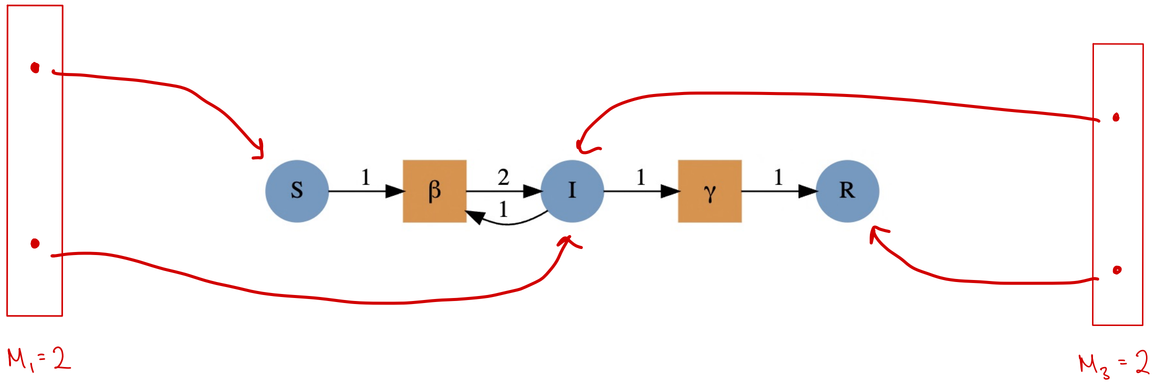
Operad Algebras for composition

Problem: not enough flexibility for n-ary composition or what's exposed



Solution: operad algebras

Operad Algebras for composition

semantics

What kind of component models?

- ODEs
- DDEs
- other DEs
- discrete time dynamical systems
- Petri nets
- stock and flow diagrams

syntax

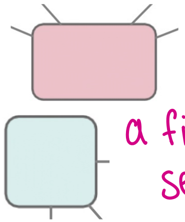
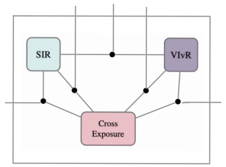
How do they interact?

- Parameterize one model by another
  - process and send information
- Identifying parts of models
  - state variables
  - places
  - stocks

Operad Algebras for composition

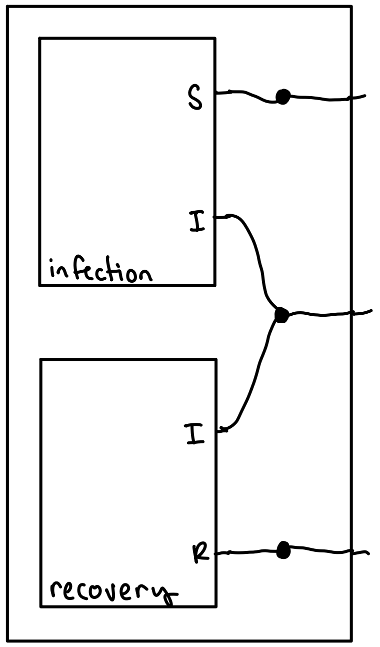
	syntax (operad)	semantics (operad algebra)
interfaces	what are the model interfaces?	For an interface t, what are the models of type t?
interactions	what are the interaction patterns $\phi : S_1, \dots, S_n \to t$?	$\underbrace{\qquad\qquad}_{\text{interaction}} + \underbrace{\qquad\qquad}_{\text{component models}} = \underbrace{\qquad\qquad}_{\text{composite model}}$

Operad Algebras for composition

	syntax (operad)	semantics (operad algebra)
interfaces	 a finite set	
interactions	a UWD e.g. 	

