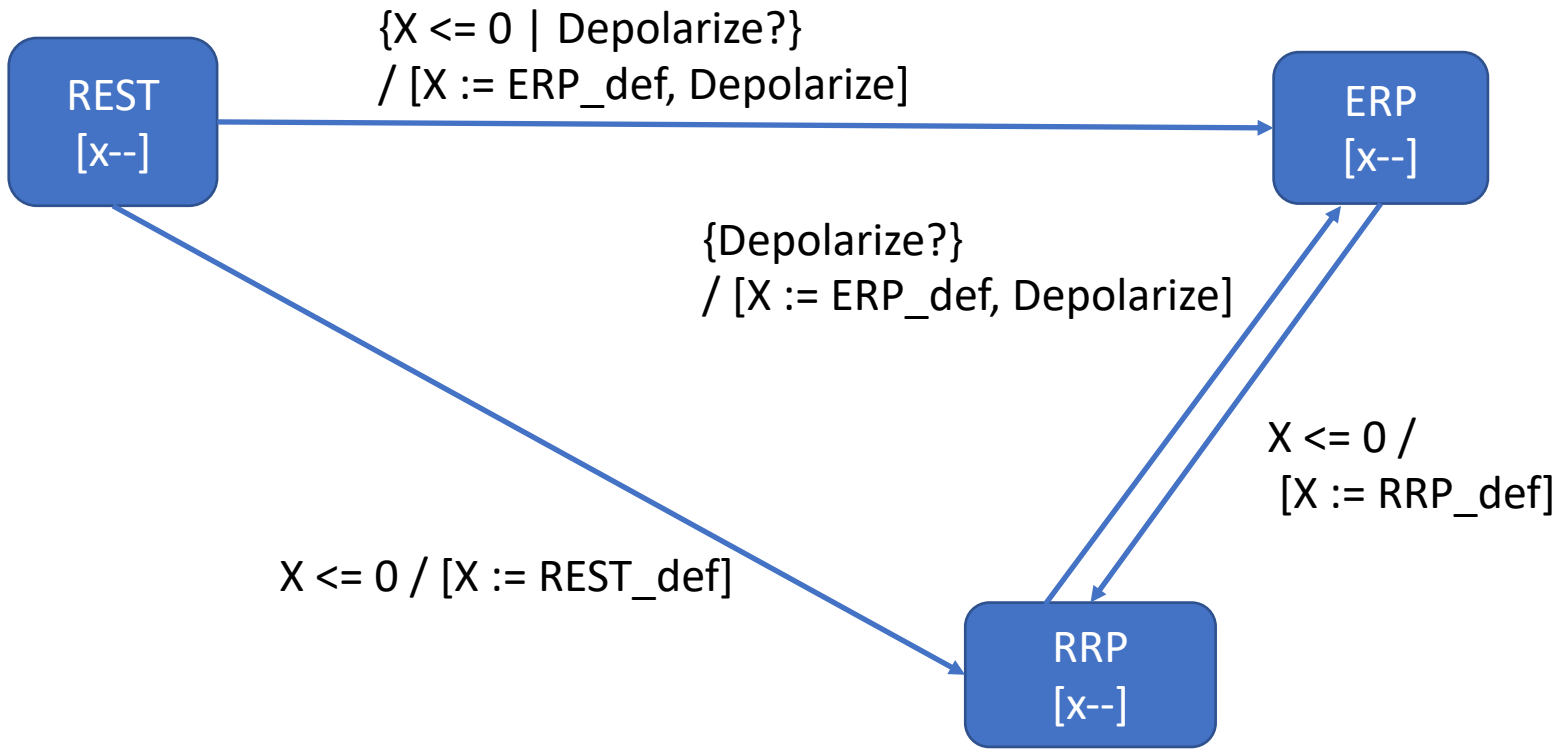
Timed Automaton



Different tissues with different properties = nodes with different parameters



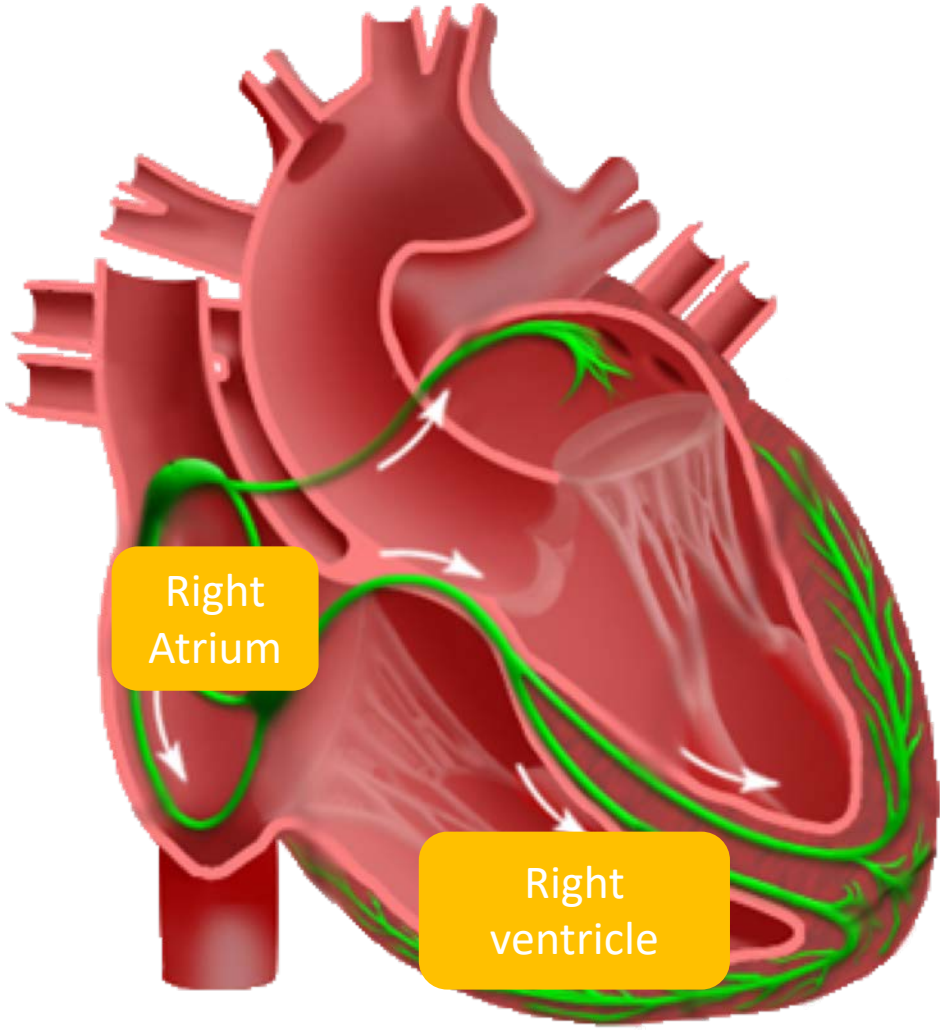
What parameters might be different between High Right Atrium (HRA) and Right Ventricular Apex (RVA)?

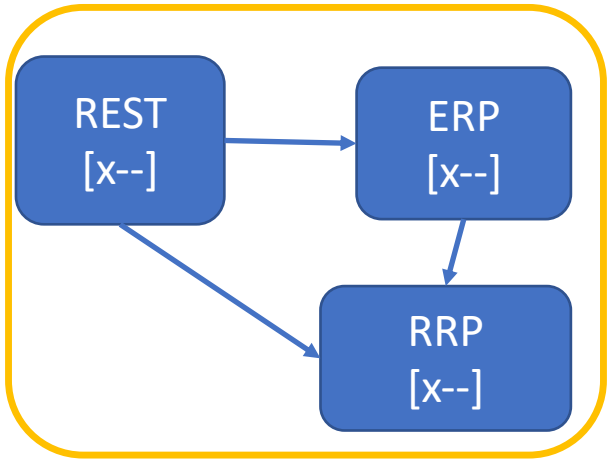


Different tissues with different properties = nodes with different parameters

Atrium must depolarize faster than everybody → Atrium.Rest_def < Ventricle.Rest_def

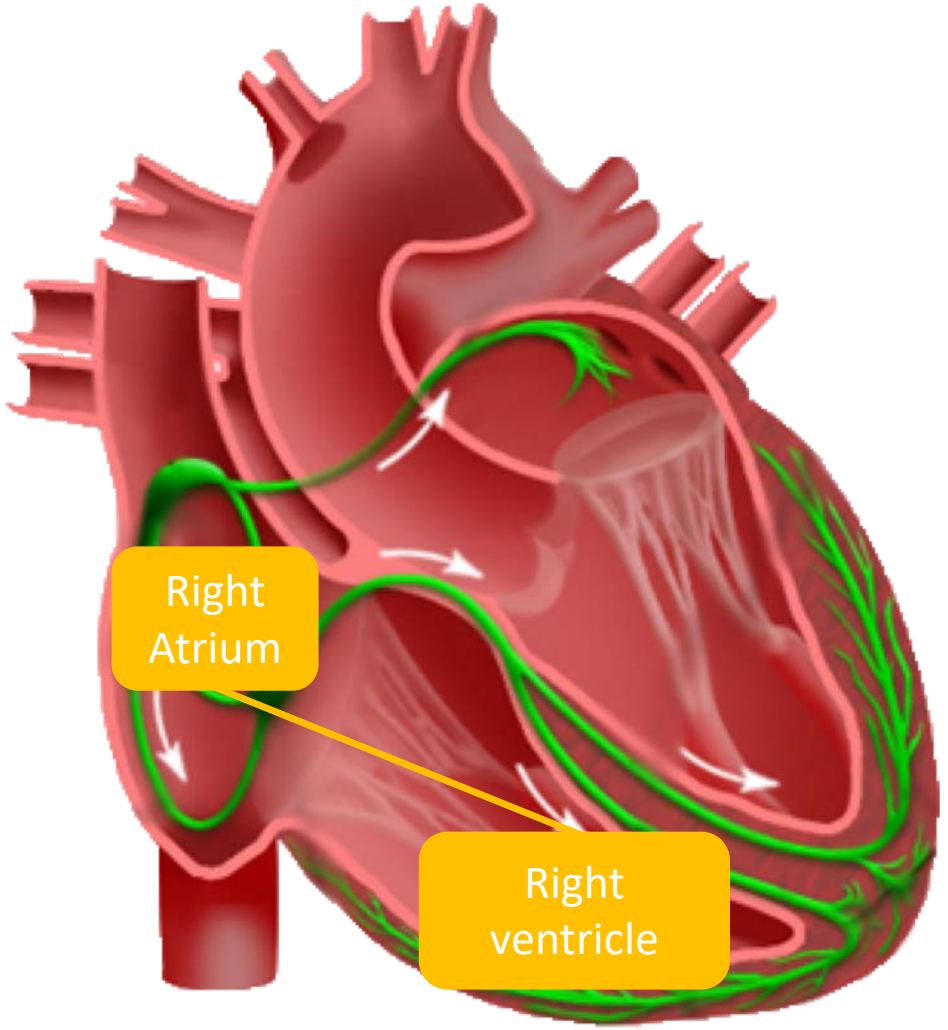
E.g. if the modeled heart rate is 60 beats per minute (bpm), atrium fires every 1 sec → Atrium.Rest_def = 1sec. Can choose Ventricle.Rest_def = 10secs, say.

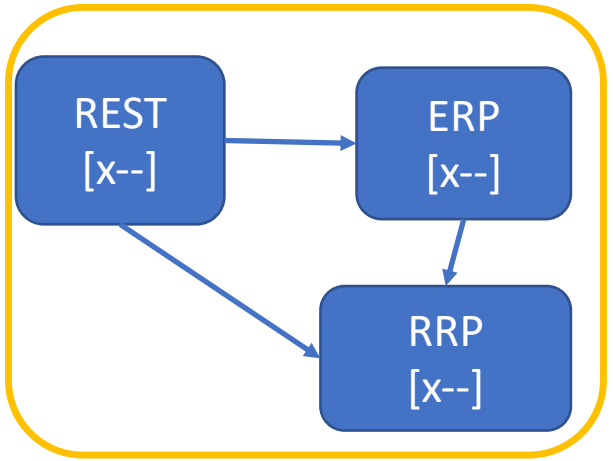




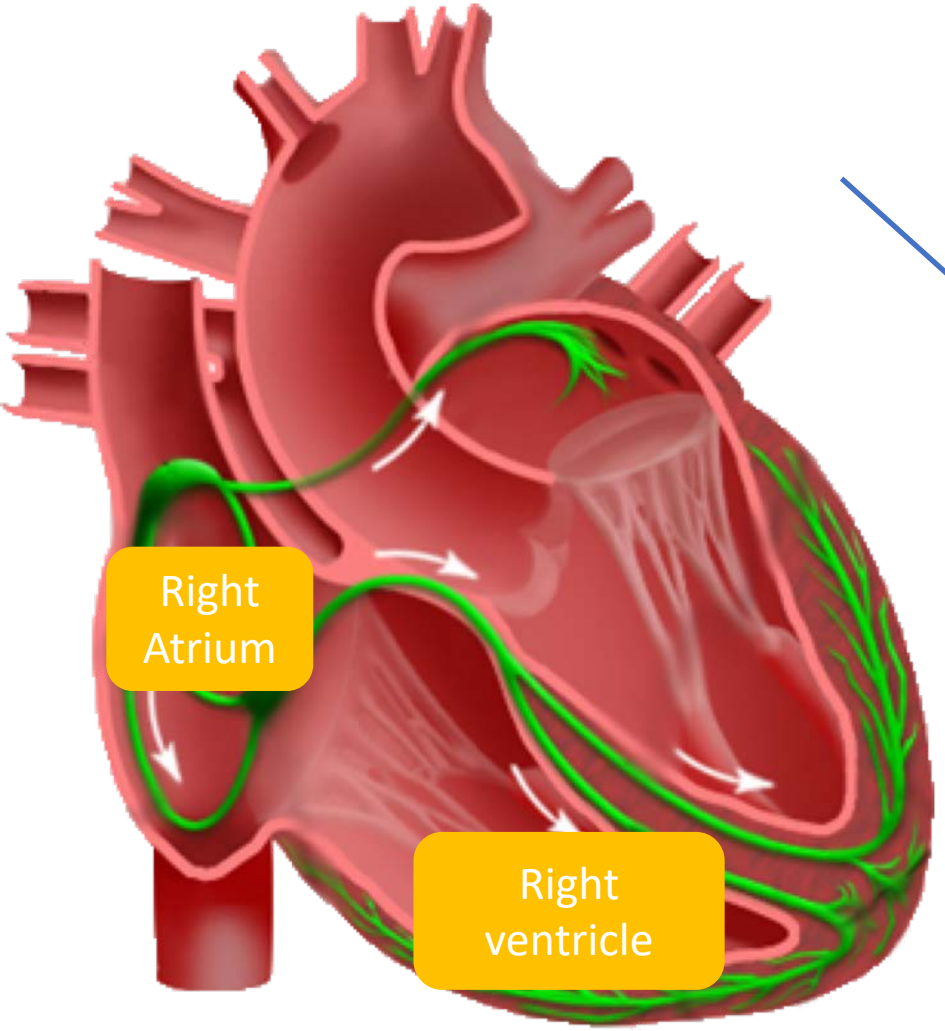
Connecting two tissues = connecting two nodes

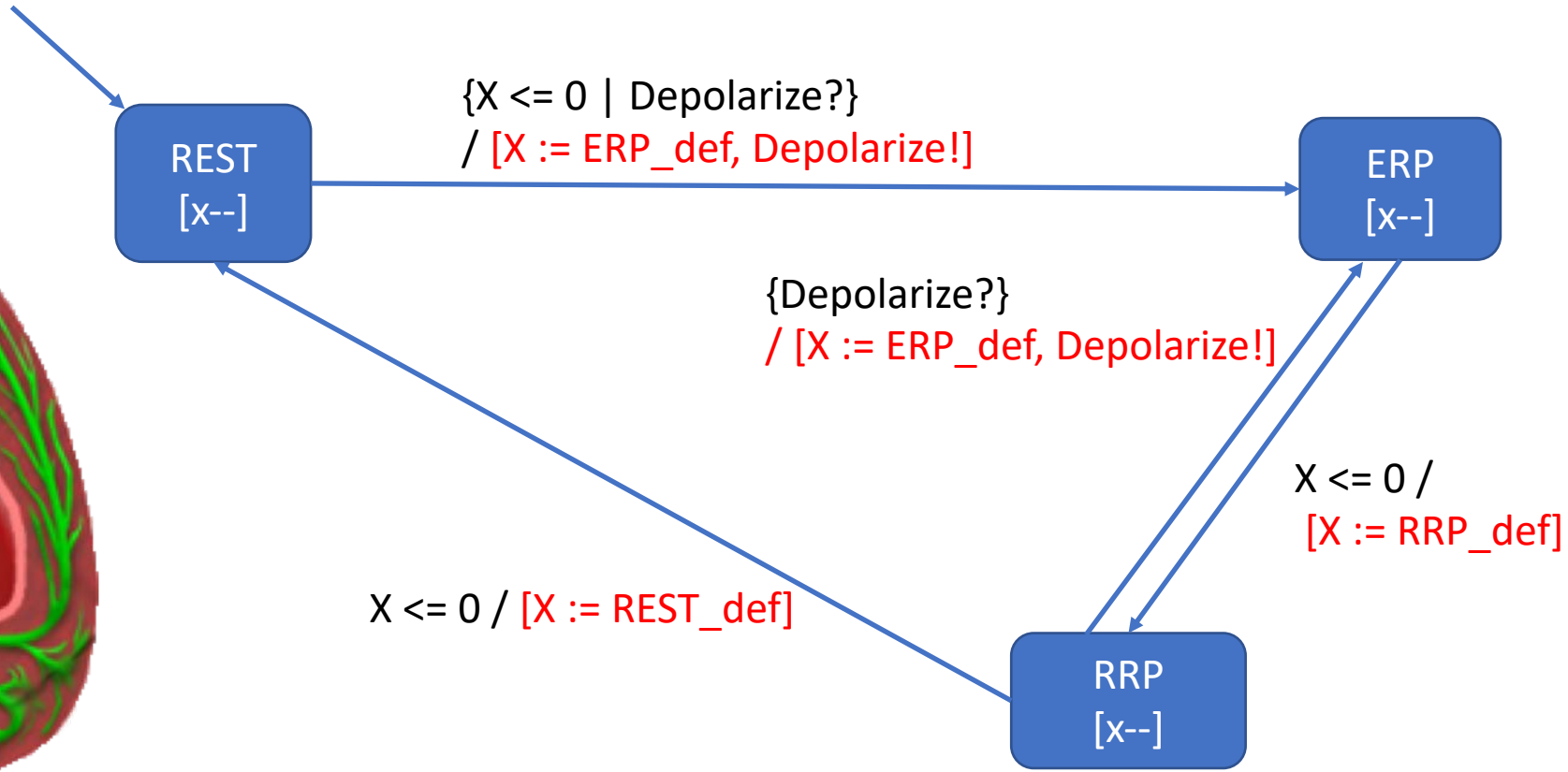
How do we model the propagation of activity, i.e. of event [Depolarize], from Atrium to Ventricle?





Communicating between timed automata





REST
[x--]

{X <= 0 | Depolarize?}
/ [X := ERP_def, Depolarize!]

ERP
[x--]

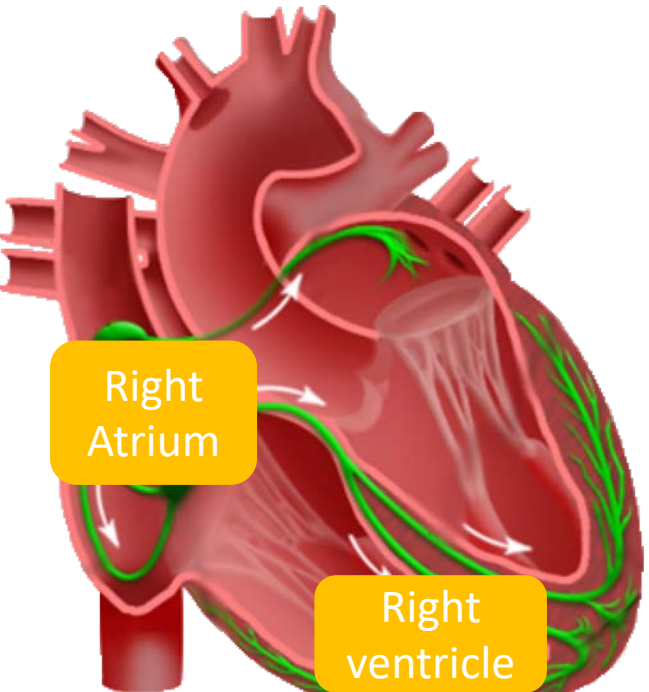
SA Node

{Depolarize?}
/ [X := ERP_def, Depolarize!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]



REST
[x--]

{X <= 0 | Depolarize?}
/ [X := ERP_def, Depolarize!]

ERP
[x--]

{Depolarize?}
/ [X := ERP_def, Depolarize!]