Operad Algebras for composition

Undirected Wiring Diagrams (UWDs)



$$\begin{tikzcd}
 & \text{Port} \arrow[dl] \arrow[dr] & & \text{OuterPort} \arrow[dl] \\
\text{Box} & & \text{Junction} &
\end{tikzcd}$$

```
interaction_pattern = @relation (S, I, R) begin
    infection(S, I)
    recovery(I, R)
end
```

Catlab.Programs.RelationalPrograms.UntypedUnnamedRelationDiagram{Symbol, Symbol} with elements Box = 1:2, Port = 1:4, OuterPort = 1:3, Junction = 1:3

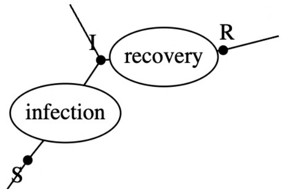
Box	name
1	infection
2	recovery

Port	box	junction
1	1	1
2	1	2
3	2	2
4	2	3

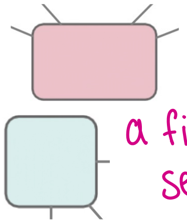
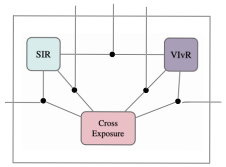
OuterPort	outer_junction
1	1
2	2
3	3

Junction	variable
1	S
2	I
3	R

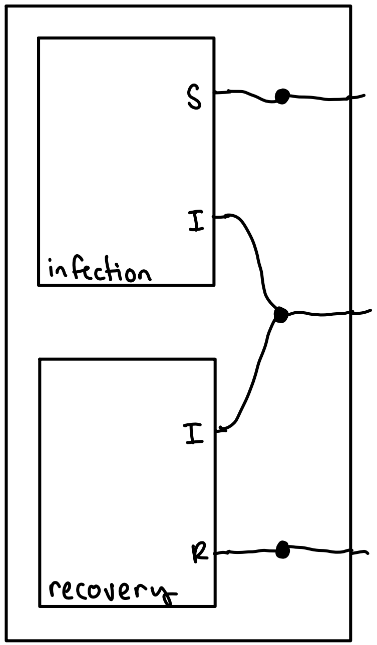
```
draw(interaction_pattern)
```



Operad Algebras for composition

	syntax (operad)	semantics (operad algebra)
interfaces	 a finite set	a model of type M is a Petri net with M exposed species
interactions	a UWD e.g. 	"identify species connected at junctions"

Operad Algebras for composition

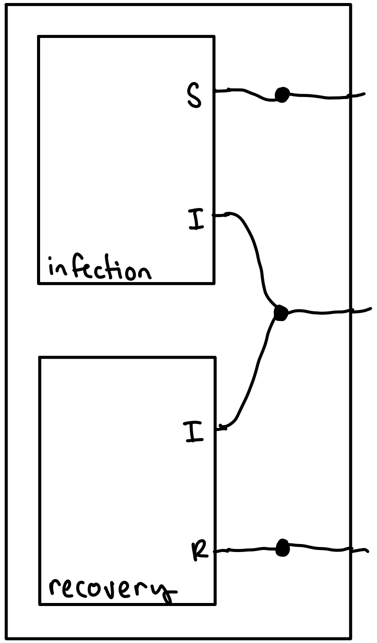


interaction

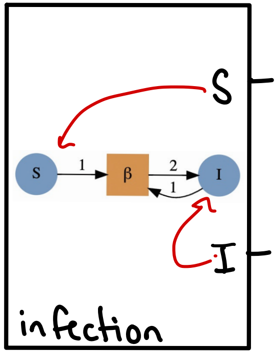
component models

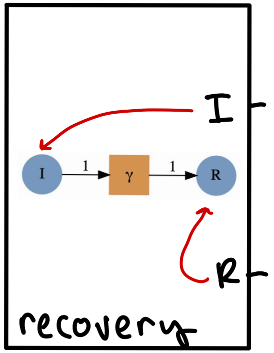
composite model

Operad Algebras for composition



+





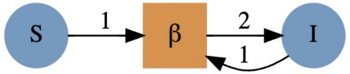
interaction

component models

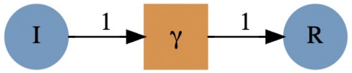
composite model

```
inf_net = LabelledPetriNet([:S, :I],
    :β => ((:S, :I) => (:I, :I))
)

open_inf = Open(inf_net, [:S], [:I])

print("interface type: ", length(legs(open_inf)))
Graph(open_inf)
```

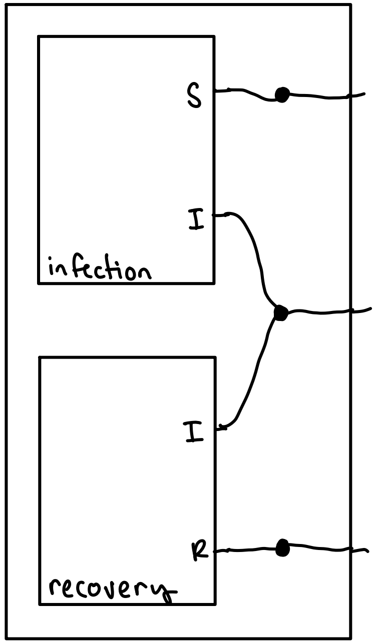
```
interface type: 2
```



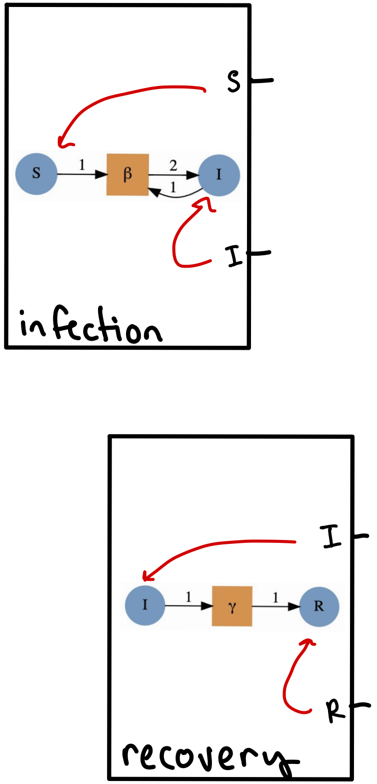
```
rec_net = LabelledPetriNet([:I, :R],
    :γ => (:I => :R)
)

open_rec = Open(rec_net, [:I], [:R])

Graph(open_rec)
```