X <= 0 /
[X :=
RRP_def]

X <= 0 / [X := REST_def]

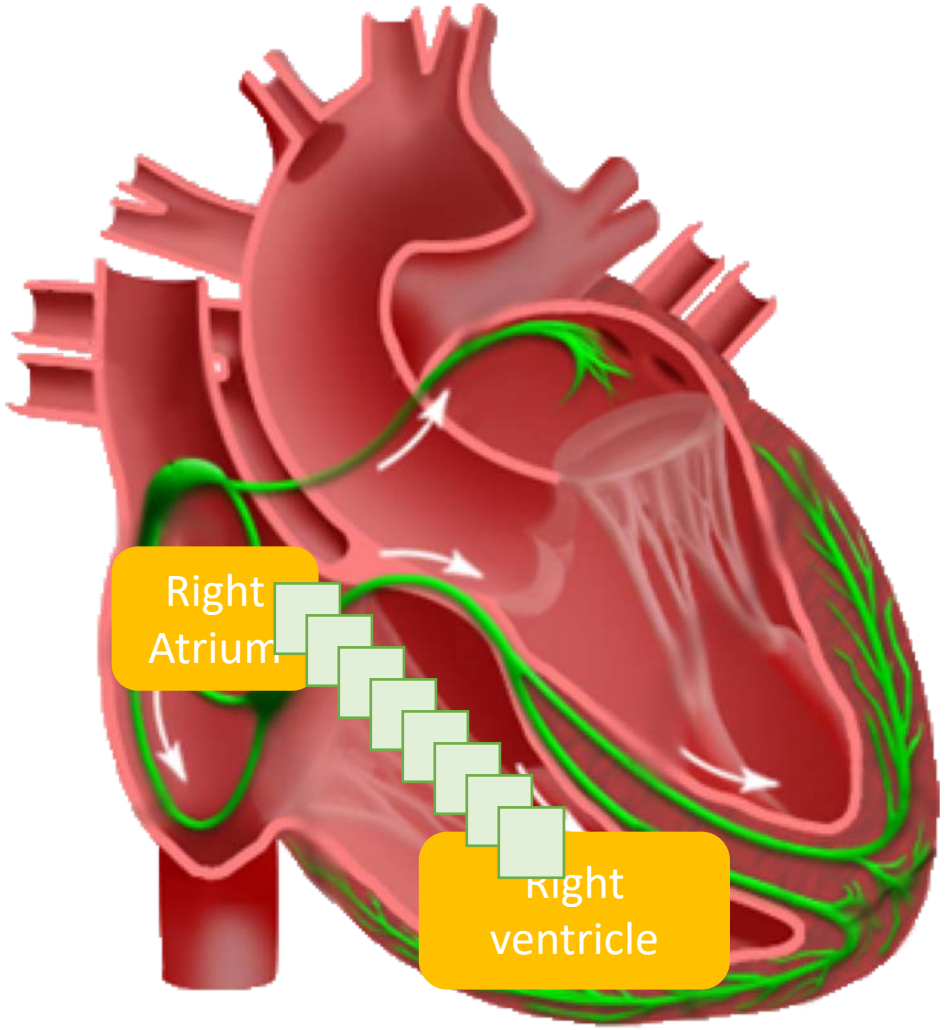
RRP
[x--]

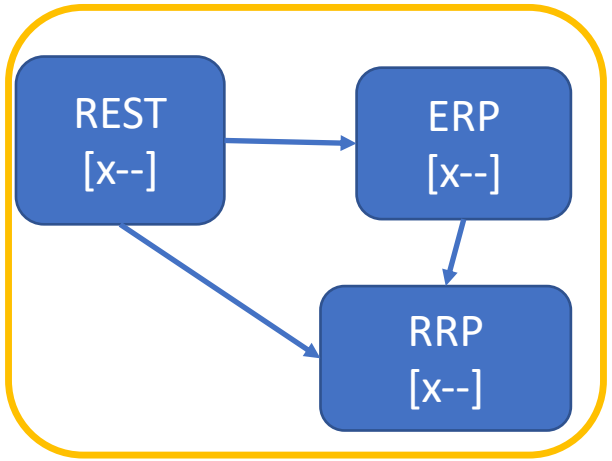
Right ventricle

Connecting two tissues = connecting two nodes

How do we model the propagation of activity, i.e. of event [Depolarize], from Atrium to Ventricle?

A large # of cells between the two…

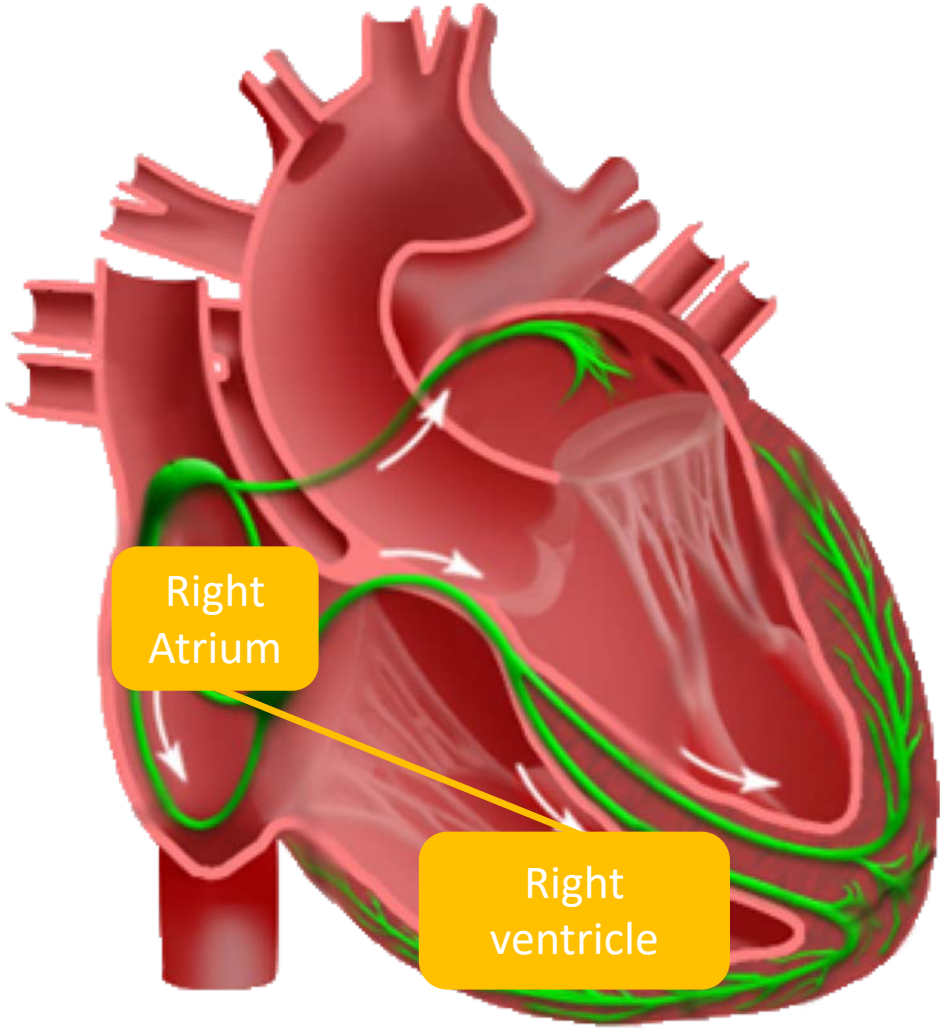


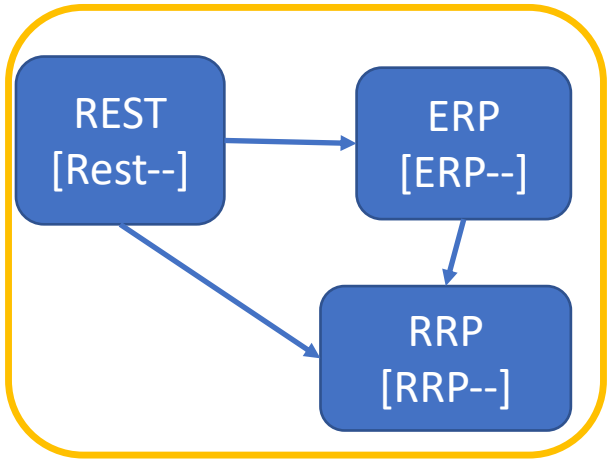


Connecting two tissues = connecting two nodes

How do we model the propagation of activity, i.e. of event [Depolarize], from Atrium to Ventricle *as a timed automaton*?

Stick to same formalism as much as possible: Different formalisms add complexity to the analysis as you try to make two mathematical languages talk to each other.





REST
[x--]

{X <= 0 | Depolarize?}
/ [X := ERP_def, Depolarize!]

ERP
[x--]

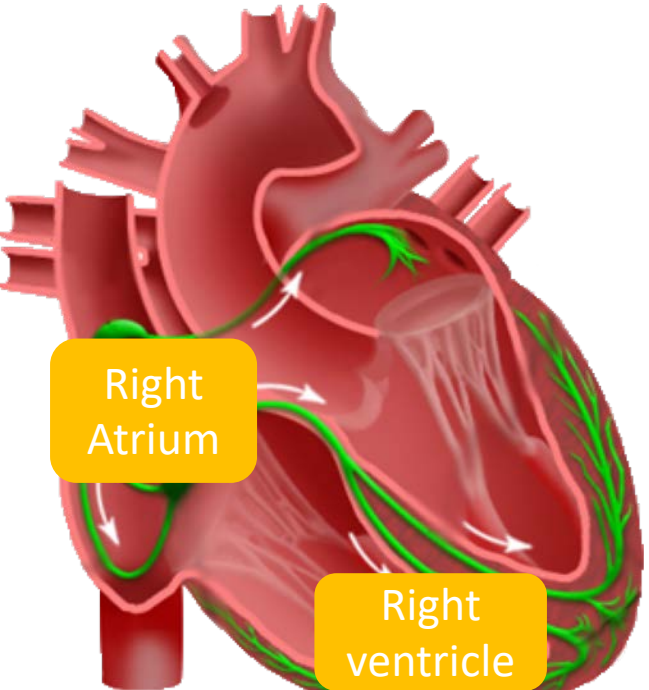
SA Node

{Depolarize?}
/ [X := ERP_def, Depolarize!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]



Delay
[x--]

{X <= 0?}
/ [Depolarize!]

REST
[x--]

{X <= 0 | Depolarize?}
/ [X := Delay_def!]

ERP
[x--]

{Depolarize?}
/ [X := ERP_def, Depolarize!]

X <= 0 /
[X :=
RRP_def]

X <= 0 / [X := REST_def]

RRP
[x--]

Right ventricle

REST
[x--]

{X <= 0 | Depolarize?}
/ [X := ERP_def, Depolarize!]

ERP
[x--]

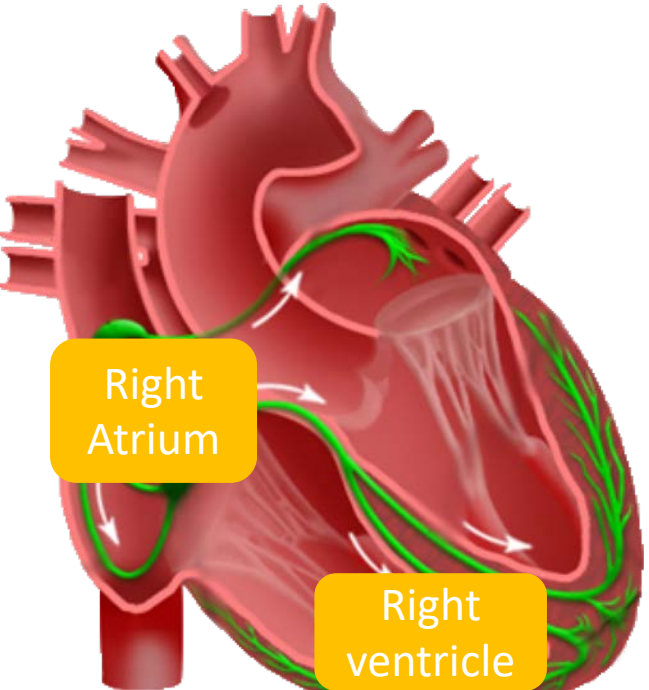
SA Node

{Depolarize?}
/ [X := ERP_def, Depolarize!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]



Delay
[x--]

<= 0?}
olarize!]

REST
[x--]

{X <=
/ [X := D

ERP
[x--]

e!]

X <= 0 / [X := RES _def]

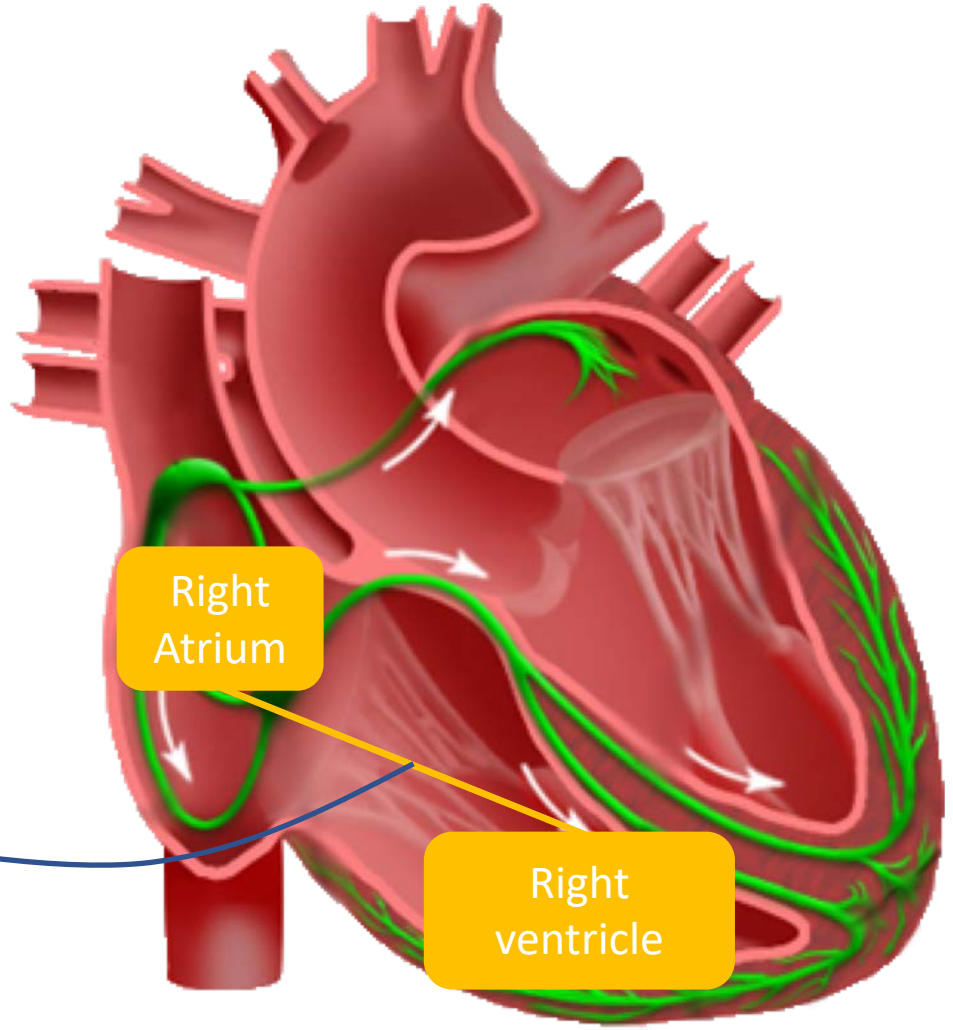
X <= 0 /
[X :=
RRP_def]

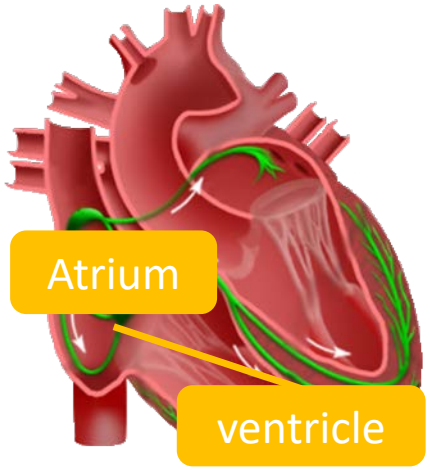
RRP
[x--]

Right ventricle

A timed automaton model of conduction paths

Model this conduction path (which is a piece of tissue) as a different kind of timed automaton.





IDLE

{Depolarize?}

SA Node

REST
[x--]

{X <= 0 | Depolarize?}
/ [X := ERP_def, Depolarize!]

ERP
[x--]

{Depolarize?}
/ [X := ERP_def, Depolarize!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

REST
[x--]

{X <= 0 | Depolarize?}
/ [X := ERP_def, Depolarize!]

ERP
[x--]

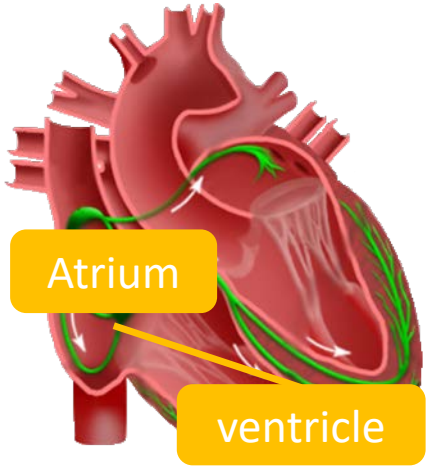
{Depolarize?}
/ [X := ERP_def, Depolarize!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

Right ventricle



IDLE

{Depolarize?}

CONDUCT

SA Node

REST
[x--]

{X <= 0 | Depolarize?}
/ [X := ERP_def, Depolarize!]

ERP
[x--]

{Depolarize?}
/ [X := ERP_def, Depolarize!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

REST
[x--]

{X <= 0 | Depolarize?}
/ [X := ERP_def, Depolarize!]

ERP
[x--]

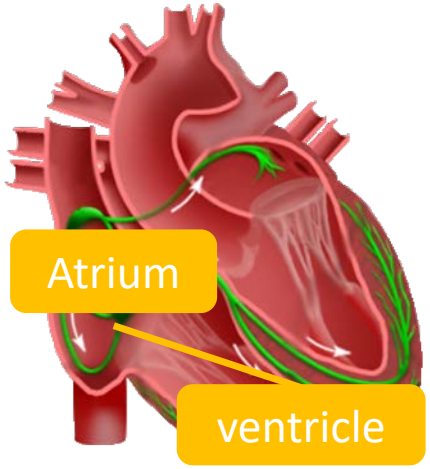
{Depolarize?}
/ [X := ERP_def, Depolarize!]

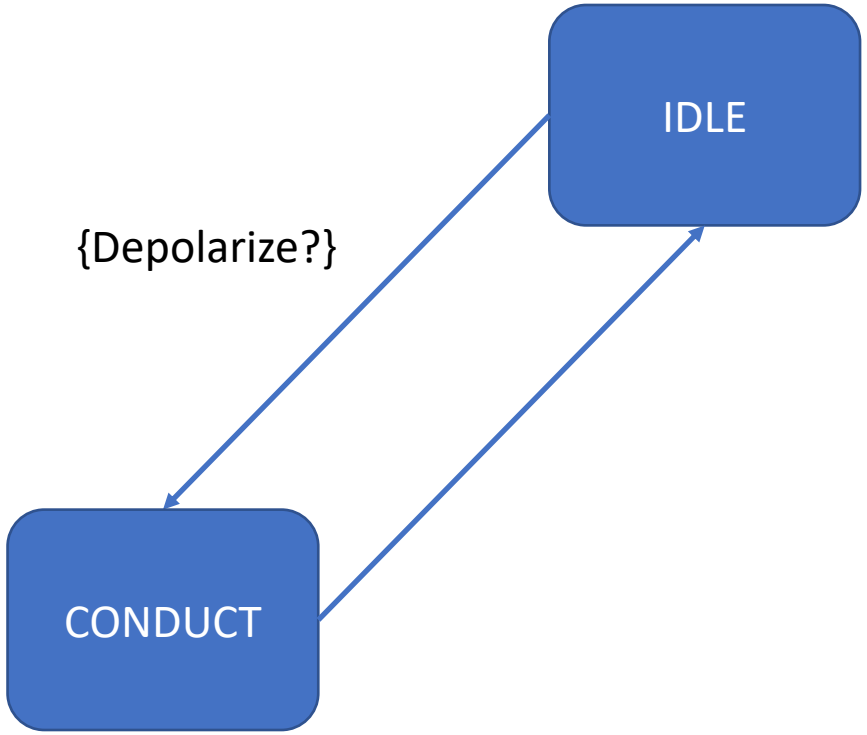
X <= 0 / [X := REST_def]

RRP
[x--]

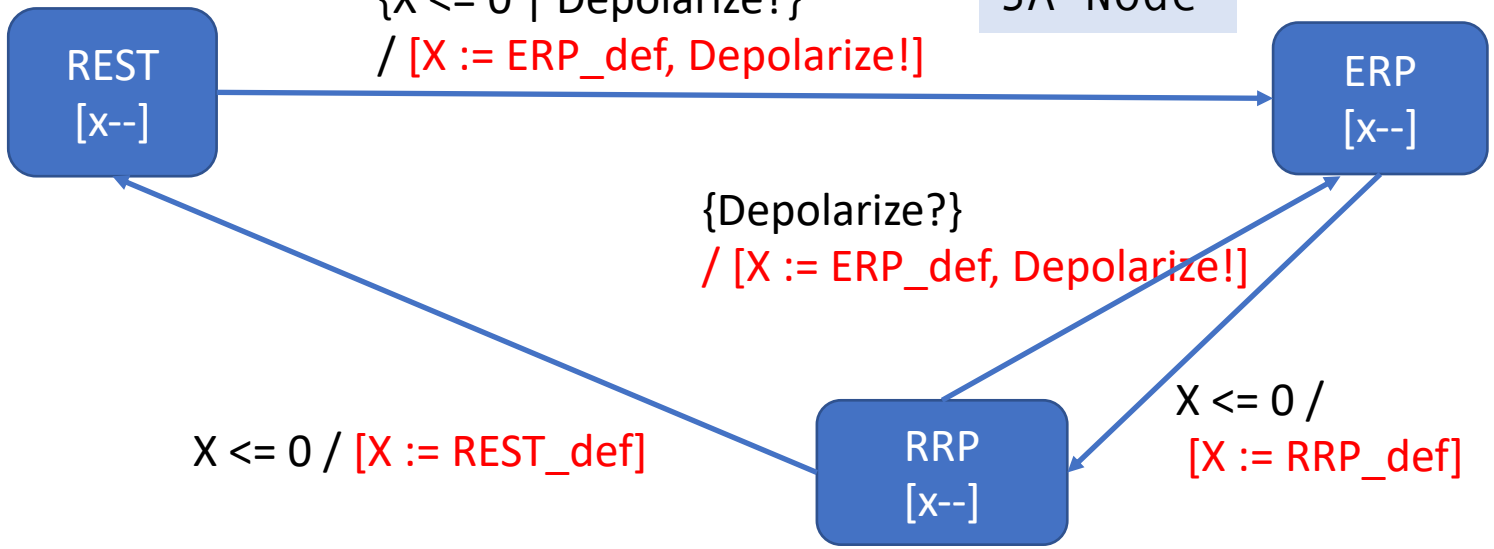
X <= 0 /
[X := RRP_def]

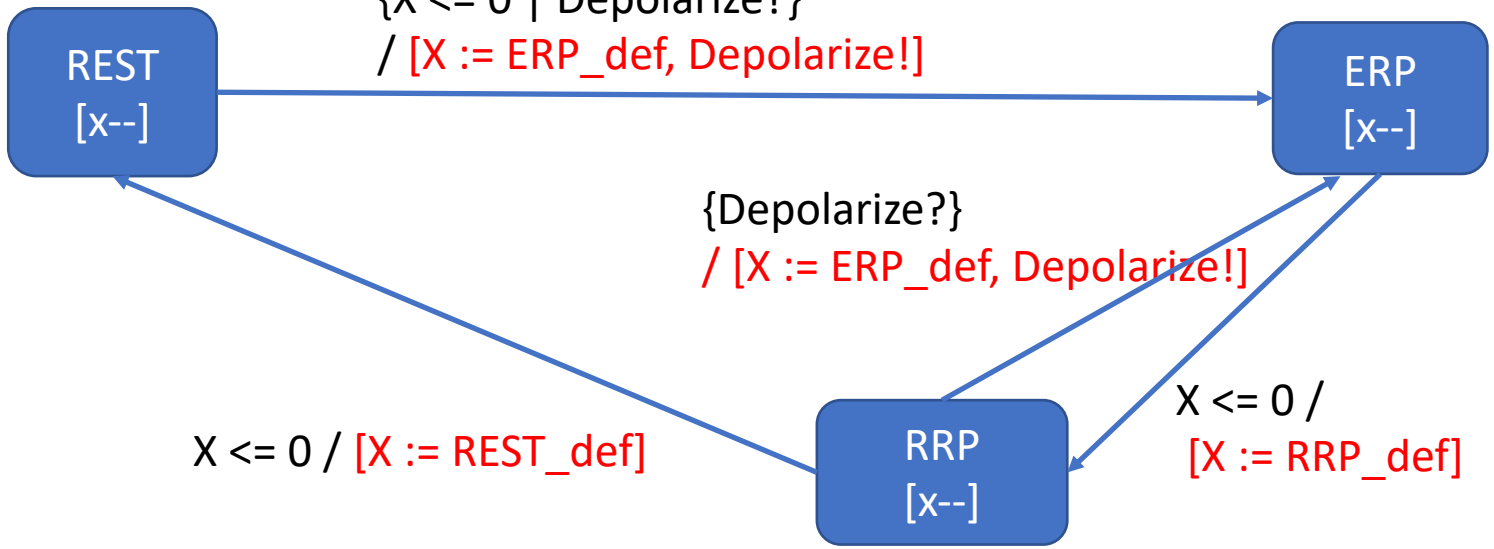
Right ventricle



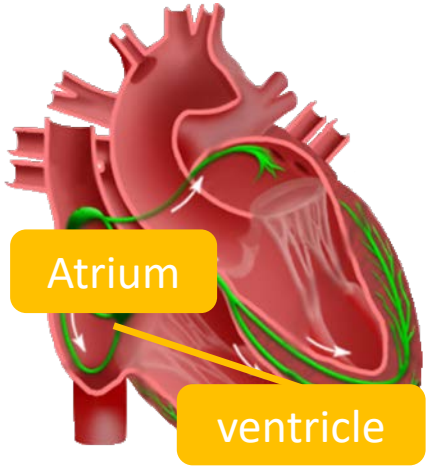


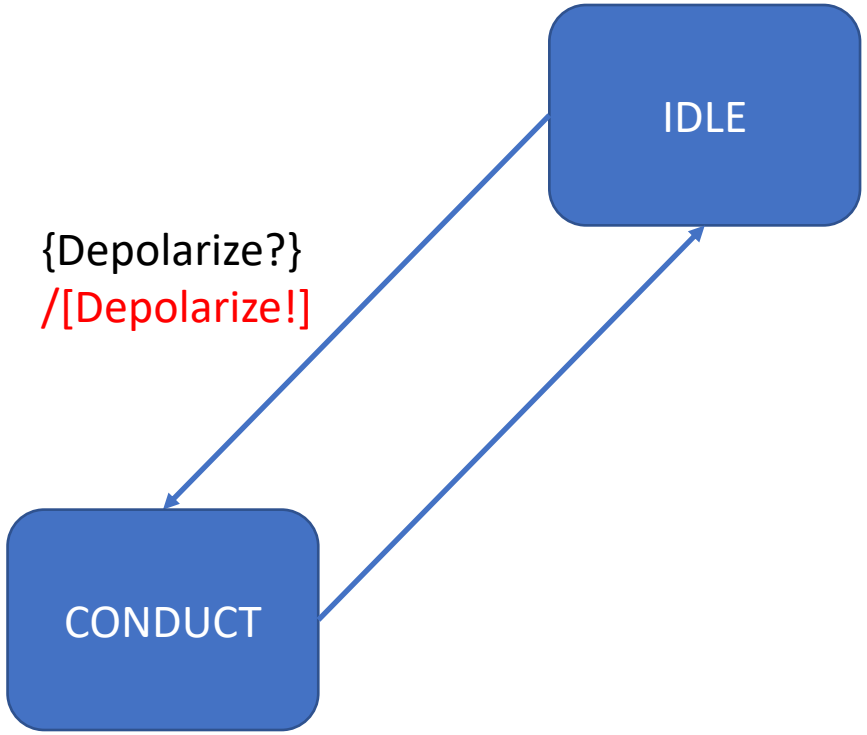
SA Node



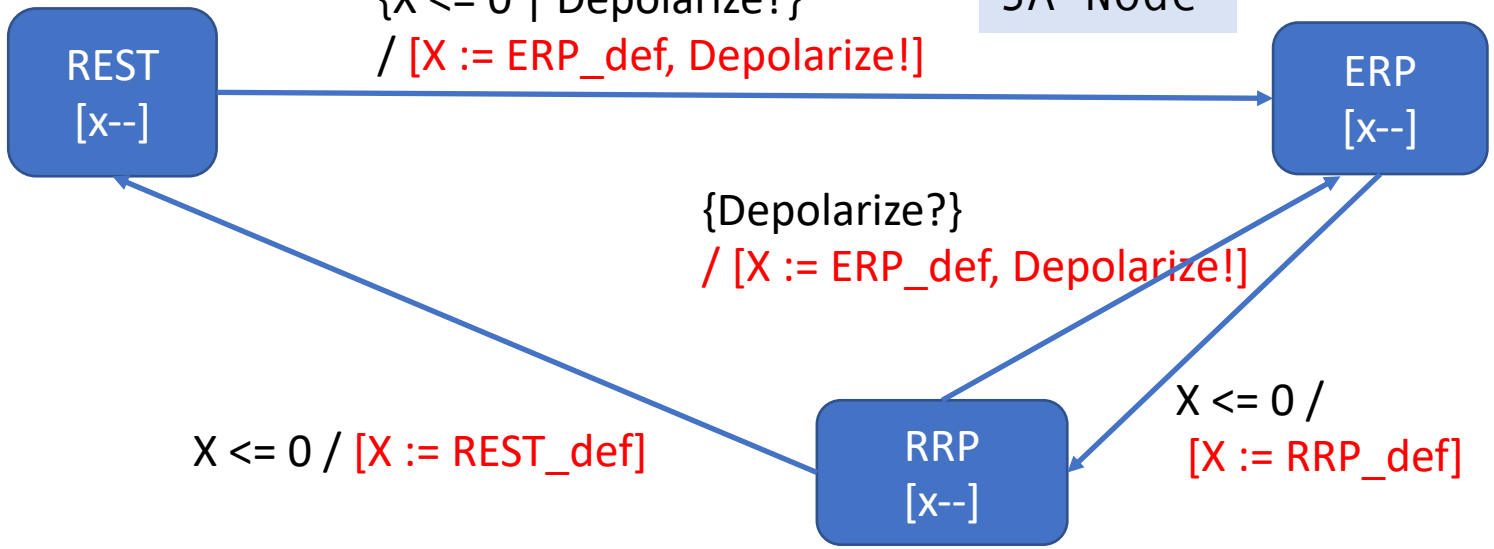


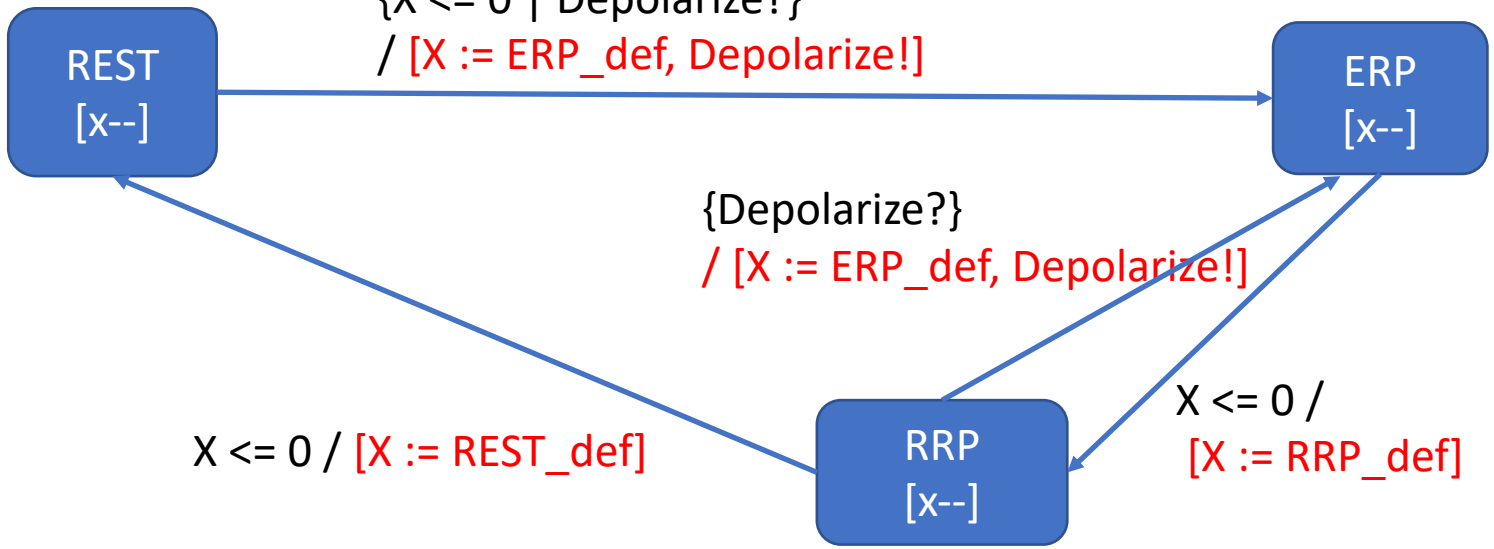
Right ventricle



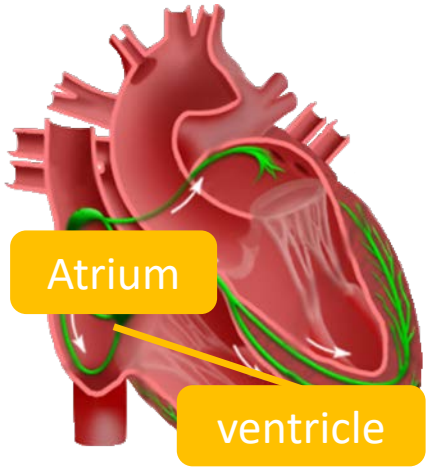


SA Node





Right ventricle



SA Node

REST
[x--]

{X <= 0 | Depolarize?}
/ [X := ERP_def, Depolarize!]

ERP
[x--]

{Depolarize?}
/ [X := ERP_def, Depolarize!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

IDLE

{Depolarize?}
/[Act_node2!]

CONDUCT

REST
[x--]

{X <= 0 | Act_node2?}
/ [X := ERP_def, Depolarize!]

ERP
[x--]

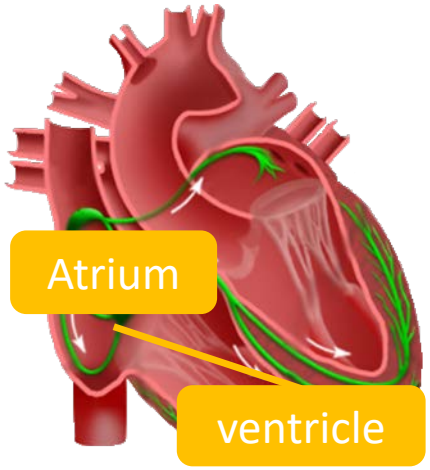
{Act_node2?}
/ [X := ERP_def, Depolarize!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

Right ventricle



SA Node

REST
[x--]

{X <= 0 | Depolarize?}
/ [X := ERP_def, Depolarize!]

ERP
[x--]

{Depolarize?}
/ [X := ERP_def, Depolarize!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

IDLE

{Depolarize?}
/[X=Tcond_def!]

CONDUCT
[x--]

{X<=0?}
/[Act_node2!]

REST
[x--]

{X <= 0 | Act_node2?}
/ [X := ERP_def, Depolarize!]

ERP
[x--]

{Act_node2?}
/ [X := ERP_def, Depolarize!]

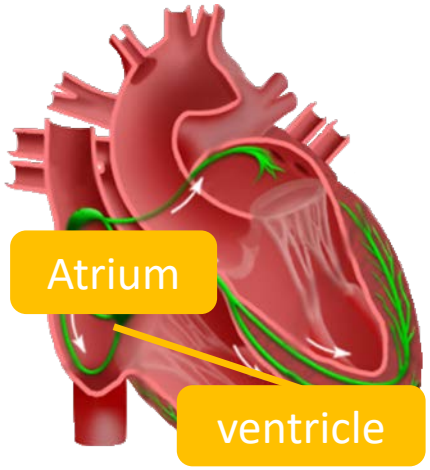
X <= 0 / [X := REST_def]

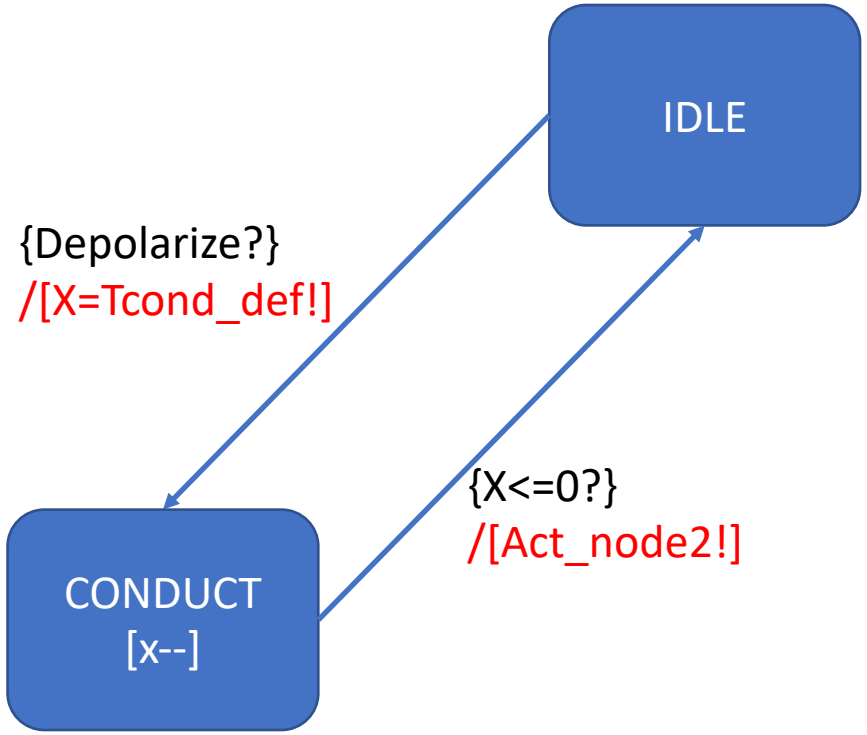
RRP
[x--]

X <= 0 /
[X := RRP_def]