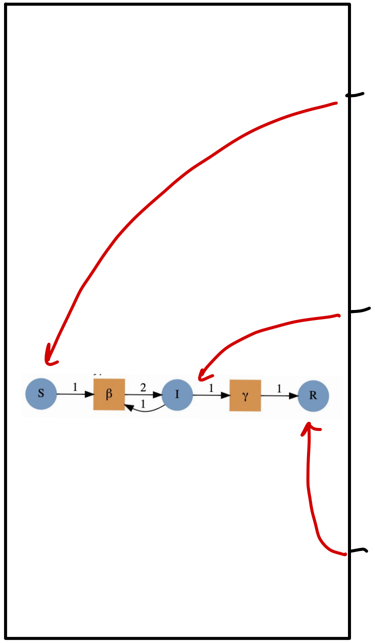
Operad Algebras for composition



+



=



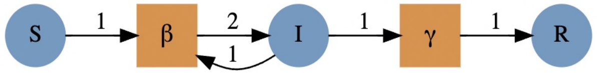
interaction

component models

composite model

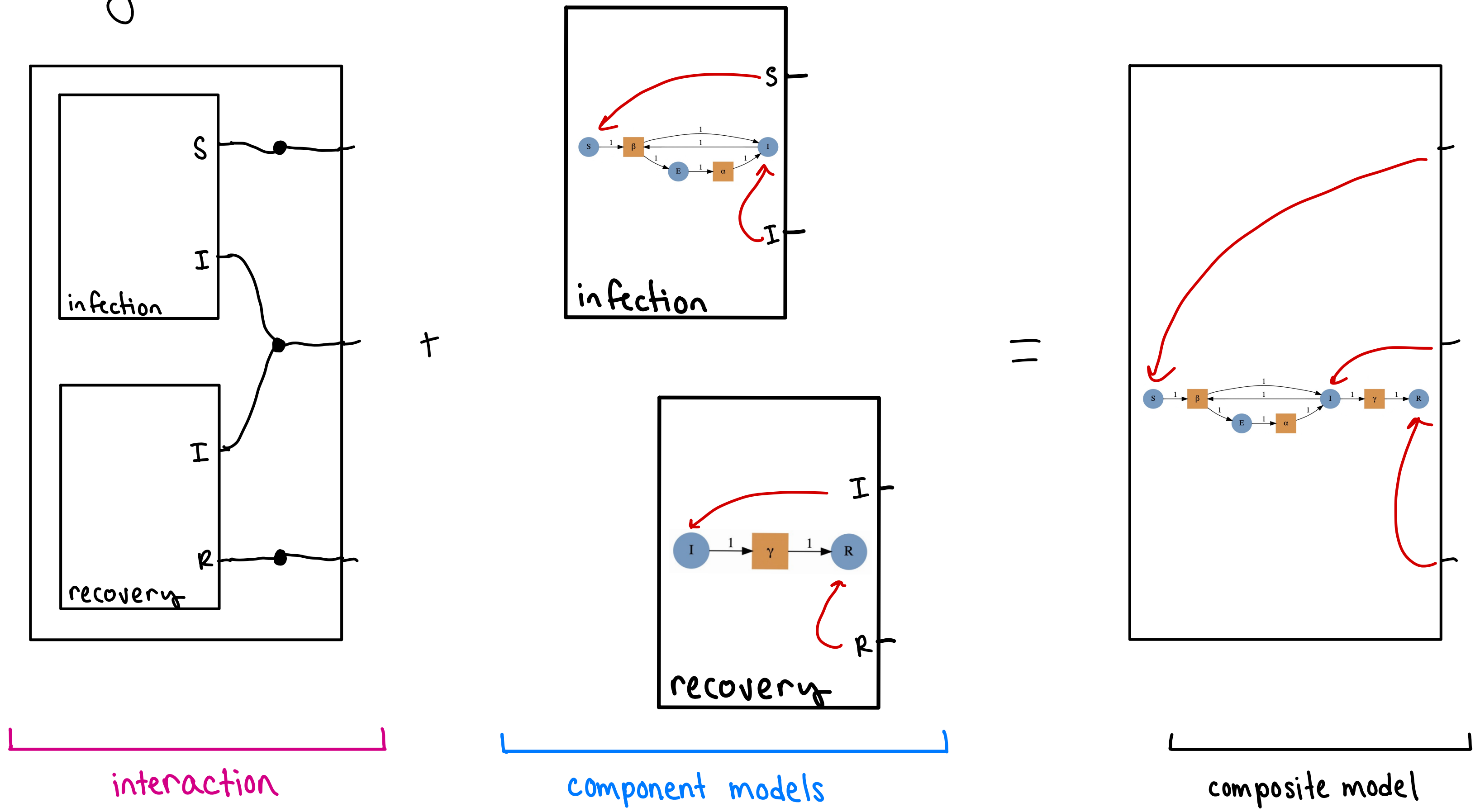
```
sir = oapply(interaction_pattern, Dict([
            :infection => open_inf,
            :recovery => open_rec
]));

print("interface type: ", length(legs(sir)))
Graph(sir)
```

```
interface type: 3
```