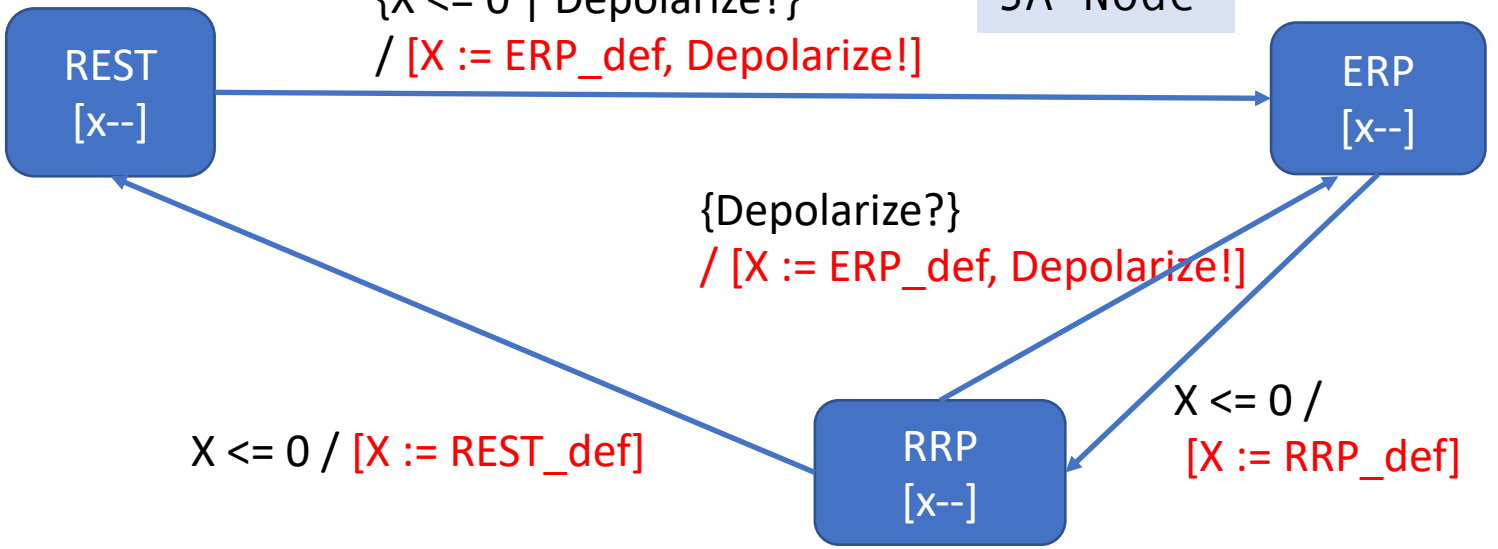
Right ventricle

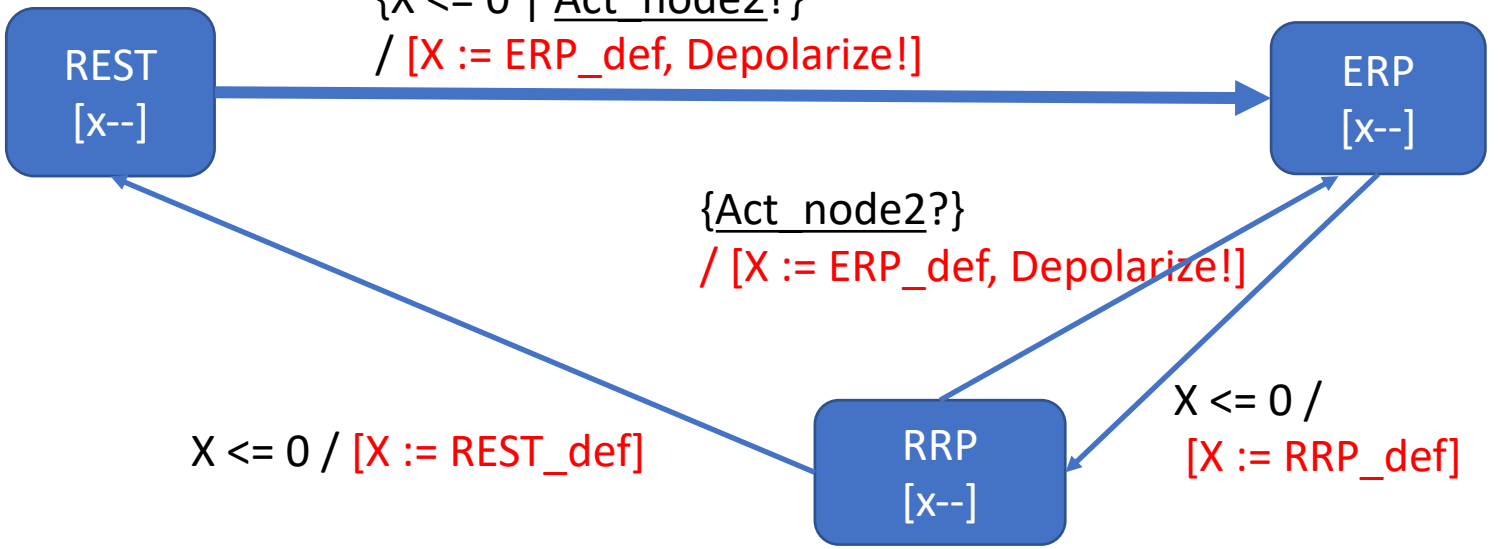
Retrograde Conduction





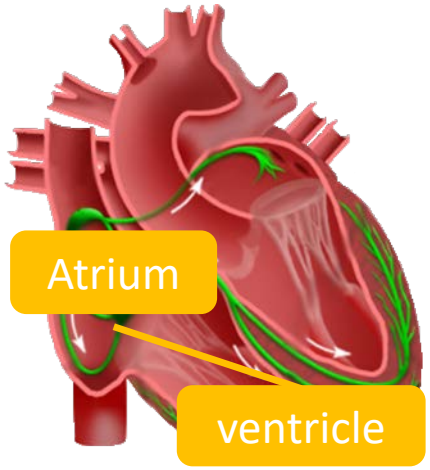
SA Node

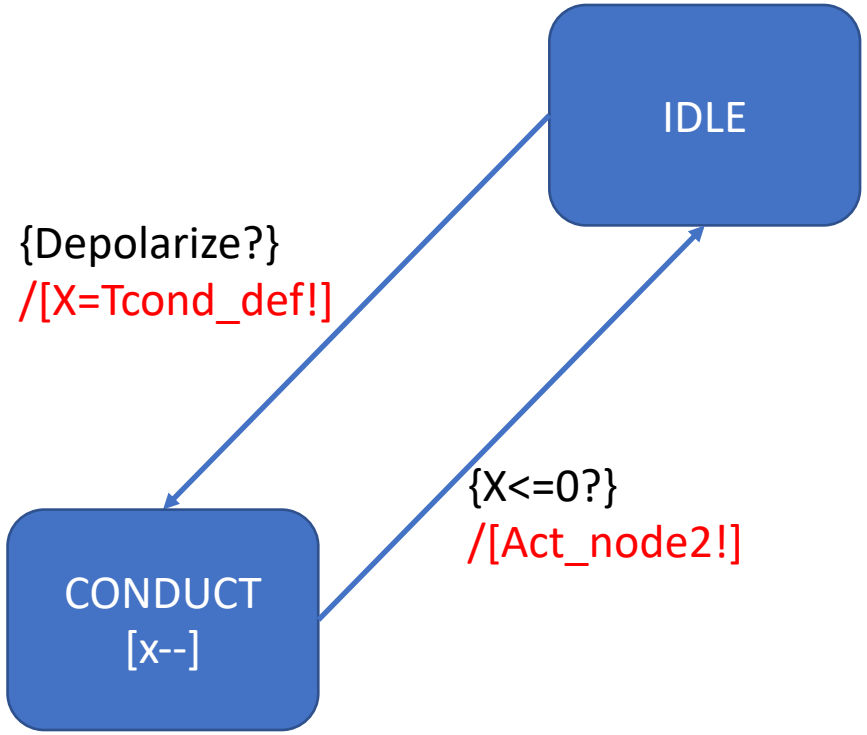




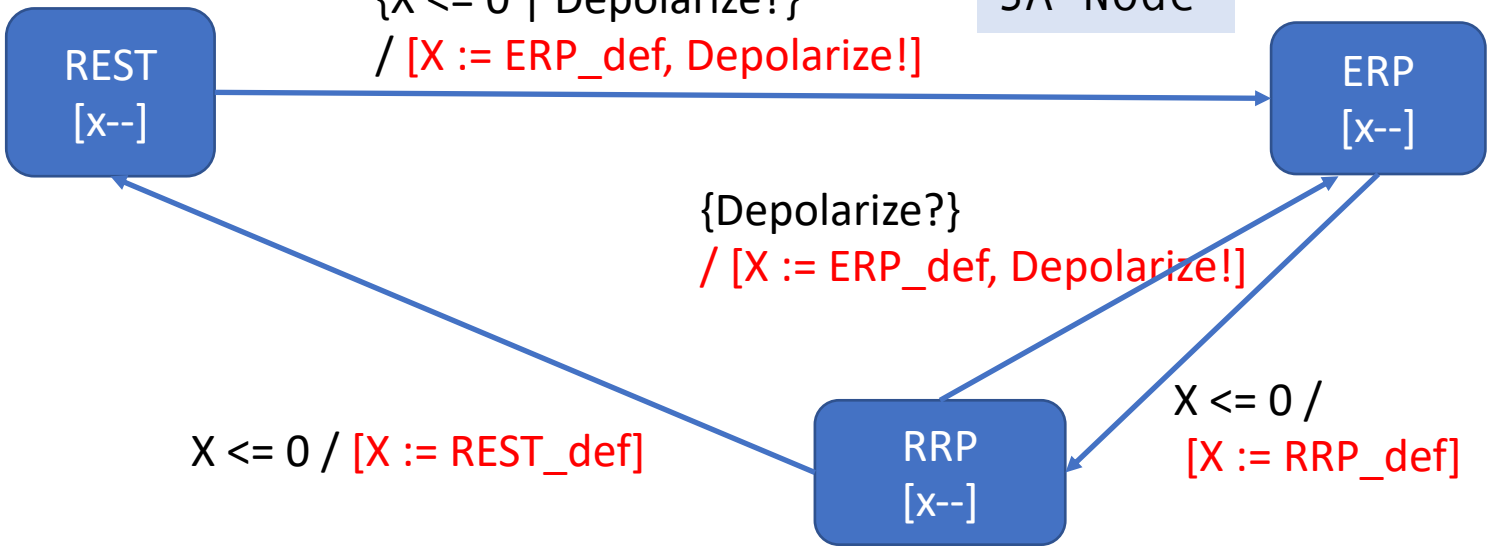
Right ventricle

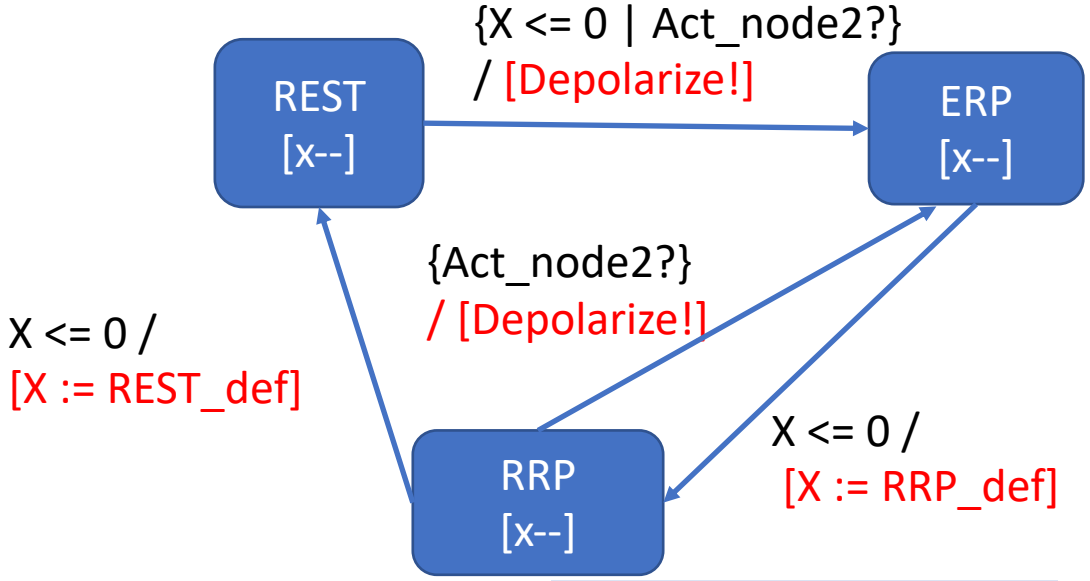
Retrograde Conduction





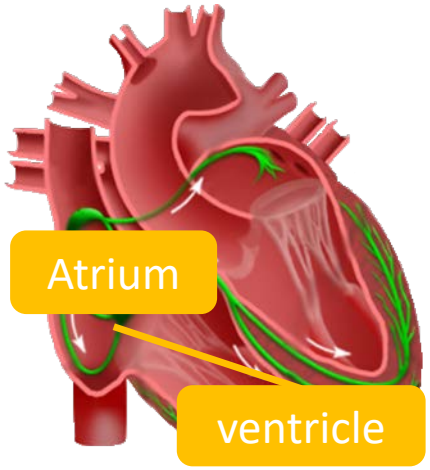
SA Node



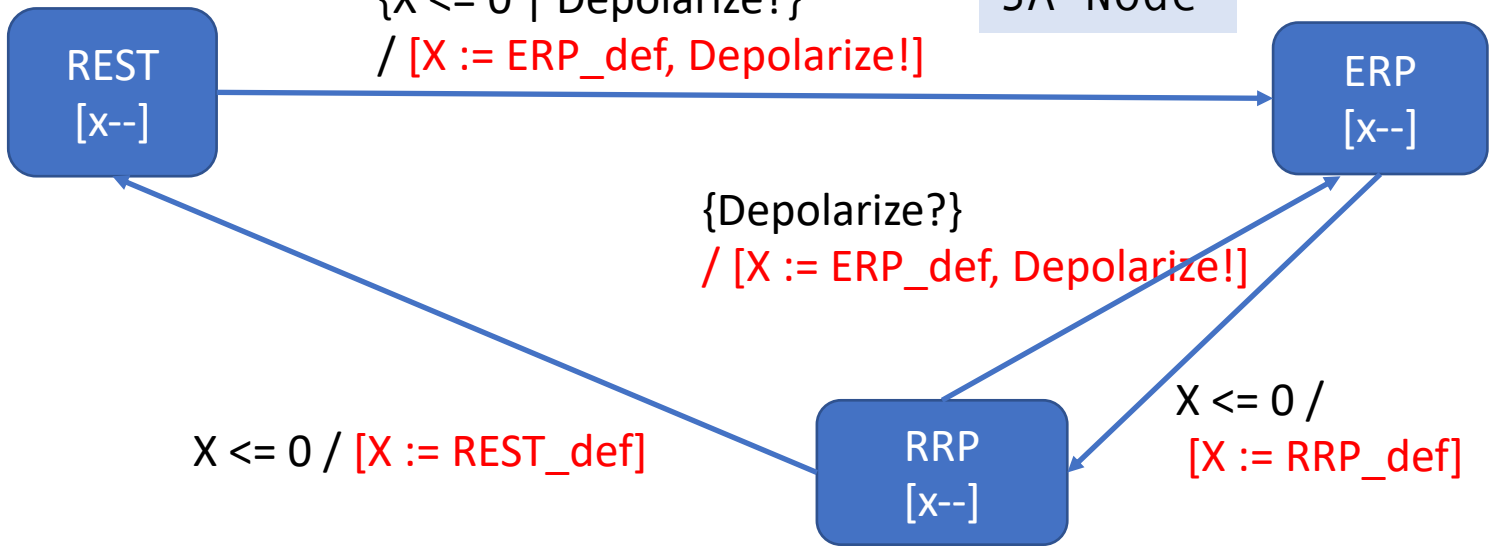


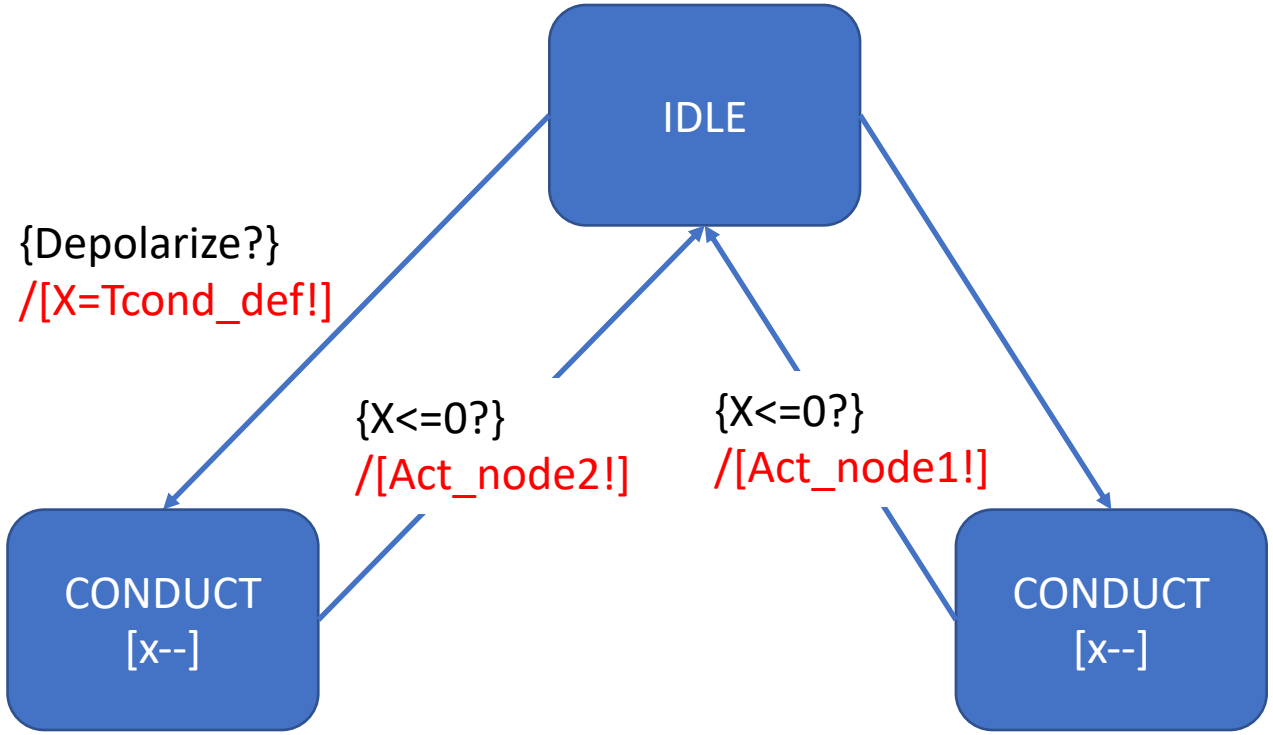
Right ventricle

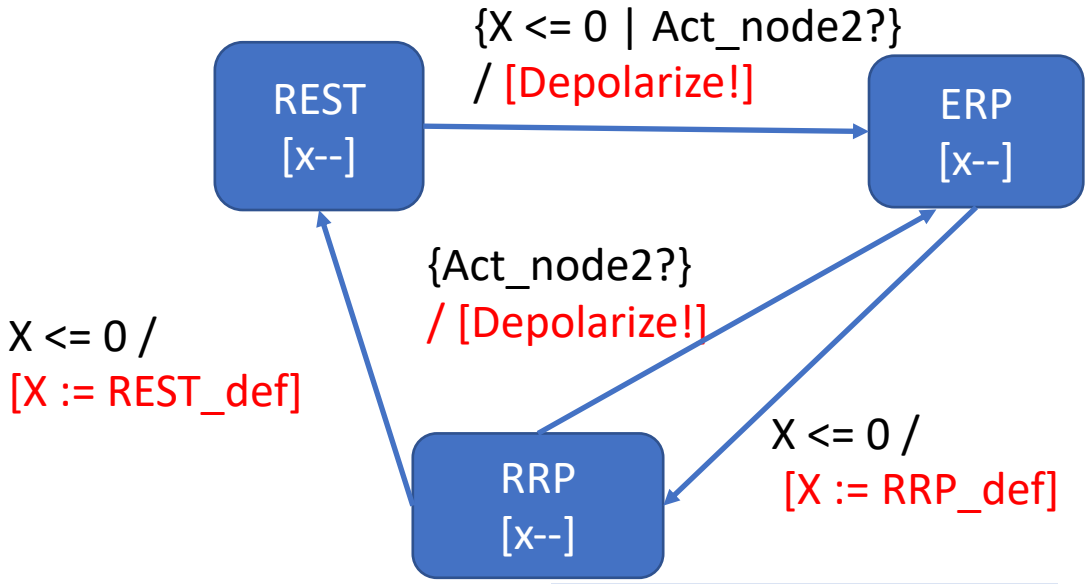
Retrograde Conduction



SA Node

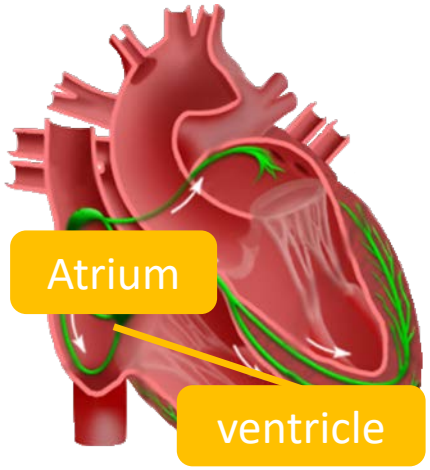




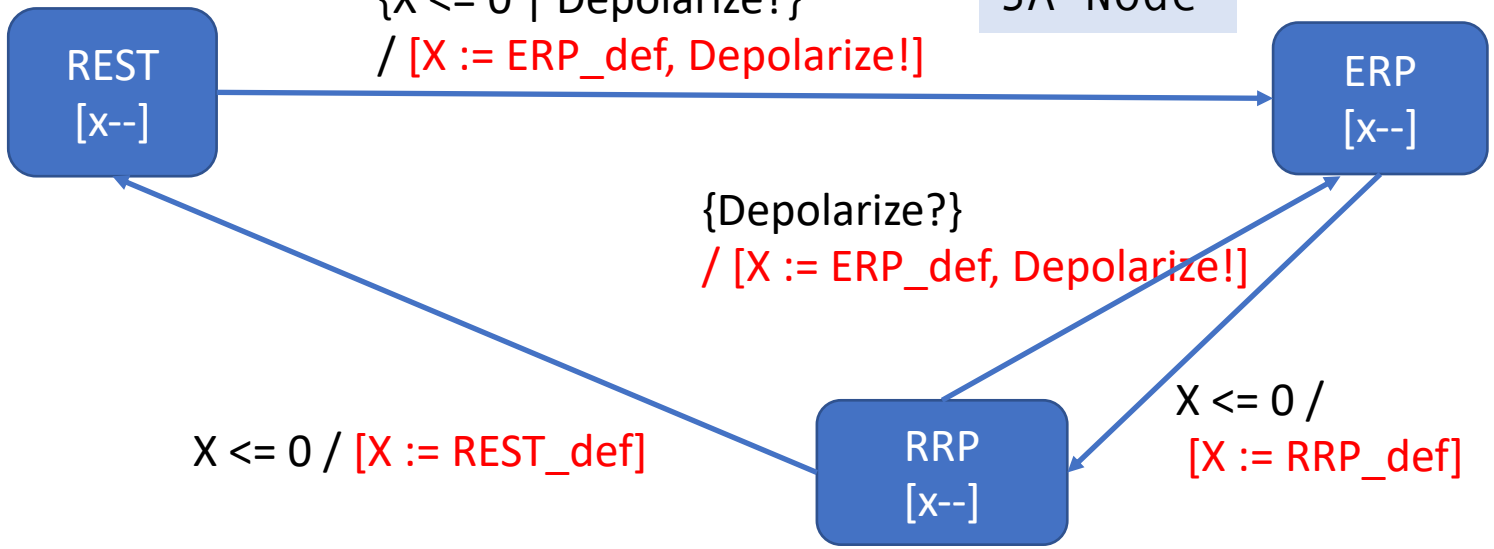


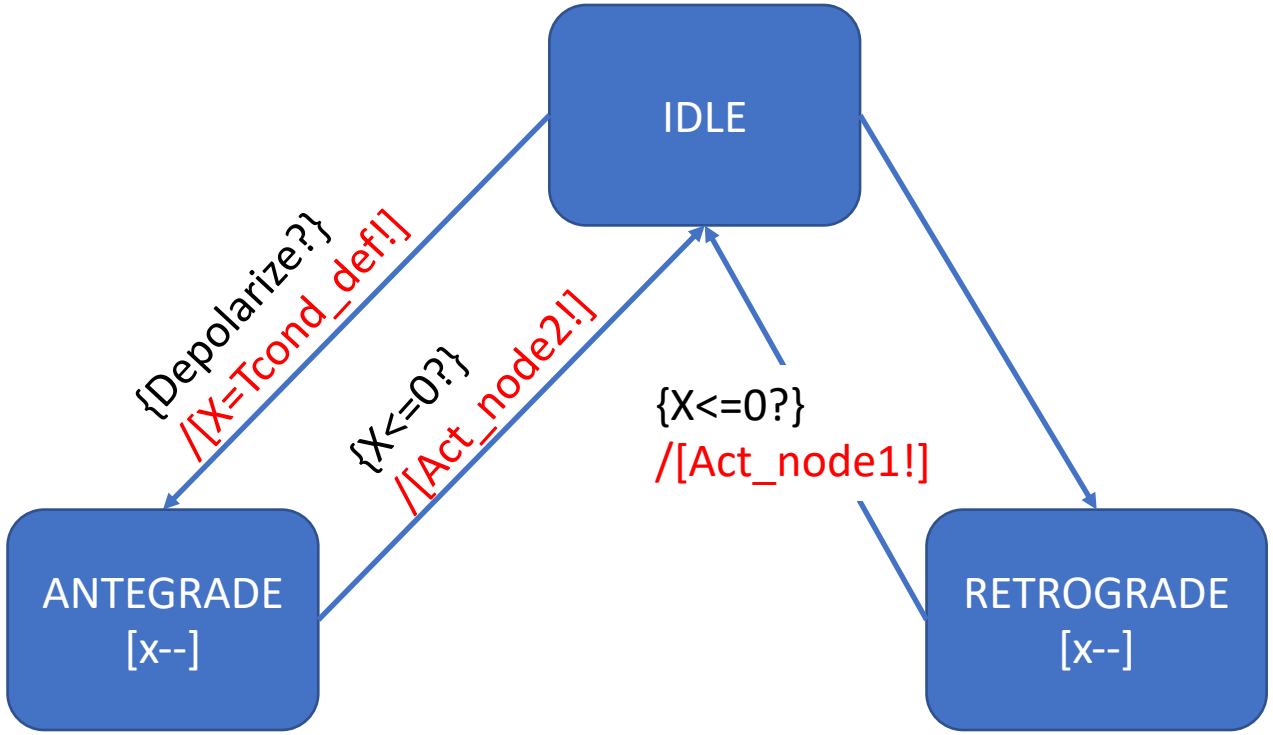
Right ventricle

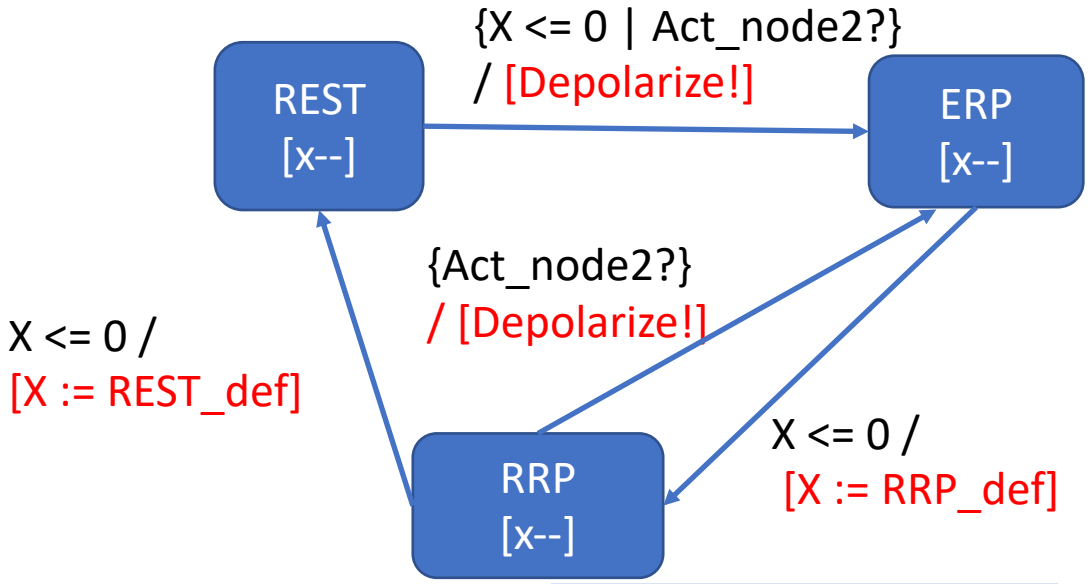
Retrograde Conduction



SA Node

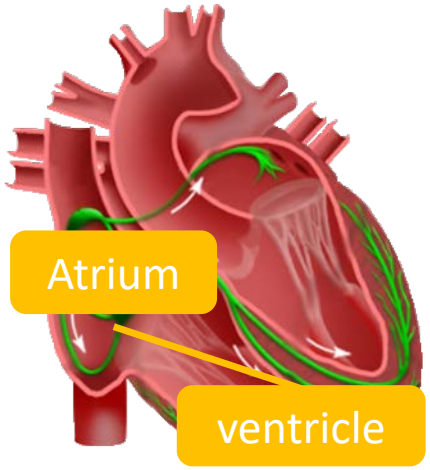




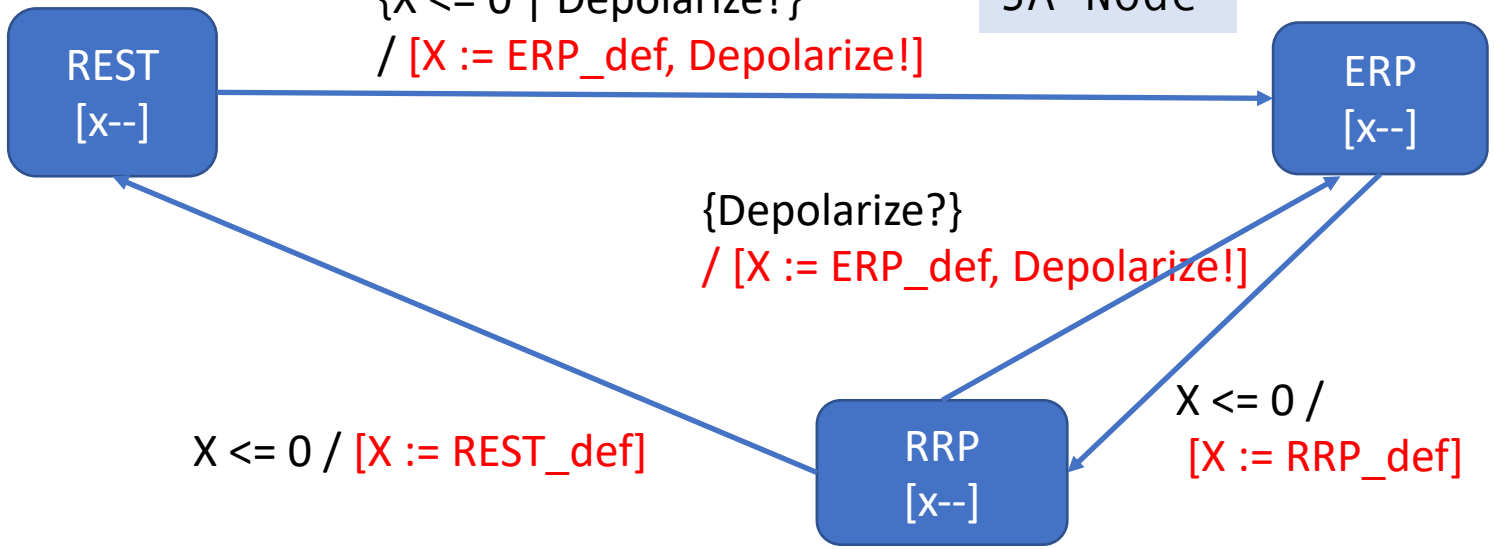


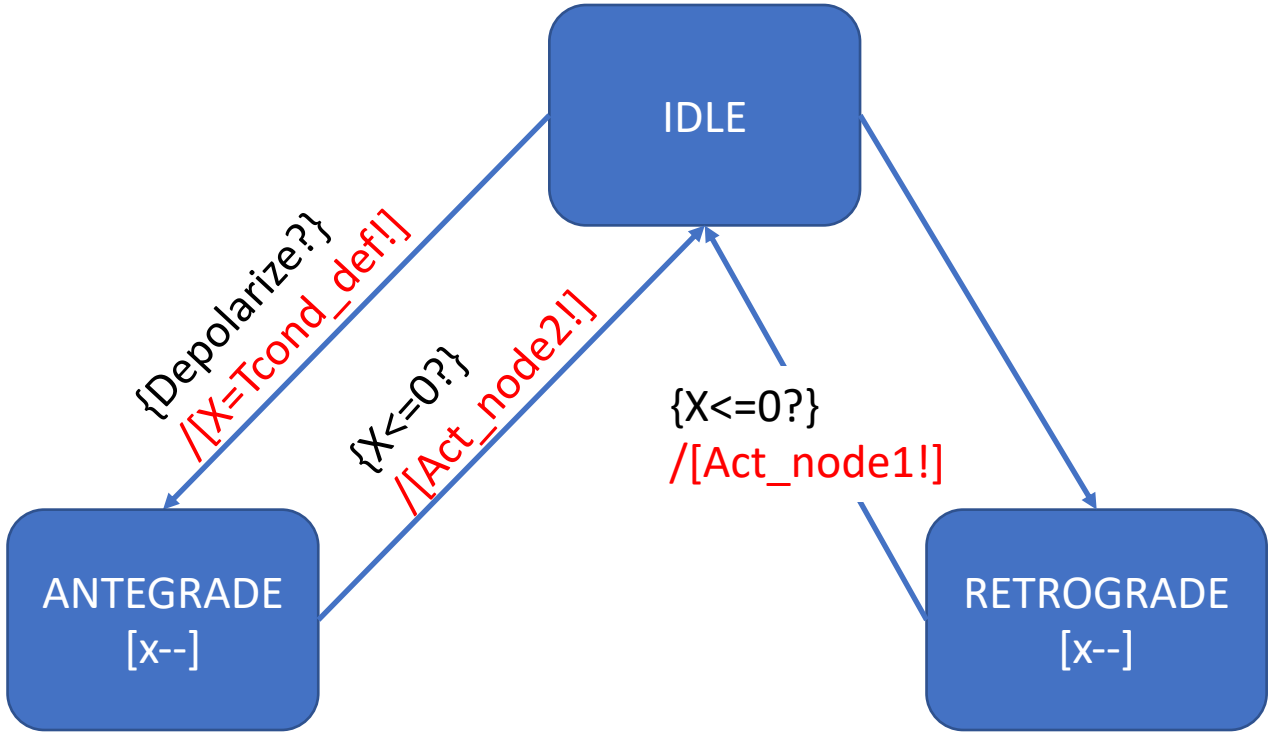
Right ventricle

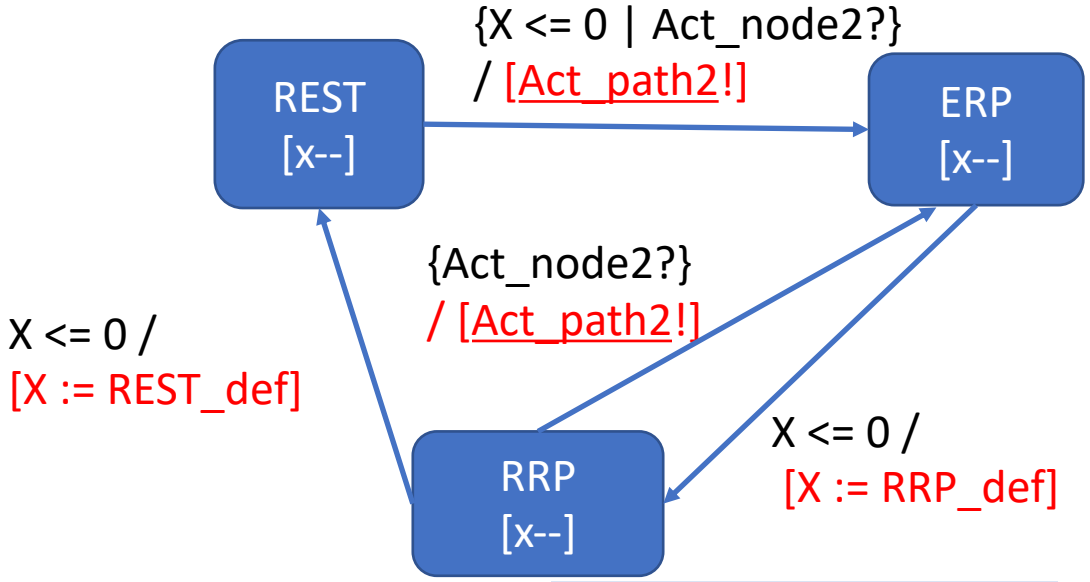
Retrograde Conduction



SA Node

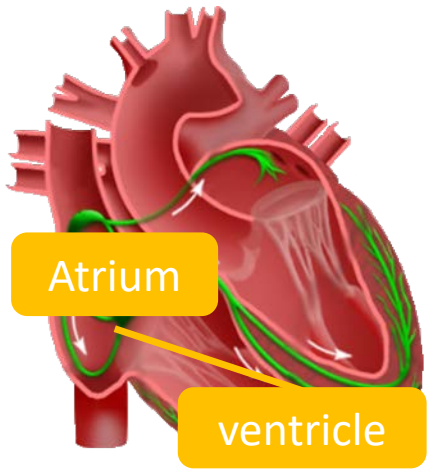




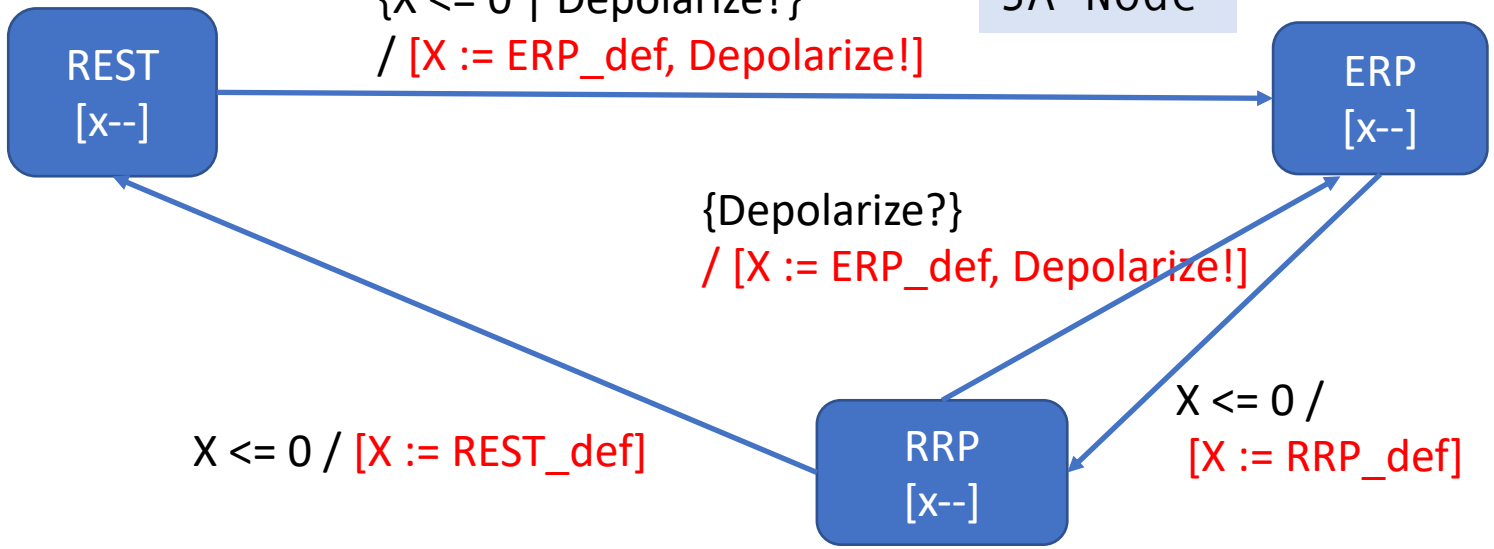


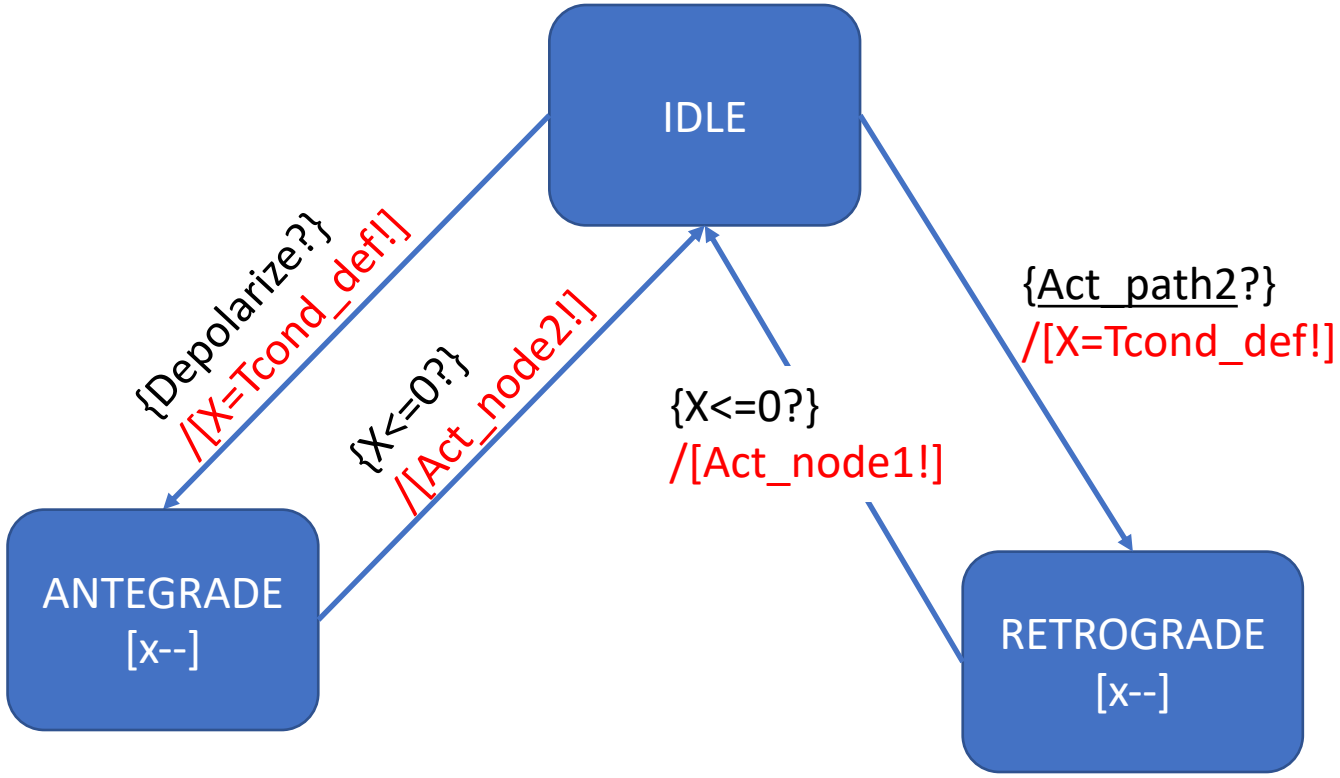
Right ventricle

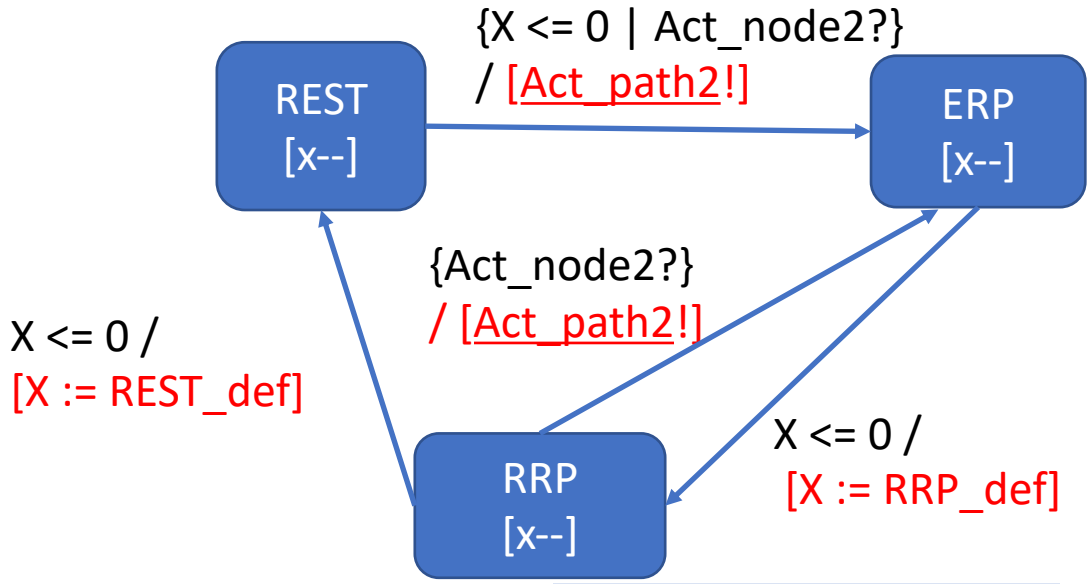
Retrograde Conduction



SA Node

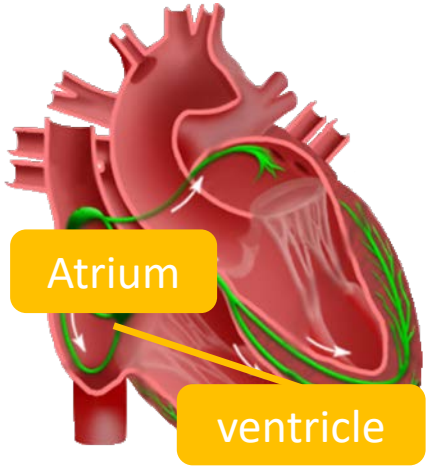




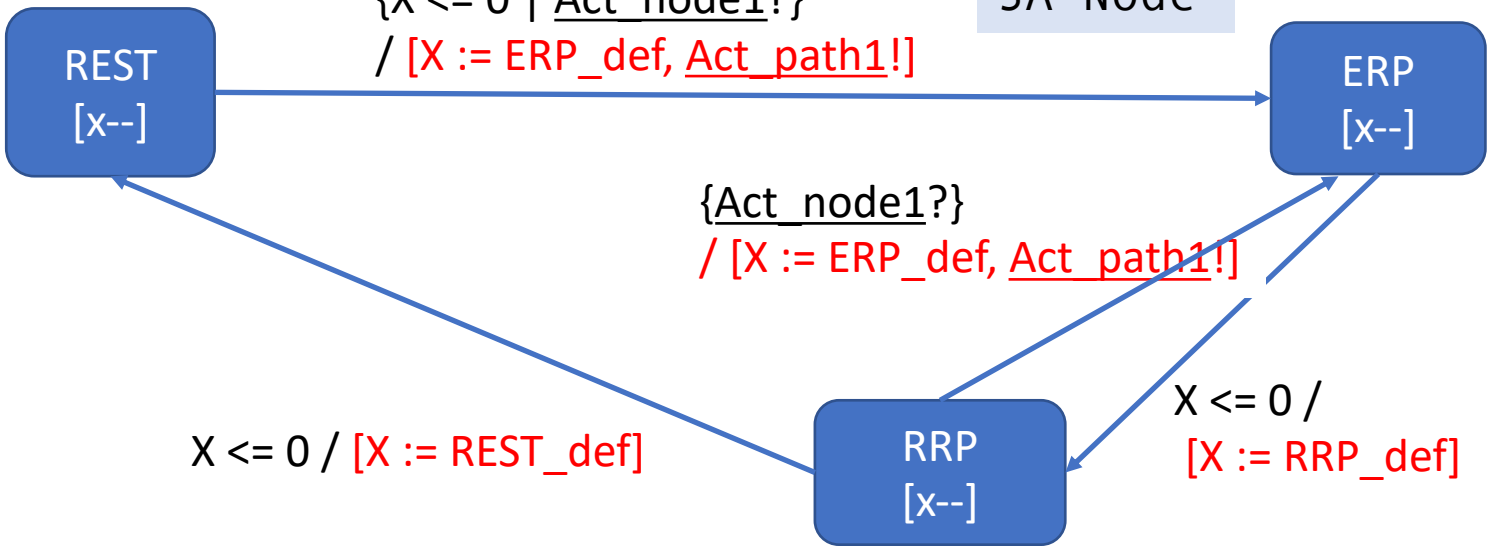


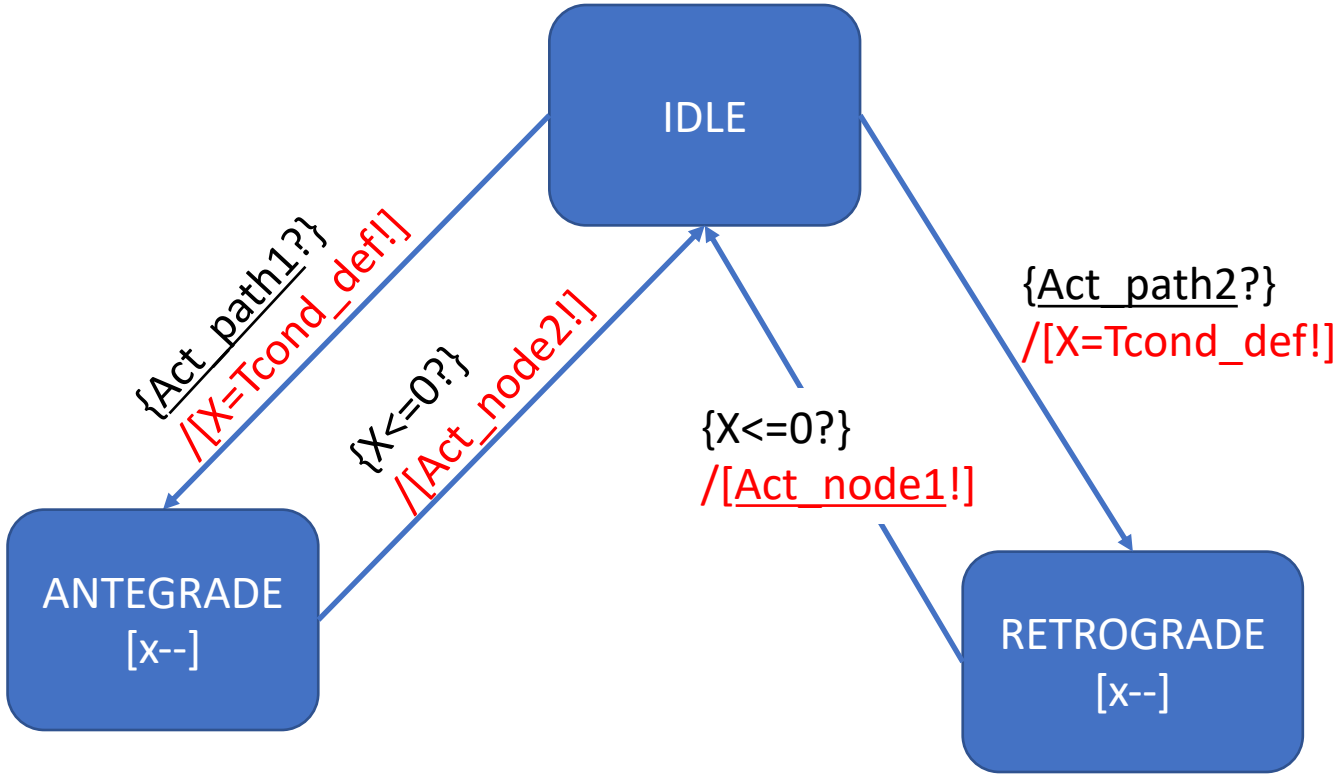
Right ventricle

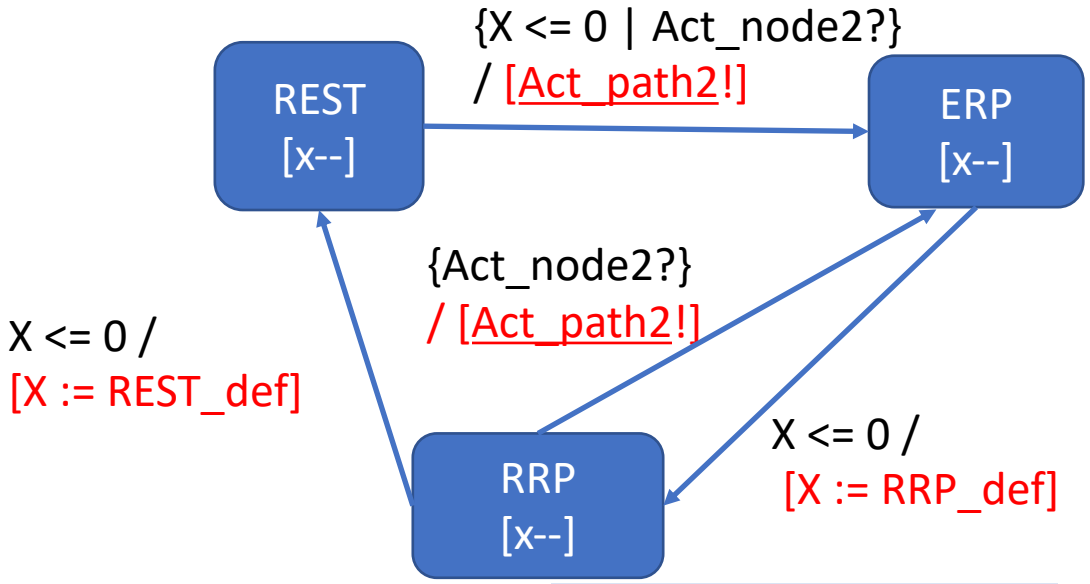
Uniform naming



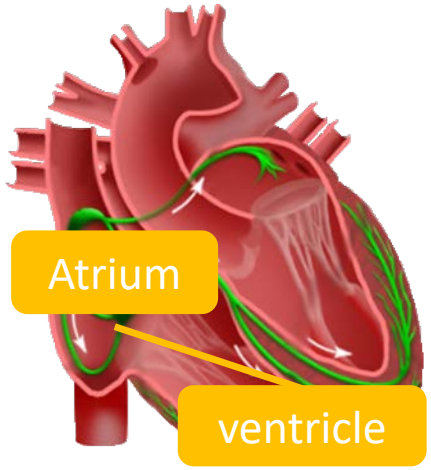
SA Node







Right ventricle



SA Node

REST
[x--]

{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]

ERP
[x--]

{Act_node1?}
/ [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

IDLE

{Act_path1?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node2!]