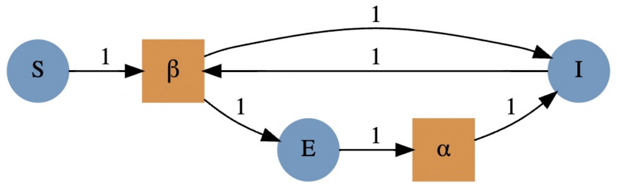


Independence of syntax and semantics

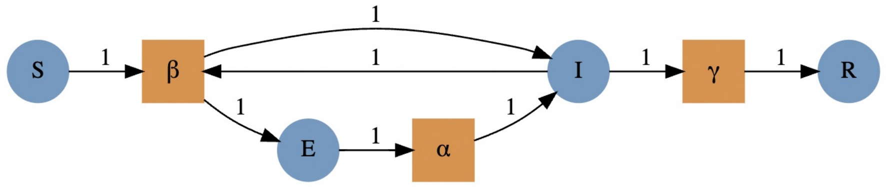
Change the semantics



```
sei_net = LabelledPetriNet([:S, :E, :I],
    :β => ((:S, :I) => (:E, :I)),
    :α => (:E => :I)
)

open_sei = Open(sei_net, [:S], [:I])

Graph(sei_net)
```

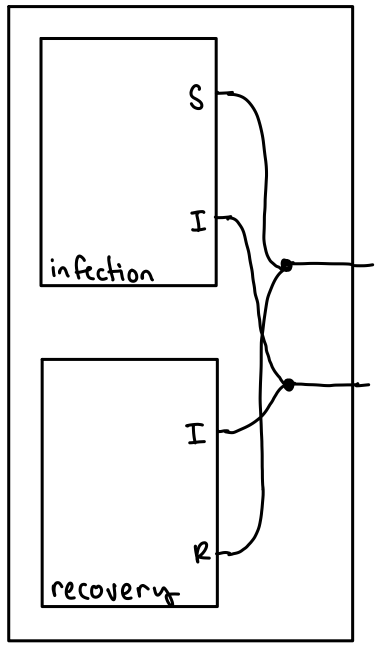


```
seir = oapply(interaction_pattern, Dict([
            :infection => open_sei,
            :recovery => open_rec
]));

Graph(seir)
```