{Act_path2?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node1!]

ANTEGRADE
[x--]

RETROGRADE
[x--]

REST
[x--]

{X <= 0 | Act_node2?}
/ [Act_path2!]

ERP
[x--]

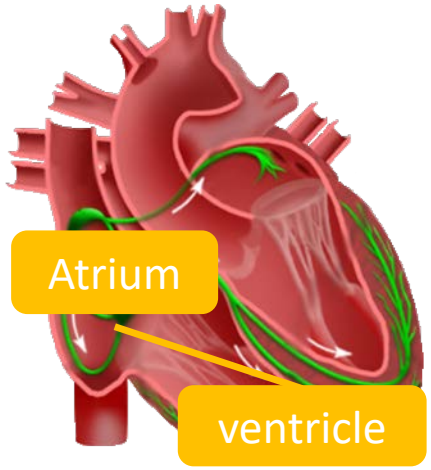
{Act_node2?}
/ [Act_path2!]

X <= 0 /
[X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

Right ventricle



SA Node

REST
[x--]

{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]

ERP
[x--]

{Act_node1?}
/ [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

IDLE

{Act_path1?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node2!]

{Act_path2?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node1!]

ANTEGRADE
[x--]

RETROGRADE
[x--]

REST
[x--]

{X <= 0 | Act_node2?}
/ [Act_path2!]

ERP
[x--]

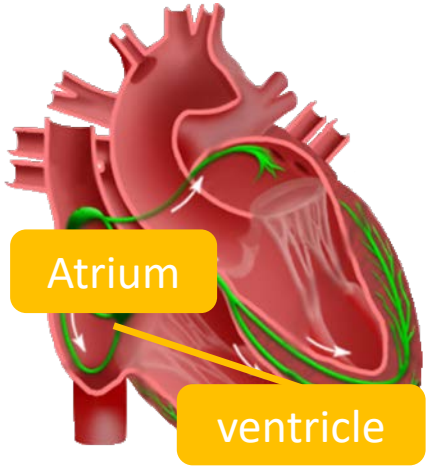
{Act_node2?}
/ [Act_path2!]

X <= 0 /
[X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

Right ventricle



SA Node

REST
[x--]

{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]

ERP
[x--]

{Act_node1?}
/ [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

IDLE

{Act_path1?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node2!]

{Act_path2?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node1!]

ANTEGRADE
[x--]

RETROGRADE
[x--]

REST
[x--]

{X <= 0 | Act_node2?}
/ [Act_path2!]

ERP
[x--]

{Act_node2?}
/ [Act_path2!]

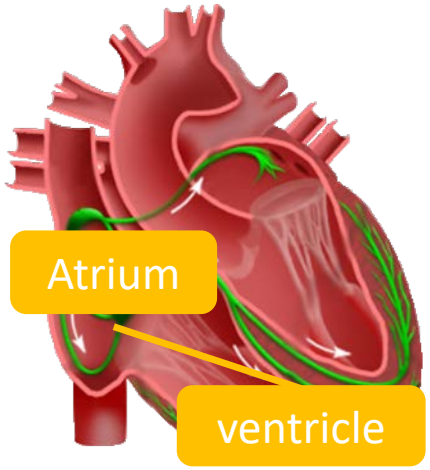
X <= 0 /
[X := REST_def]

RRP
[x--]

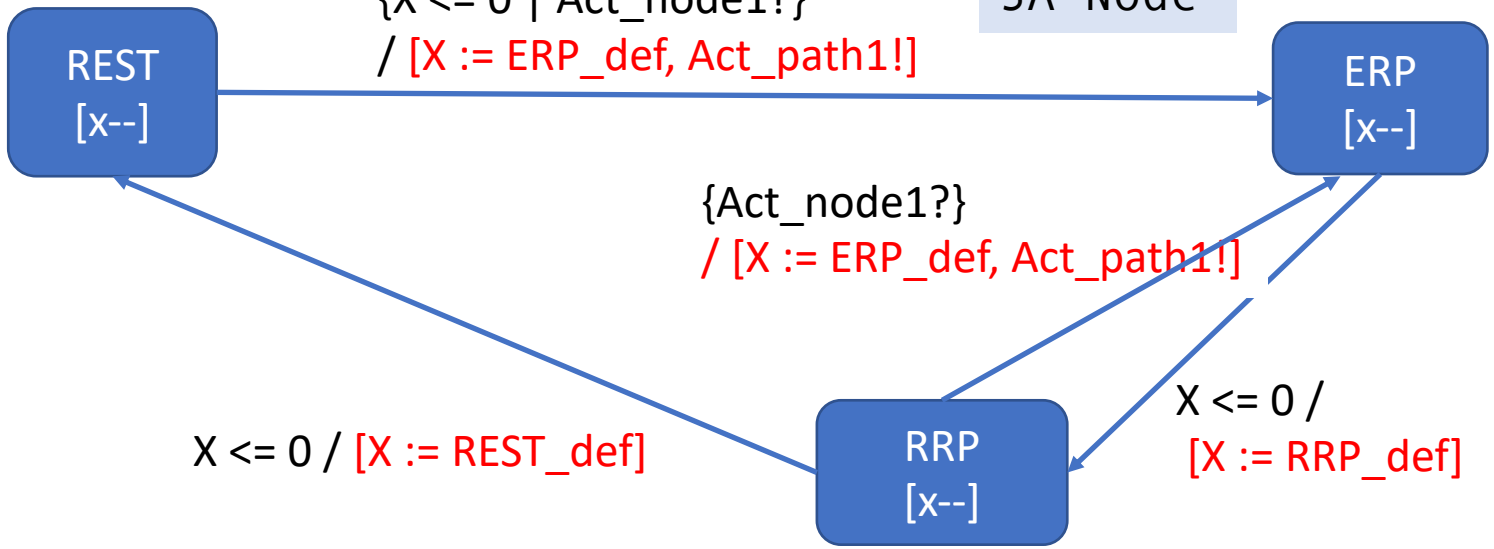
X <= 0 /
[X := RRP_def]

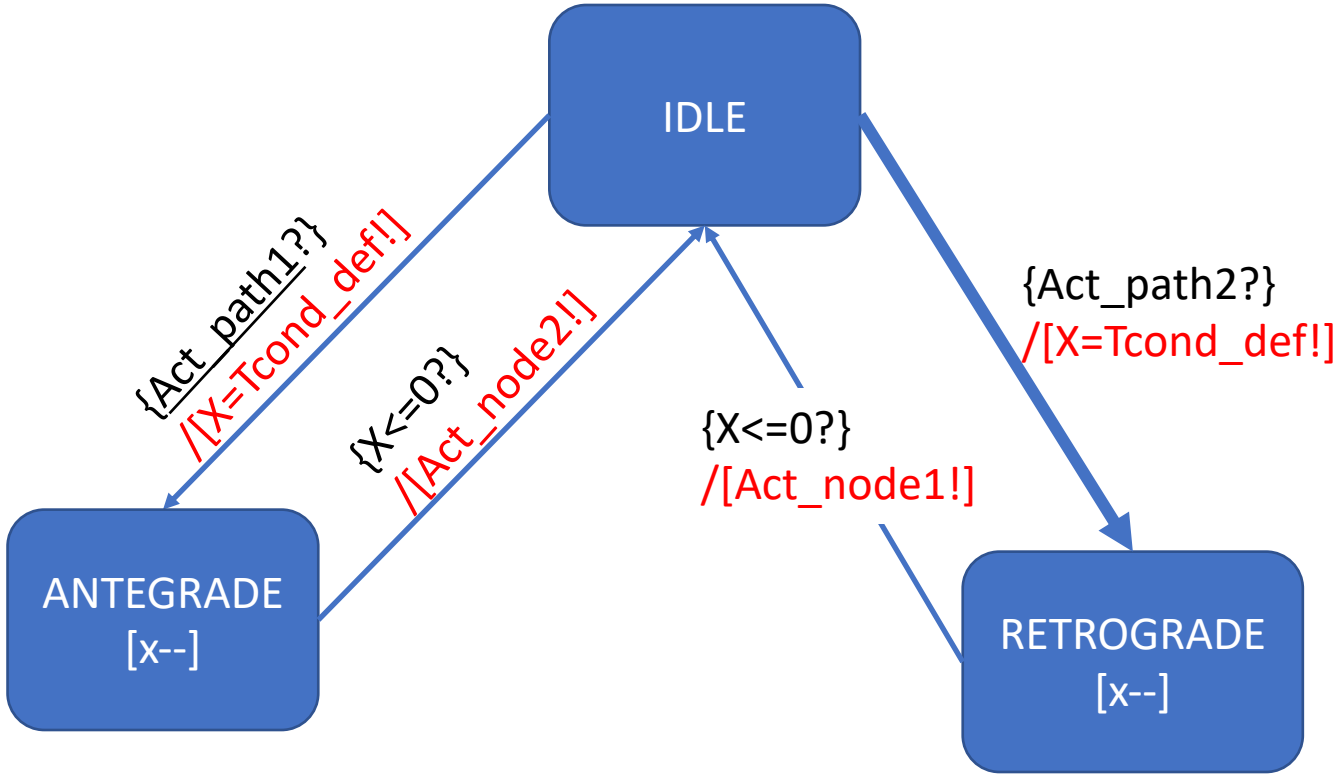
Right ventricle

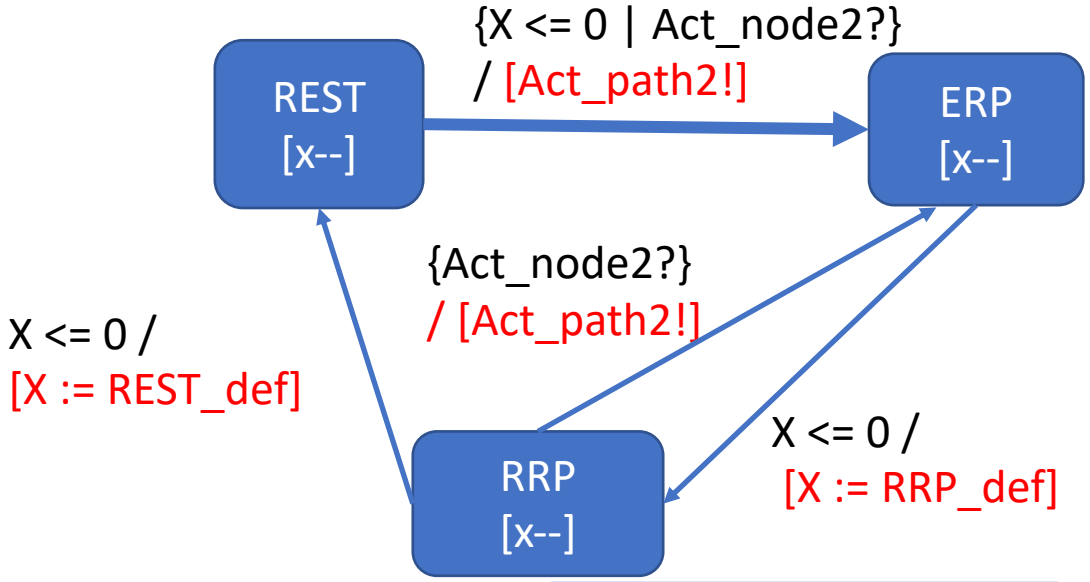
Infinite loop?



SA Node

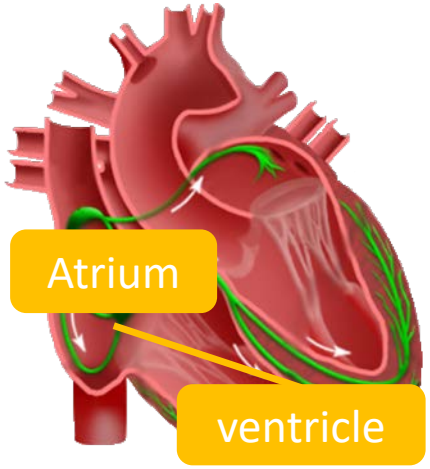






Right ventricle

Infinite loop?



SA Node

REST
[x--]

{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]

ERP
[x--]

{Act_node1?}
/ [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

IDLE

{Act_path1?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node2!]

{Act_path2?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node1!]

ANTEGRADE
[x--]

RETROGRADE
[x--]

REST
[x--]

{X <= 0 | Act_node2?}
/ [Act_path2!]

ERP
[x--]

{Act_node2?}
/ [Act_path2!]

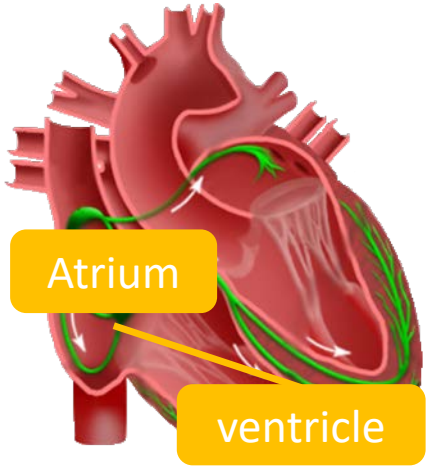
X <= 0 /
[X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

Right ventricle

Infinite loop?



SA Node

REST
[x--]

{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]

ERP
[x--]

{Act_node1?}
/ [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

IDLE

{Act_path1?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node2!]

{Act_path2?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node1!]

ANTEGRADE
[x--]

RETROGRADE
[x--]

REST
[x--]

{X <= 0 | Act_node2?}
/ [Act_path2!]

ERP
[x--]

{Act_node2?}
/ [Act_path2!]

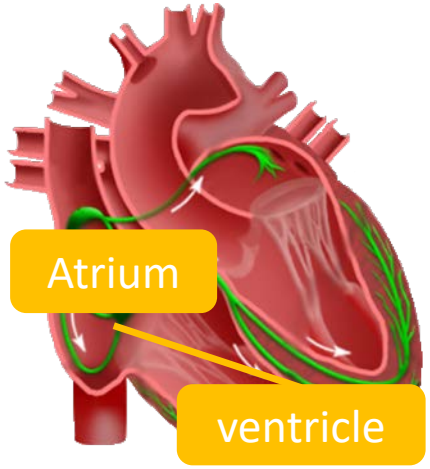
X <= 0 /
[X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

Right ventricle

Infinite loop!



SA Node

REST
[x--]

{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]

ERP
[x--]

{Act_node1?}
/ [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

IDLE

{Act_path1?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node2!]

{Act_path2?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node1!]

ANTEGRADE
[x--]

RETROGRADE
[x--]

REST
[x--]

{X <= 0 | Act_node2?}
/ [Act_path2!]

ERP
[x--]

{Act_node2?}
/ [Act_path2!]

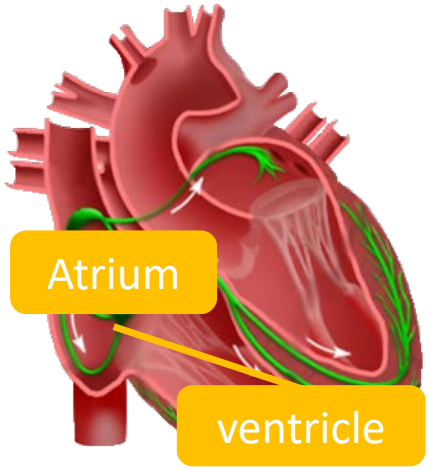
X <= 0 /
[X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

Right ventricle

Infinite loop!



SA Node

REST
[x--]

{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]

2

ERP
[x--]

{Act_node1?}
/ [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

IDLE

2

{Act_path1?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node2!]

3

ANTEGRADE
[x--]

{X<=0?}
/[Act_node1!]

2

{Act_path2?}
/[X=Tcond_def!]

1

RETROGRADE
[x--]

3

{X <= 0 | Act_node2?}
/ [Act_path2!]

REST
[x--]

ERP
[x--]

1

{Act_ ?}
/ [Act_path2!]

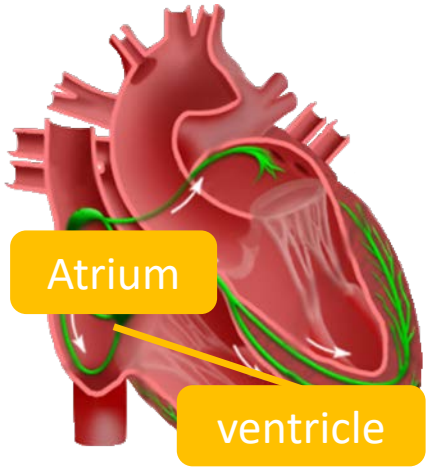
X <= 0 /
[X := REST_def]

RRP
[x--]

X <= 0 /
[X := RRP_def]

Right ventricle

Breaking the Infinite loop



SA Node

REST [x--]

{X <= 0 | Act_node1?} / [X := ERP_def, Act_path1!]

2

ERP [x--]

{Act_node1?} / [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP [x--]

X <= 0 / [X := RRP_def]

IDLE

2

{Act_path1?} /[X=Tcond_def!]

{X<=0?} /[Act_node2!]

{X<=0?} /[Act_node1!]

2

{Act_path2?} /[X=Tcond_def!]

1

ANTEGRADE [x--]

RETROGRADE [x--]

Right ventricle

REST [x--]

{X <= 0 | Act_node2?} / [Act_path2!]

ERP [x--]

1

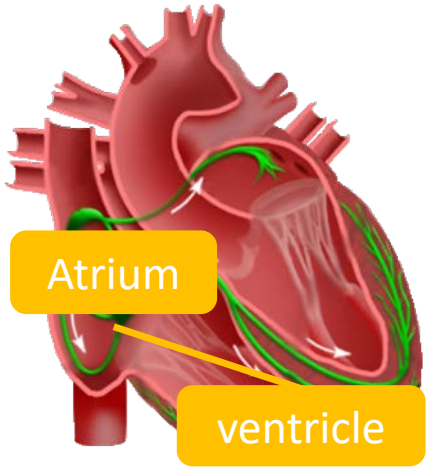
{Act_ ?} / [Act_path2!]

X <= 0 / [X := REST_def]

RRP [x--]

X <= 0 / [X := RRP_def]

Breaking the Infinite loop



SA Node

REST [x--]

{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]

2

ERP [x--]

{Act_node1?}
/ [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP [x--]

X <= 0 /
[X := RRP_def]

IDLE

2

{Act_path1?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node2!]

{Act_path2?}
/[X=Tcond_def!]

1

{X<=0?}
/[Act_node1!]

2

ANTEGRADE [x--]

RETROGRADE [x--]

REST [x--]

{X <= 0 | Act_node2?}
/ [Act_path2!]

ERP [x--]

1

{Act_ ?}
/ [Act_path2!]

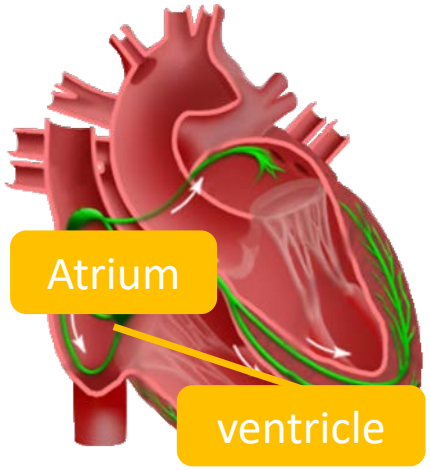
X <= 0 /
[X := REST_def]

RRP [x--]

X <= 0 /
[X := RRP_def]

Right ventricle

Breaking the Infinite loop



SA Node

REST [x--]

{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]

ERP [x--]

{Act_node1?}
/ [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP [x--]

X <= 0 /
[X := RRP_def]

IDLE

{Act_path1?}
/[X=Tcond_def!]

{X<=0?}
/[Act_node2!]

{Act_path2?}
/[X=Tcond_def!]

{x<=0?}

ANTEGRADE [x--]

RETROGRADE [x--]

WAIT [x--]

{X<=0?}
/[Act_node1!,
X=1]

REST [x--]

{X <= 0 | Act_node2?}
/ [Act_path2!]

ERP [x--]

{Act_node2?}
/ [Act_path2!]

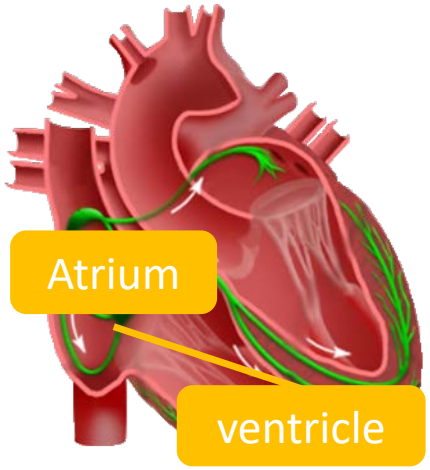
X <= 0 /
[X := REST_def]

RRP [x--]

X <= 0 /
[X := RRP_def]

Right ventricle

Breaking the Infinite loop



SA Node

REST [x--]

{X <= 0 | Act_node1?} / [X := ERP_def, Act_path1!]

2

ERP [x--]

{Act_node1?} / [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP [x--]

X <= 0 / [X := RRP_def]

IDLE

{Act_path1?} /[X=Tcond_def!]

{X<=0?} /[Act_node2!]

{x<=0?}

{Act_path2?} /[X=Tcond_def!]

1

ANTEGRADE [x--]

RETROGRADE [x--]

WAIT [x--]

{X<=0?} /[Act_node1, X=1!]

2

X <= 0 / [X := REST_def]

REST [x--]

{X <= 0 | Act_node2?} / [Act_path2!]

ERP [x--]

1

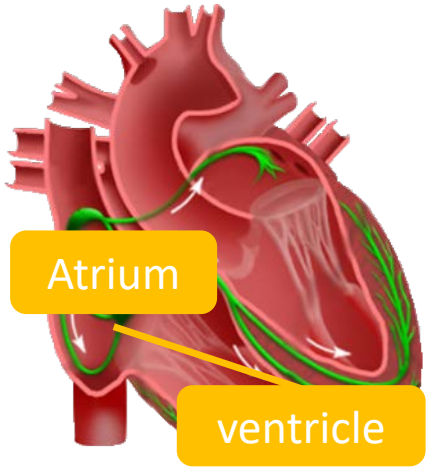
{Act_ ?} / [Act_path2!]

RRP [x--]

X <= 0 / [X := RRP_def]

Right ventricle

Breaking the Infinite loop



SA Node

REST [x--]

{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]

ERP [x--]

{Act_node1?}
/ [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP [x--]

X <= 0 /
[X := RRP_def]

IDLE

{Act_path1?}
/[X=Tcond_def!]

{Act_path2?}
/[X=Tcond_def!]

{x<=0?}

ANTEGRADE [x--]

RETROGRADE [x--]

{X<=0?}
/[Act_node2,
X=1!]

WAIT [x--]

{X<=0?}
/[Act_node1,
X=1!]

REST [x--]

{X <= 0 | Act_node2?}
/ [Act_path2!]

ERP [x--]

{Act_node2?}
/ [Act_path2!]

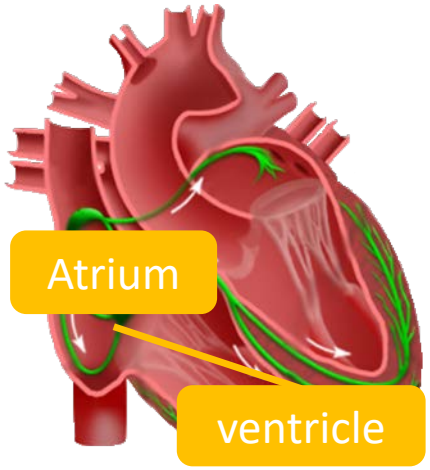
X <= 0 /
[X := REST_def]

RRP [x--]

X <= 0 /
[X := RRP_def]

Right ventricle

Breaking the Infinite loop



SA Node

REST [x--]

{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]

1

ERP [x--]

{Act_node1?}
/ [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP [x--]

X <= 0 /
[X := RRP_def]

IDLE

1

{Act_path1?}
/[X=Tcond_def!]

{Act_path2?}
/[X=Tcond_def!]

{x<=0?}

3

ANTEGRADE [x--]

2

RETROGRADE [x--]

{X<=0?}
/[Act_node2,
X=1!]

WAIT [x--]

{X<=0?}
/[Act_node1,
X=1!]

REST [x--]

{X <= 0 | Act_node2?}
/ [Act_path2!]

2

ERP [x--]

{Act_node2?}
/ [Act_path2!]

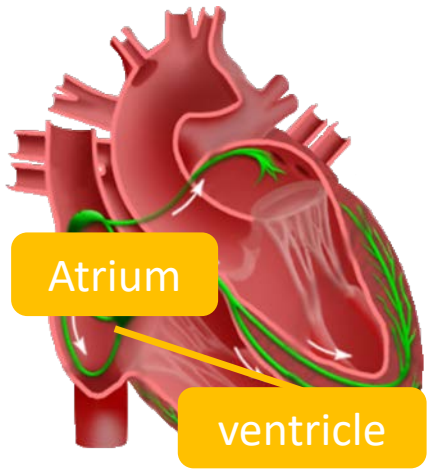
X <= 0 /
[X := REST_def]

RRP [x--]

X <= 0 /
[X := RRP_def]

Right ventricle

Simultaneous propagations



SA Node

REST [x--]

{X <= 0 | Act_node1?} / [X := ERP_def, Act_path1!]

ERP [x--]

{Act_node1?} / [X := ERP_def, Act_path1!]

X <= 0 / [X := REST_def]

RRP [x--]

X <= 0 / [X := RRP_def]

IDLE

{Act_path1?} /[X=Tcond_def!]

{Act_path2?} /[X=Tcond_def!]

{x<=0?}

ANTEGRADE [x--]

RETROGRADE [x--]

WAIT [x--]

{X<=0?} /[Act_node2,X=1!]

{X<=0?} /[Act_node1,X=1!]

REST [x--]

{X <= 0 | Act_node2?} / [Act_path2!]

ERP [x--]

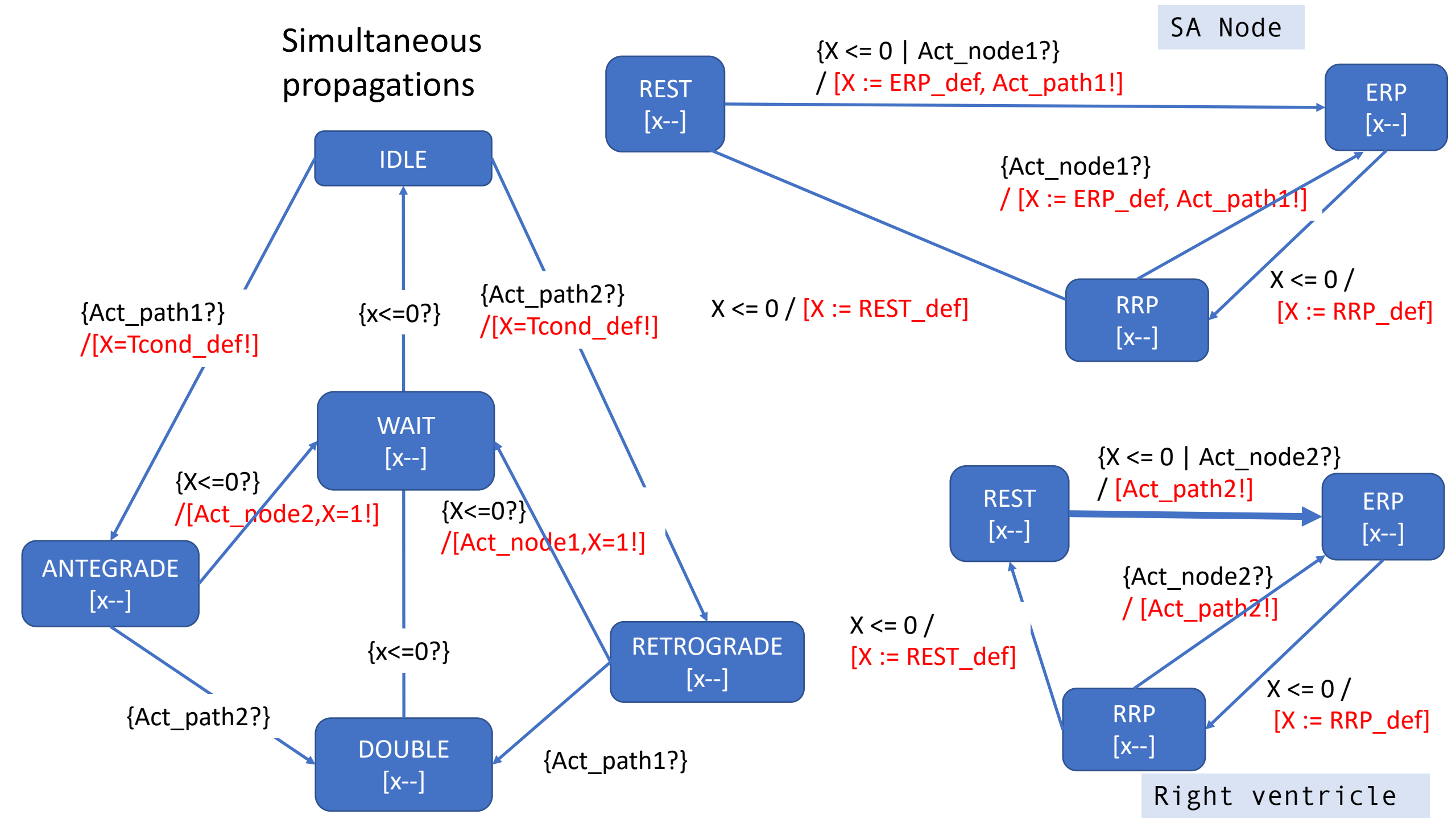
{Act_node2?} / [Act_path2!]

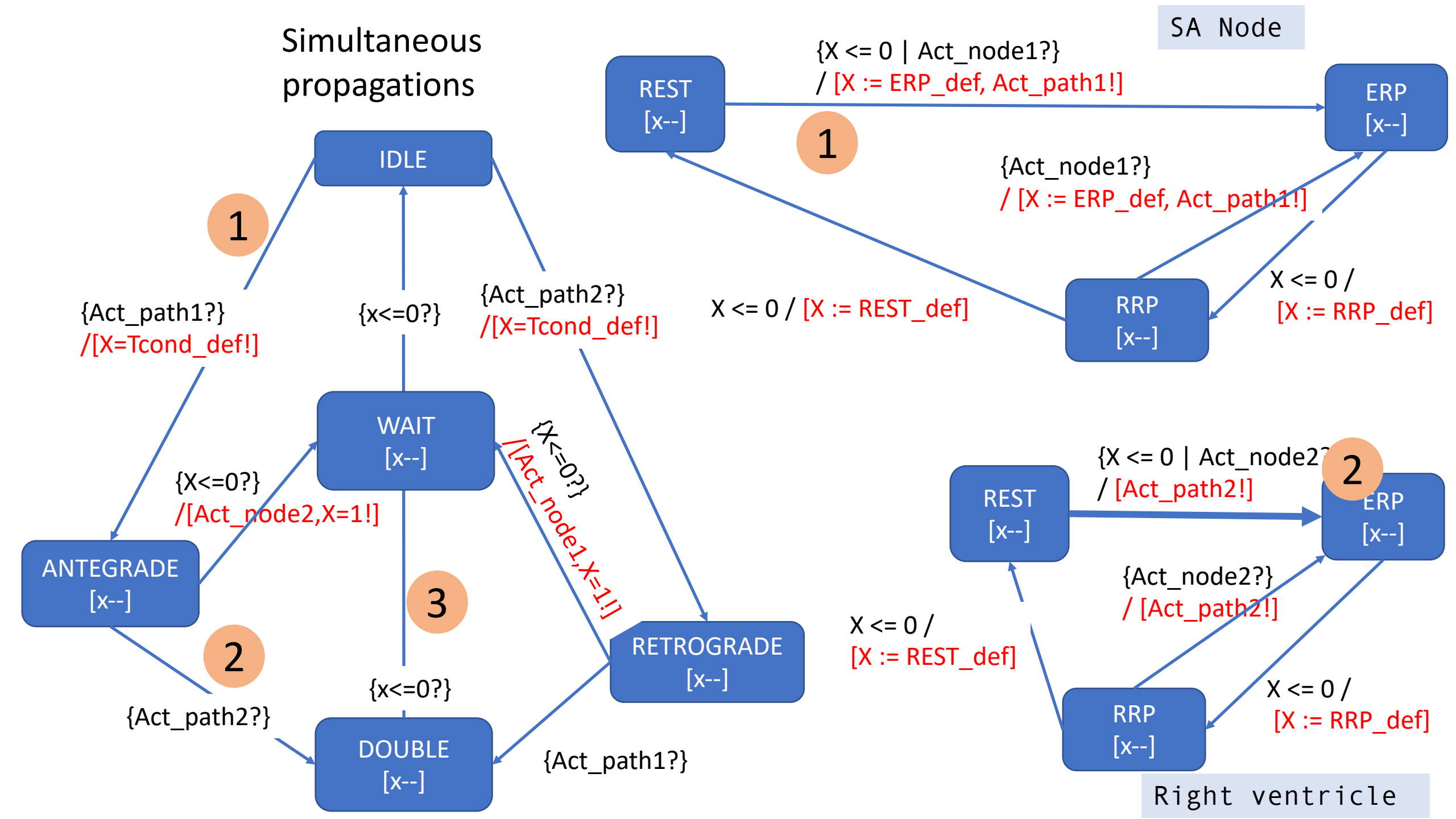
X <= 0 / [X := REST_def]

RRP [x--]

X <= 0 / [X := RRP_def]

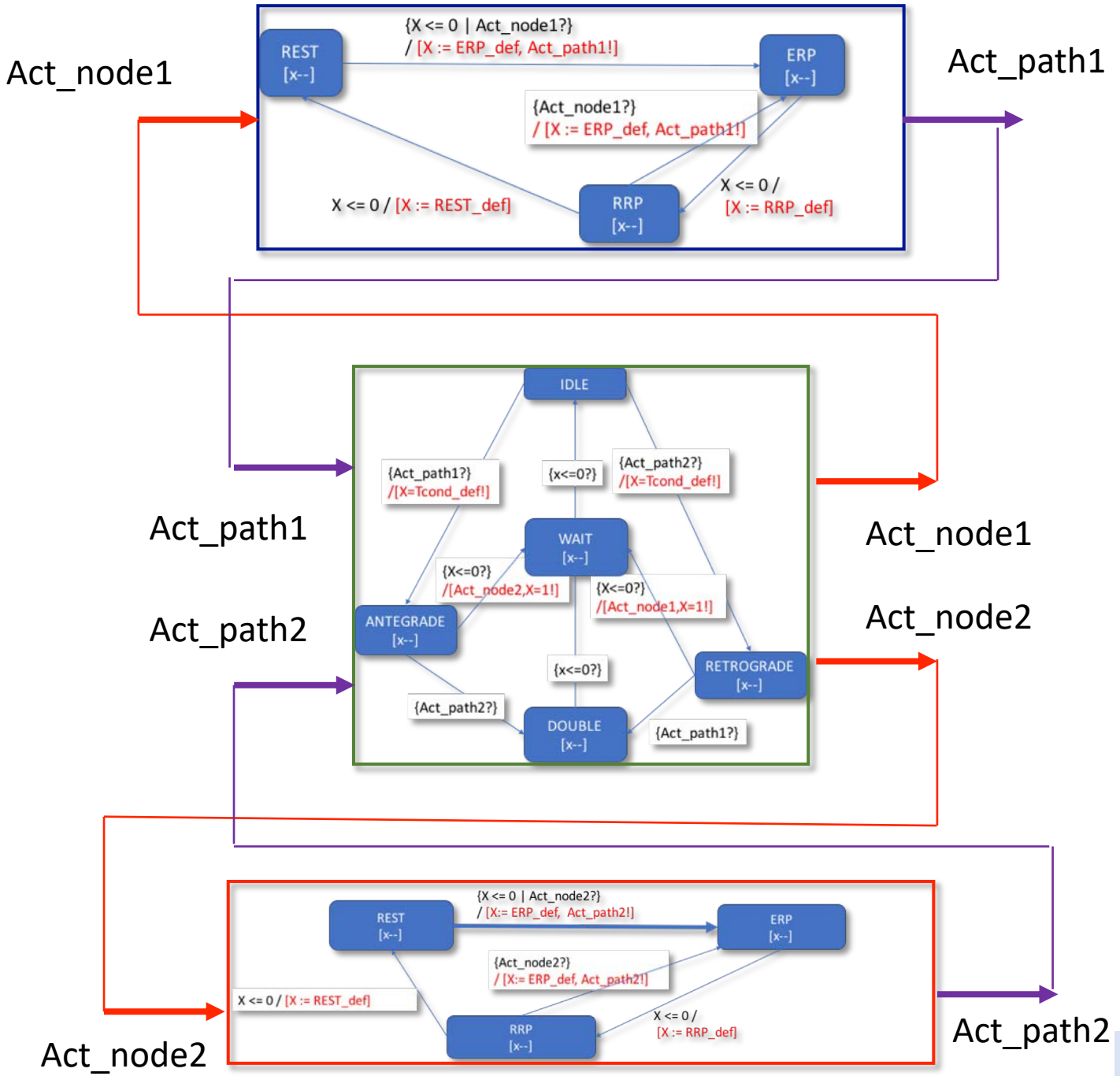
Right ventricle


Simultaneous propagations
IDLE
WAIT
[x--]
ANTEGRADE
[x--]
RETROGRADE
[x--]
DOUBLE
[x--]
{Act_path1?}
/[X=Tcond_def!]
{x<=0?}
{Act_path2?}
/[X=Tcond_def!]
{X<=0?}
/[Act_node2,X=1!]
{X<=0?}
/[Act_node1,X=1!]
{x<=0?}
{Act_path2?}
{Act_path1?}
SA Node
REST
[x--]
ERP
[x--]
RRP
[x--]
{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]
{Act_node1?}
/ [X := ERP_def, Act_path1!]
X <= 0 / [X := REST_def]
X <= 0 /
[X := RRP_def]
REST
[x--]
ERP
[x--]
RRP
[x--]
{X <= 0 | Act_node2?}
/ [Act_path2!]
{Act_node2?}
/ [Act_path2!]
X <= 0 /
[X := REST_def]
X <= 0 /
[X := RRP_def]
Right ventricle


Simultaneous propagations
IDLE
1
{Act_path1?}
/[X=Tcond_def!]
{x<=0?}
{Act_path2?}
/[X=Tcond_def!]
WAIT
[x--]
{X<=0?}
/[Act_node2,X=1!]
ANTEGRADE
[x--]
{X<=0?}
/[Act_node1,X=1!]
3
2
{Act_path2?}
{x<=0?}
DOUBLE
[x--]
RETROGRADE
[x--]
{Act_path1?}
SA Node
REST
[x--]
{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]
1
ERP
[x--]
{Act_node1?}
/ [X := ERP_def, Act_path1!]
X <= 0 / [X := REST_def]
RRP
[x--]
X <= 0 /
[X := RRP_def]
REST
[x--]
{X <= 0 | Act_node2?
/ [Act_path2!]
2
ERP
[x--]
{Act_node2?}
/ [Act_path2!]
X <= 0 /
[X := REST_def]
RRP
[x--]
X <= 0 /
[X := RRP_def]
Right ventricle

Messages

N
P
N



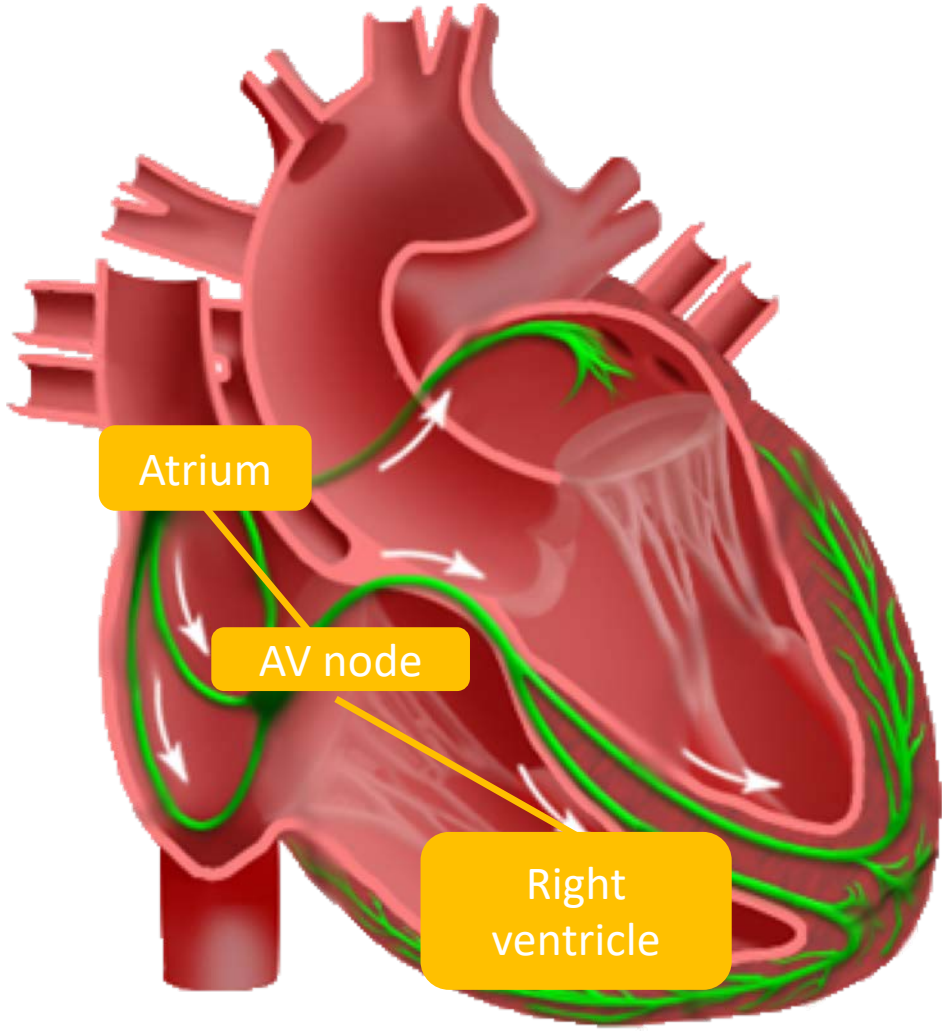
SA Node

Path

Right ventricle

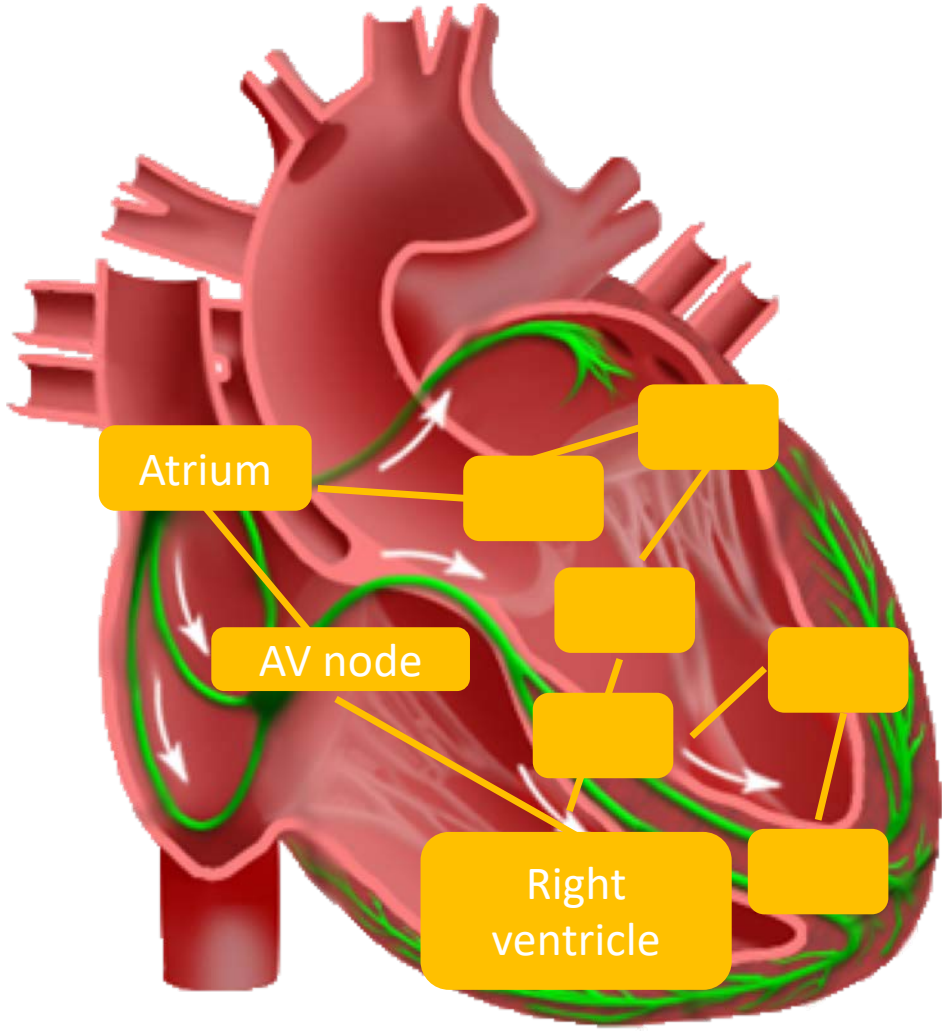
NPNPN

Of course, can add more tissue (more nodes) for a finer granularity model.



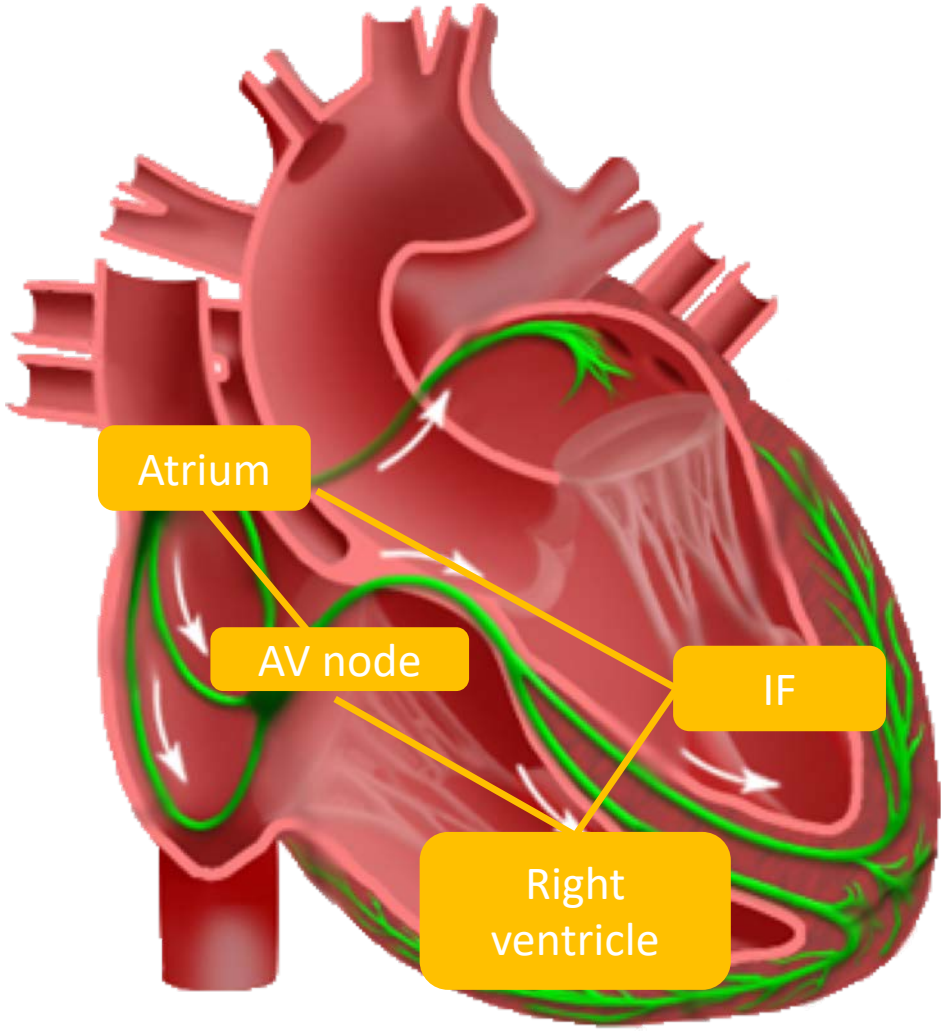
NPNPN

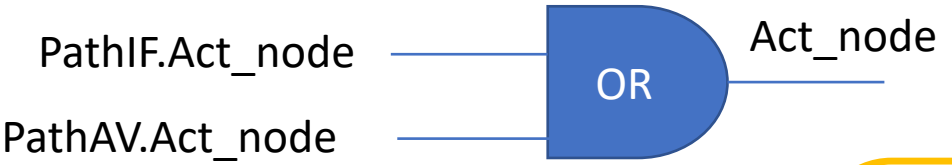
Scalability and refinement: *Model refinement* means to make the model more accurate, more faithful to reality…It is important to create a model that is refinable so when the computational resources are available, it is easy to create a more refined model and get more accurate results.



Irritable focus

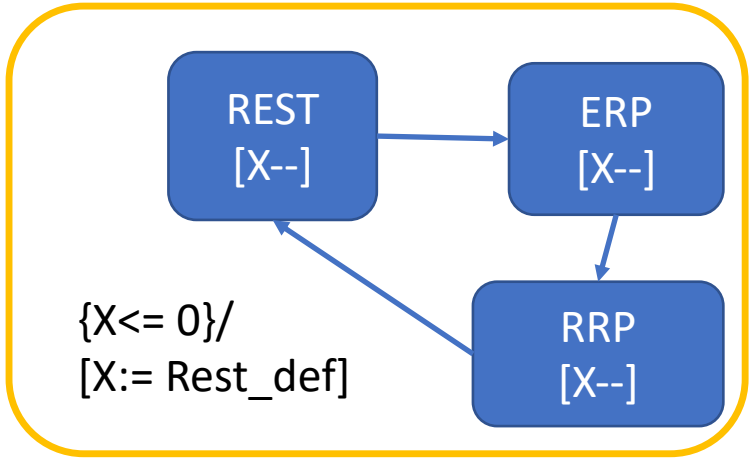
An irritable focus is a piece of tissue that starts depolarizing on its own for a brief (or extended) period of time, sometimes driving the rhythm or otherwise disturbing it.



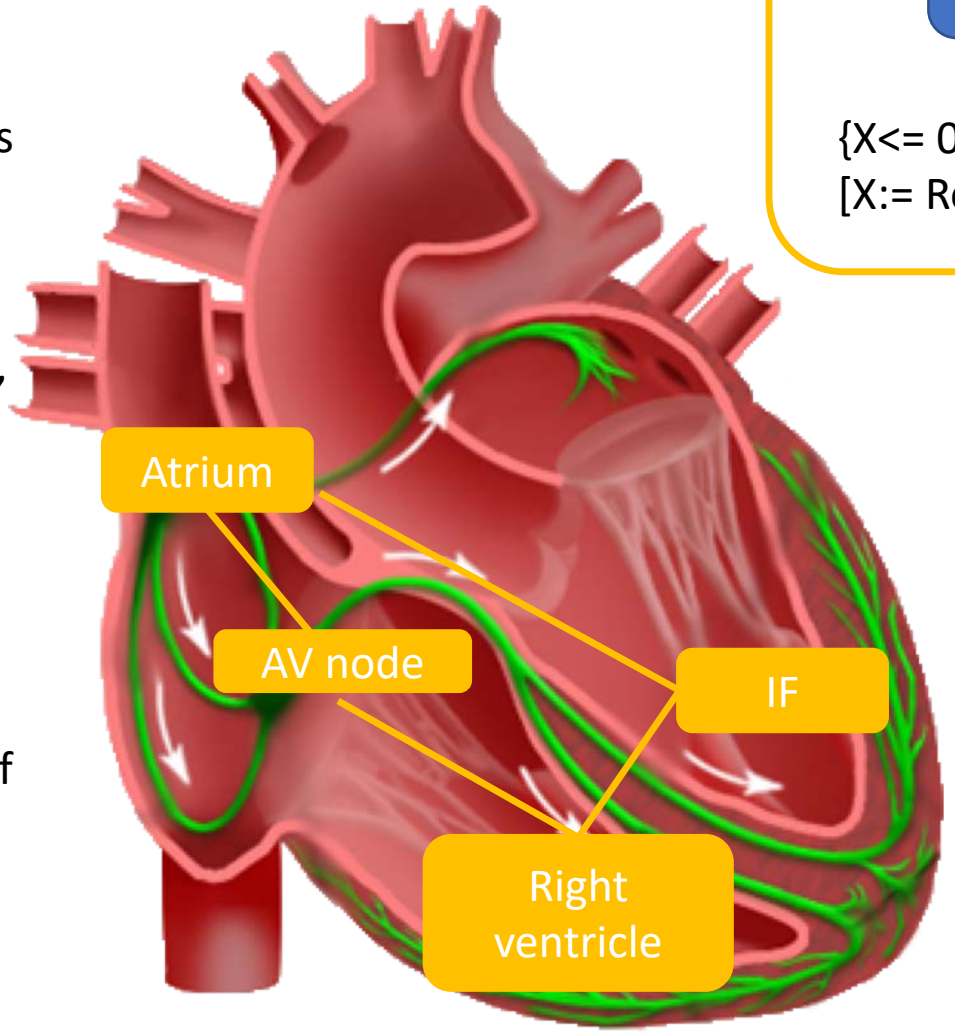


{Rest <= 0 | Act_node}/[Act_path, ERP=ERP_def]

Irritable focus



Recall: an irritable focus is a piece of tissue that starts depolarizing on its own for a brief (or extended) period of time, sometimes driving the rhythm or otherwise disturbing it.



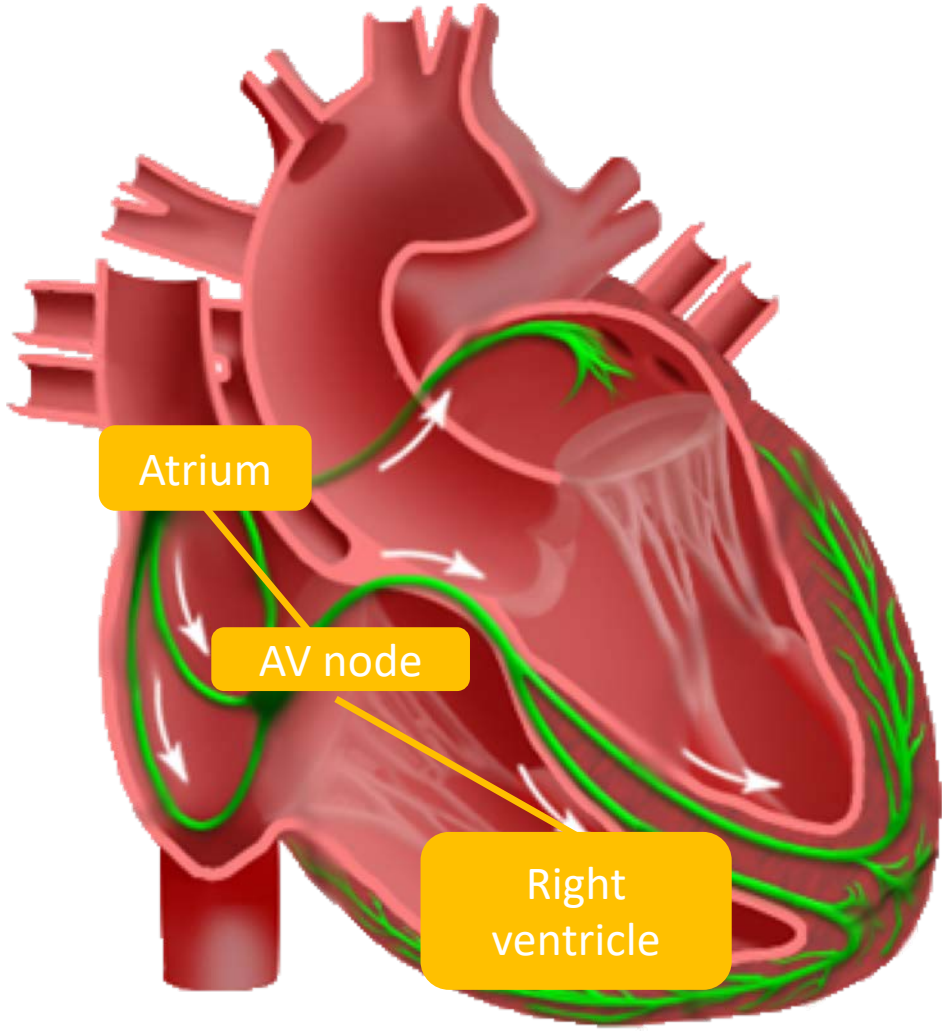
Note also the modeling of retrograde conduction.

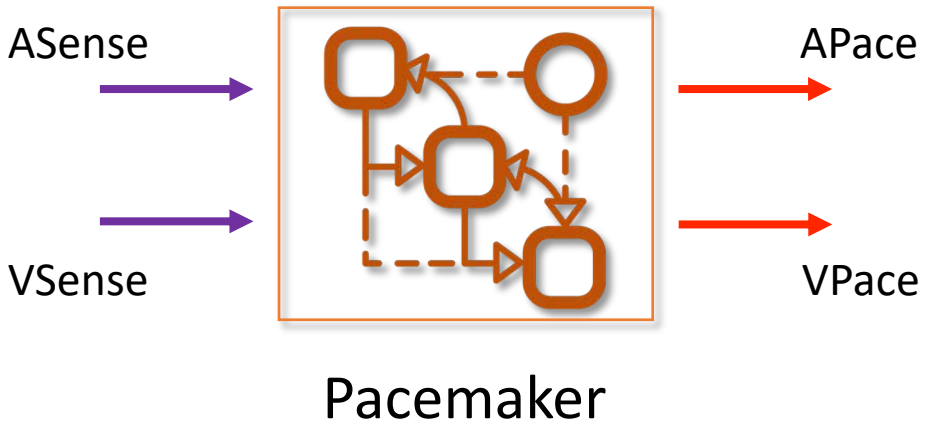
Atrio-Ventricular (AV) Node

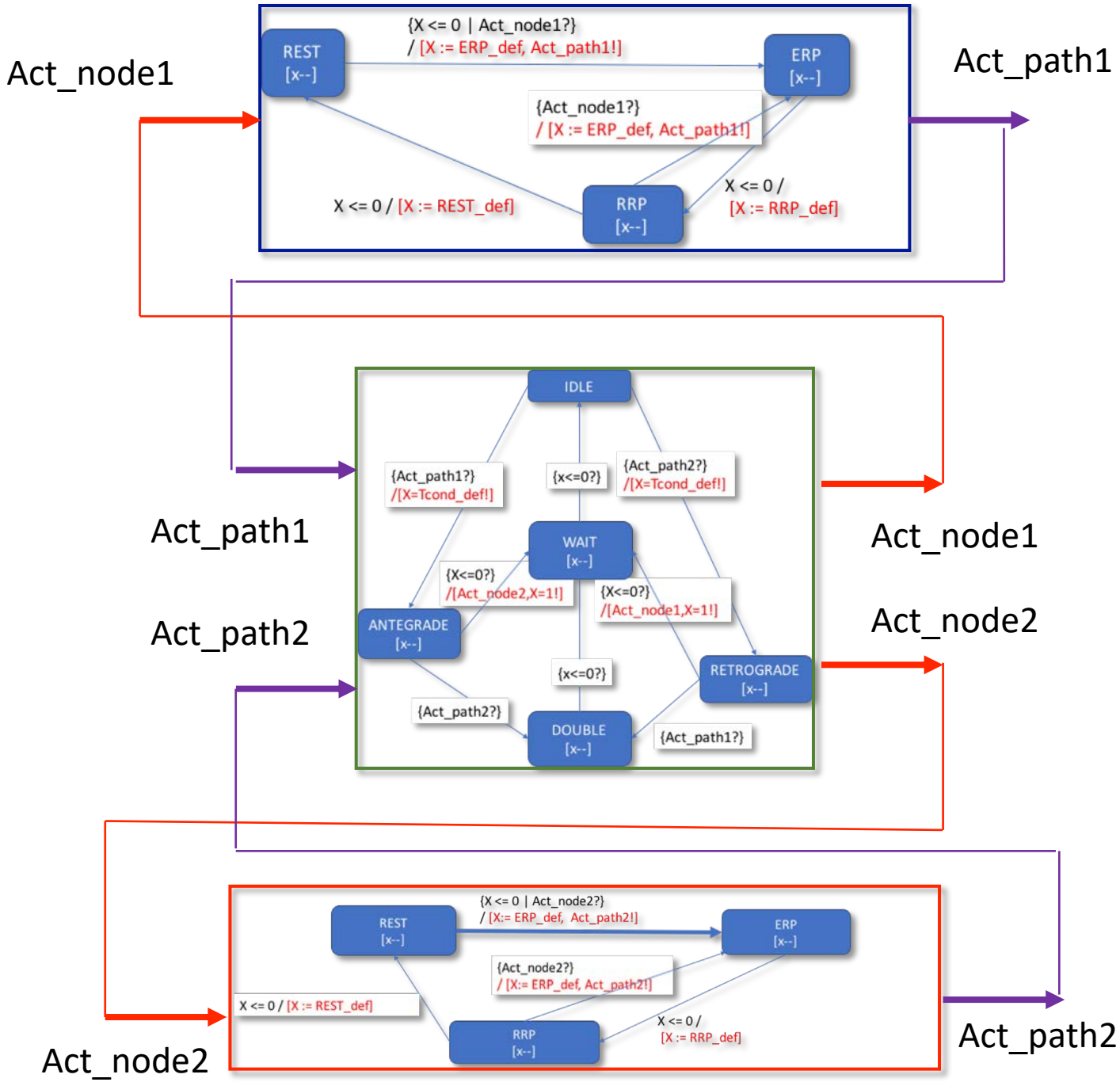
The AV node adds a delay between atria and ventricles. This delay allows the ventricles to fill with blood before they contract to pump that blood out to the body.

It also acts as a rate controller: if atria beat too fast, every few beats will be dropped by the AV node and won't make it to the ventricles. That's because those events fall in the AV's ERP.

The ERP of the AV node *increases* as the excitations come closer together.




ASense
VSense
APace
VPace
Pacemaker

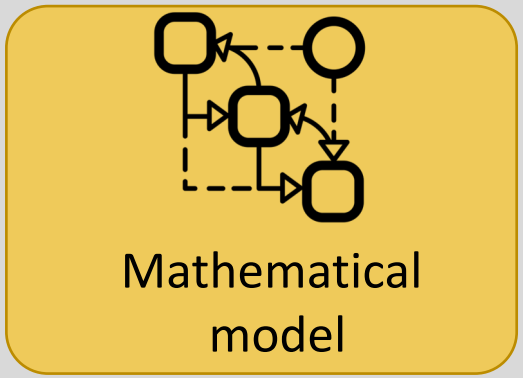

Act_node1
REST
[x--]
{X <= 0 | Act_node1?}
/ [X := ERP_def, Act_path1!]
ERP
[x--]
{Act_node1?}
/ [X := ERP_def, Act_path1!]
X <= 0 / [X := REST_def]
RRP
[x--]
X <= 0 /
[X := RRP_def]
Act_path1
Act_path1
IDLE
{Act_path1?}
/[X=Tcond_def!]
{x<=0?}
{Act_path2?}
/[X=Tcond_def!]
WAIT
[x--]
{X<=0?}
/[Act_node2,X=1!]
{X<=0?}
/[Act_node1,X=1!]
ANTEGRADE
[x--]
Act_path2
{x<=0?}
RETROGRADE
[x--]
{Act_path2?}
DOUBLE
[x--]
{Act_path1?}
Act_node1
Act_node2
Act_node2
REST
[x--]
{X <= 0 | Act_node2?}
/ [X:= ERP_def, Act_path2!]
ERP
[x--]
{Act_node2?}
/ [X:= ERP_def, Act_path2!]
X <= 0 / [X := REST_def]
RRP
[x--]
X <= 0 /
[X := RRP_def]
Act_path2

Heart as a network of timed automata: So What?

- A timed automaton is a mathematical object with clear *semantics:* we understand how it behaves and can analyze that behavior.
- A mathematical model allows us to do a more rigorous checking of correctness than we could without one. We can *prove* things about its behavior.

Model Checking

Does



satisfy

$F\ p\ \wedge q$

Formal notation

?

Linear Temporal Logic

Transition Systems – Mathematical understanding.
Linear Temporal Logic – Expressing requirements formally.
Model checking
UPPAAL