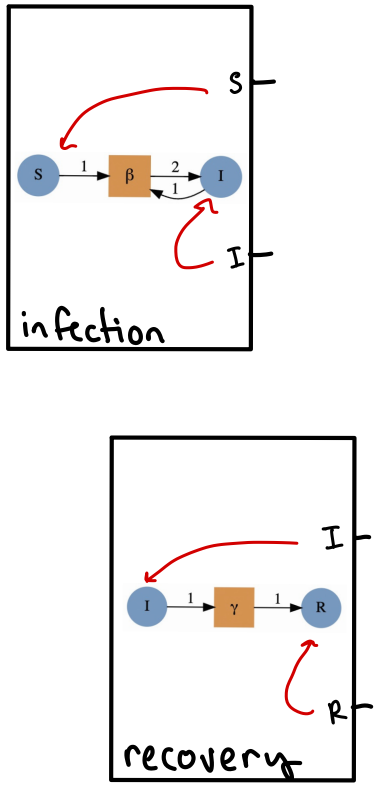


Independence of syntax and semantics

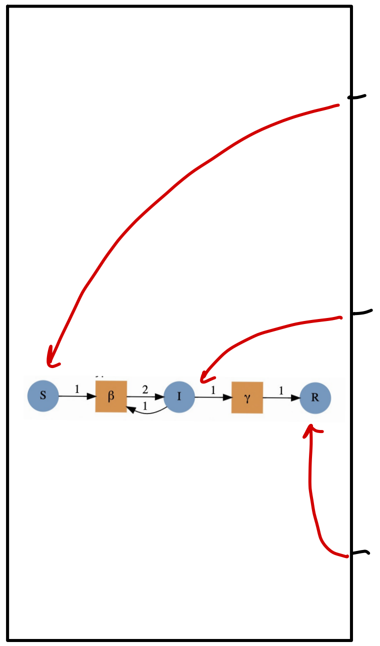
Change the syntax



+



=



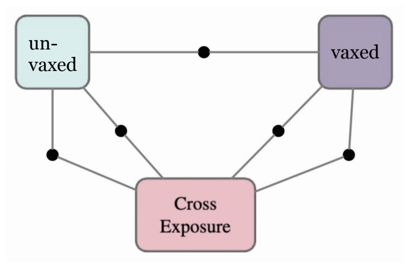
interaction

component models

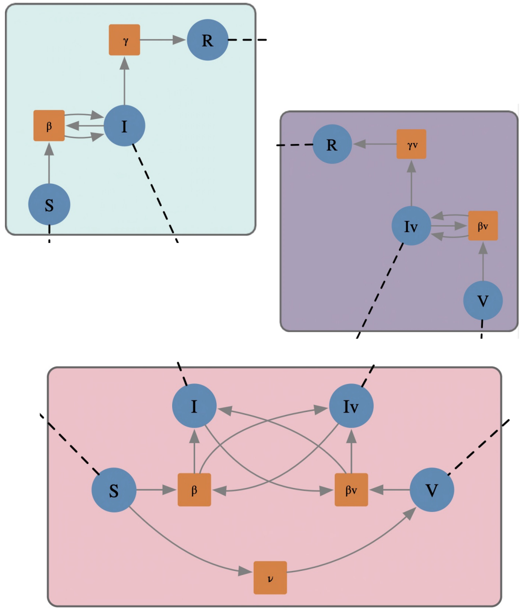
composite model

More Examples

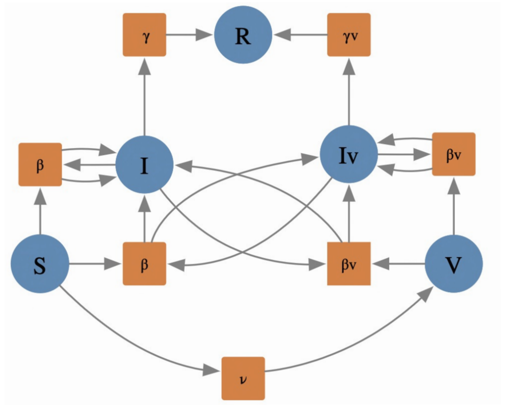
SVIIvR



$+$



$=$



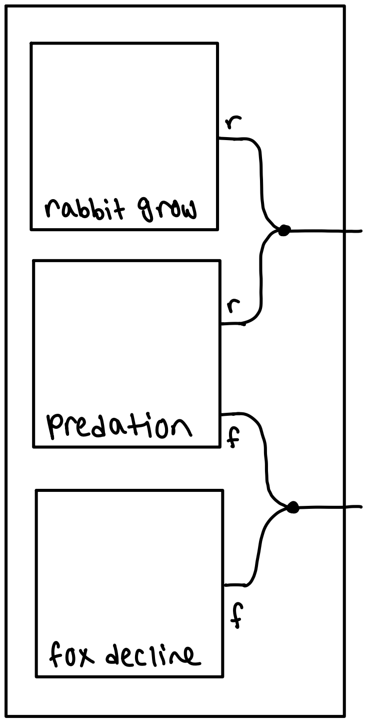
interaction

component models

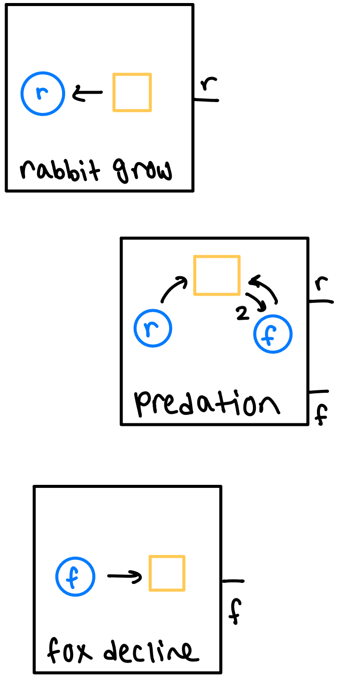
composite model

More Examples

your turn! Lotka-Volterra



+



=

?

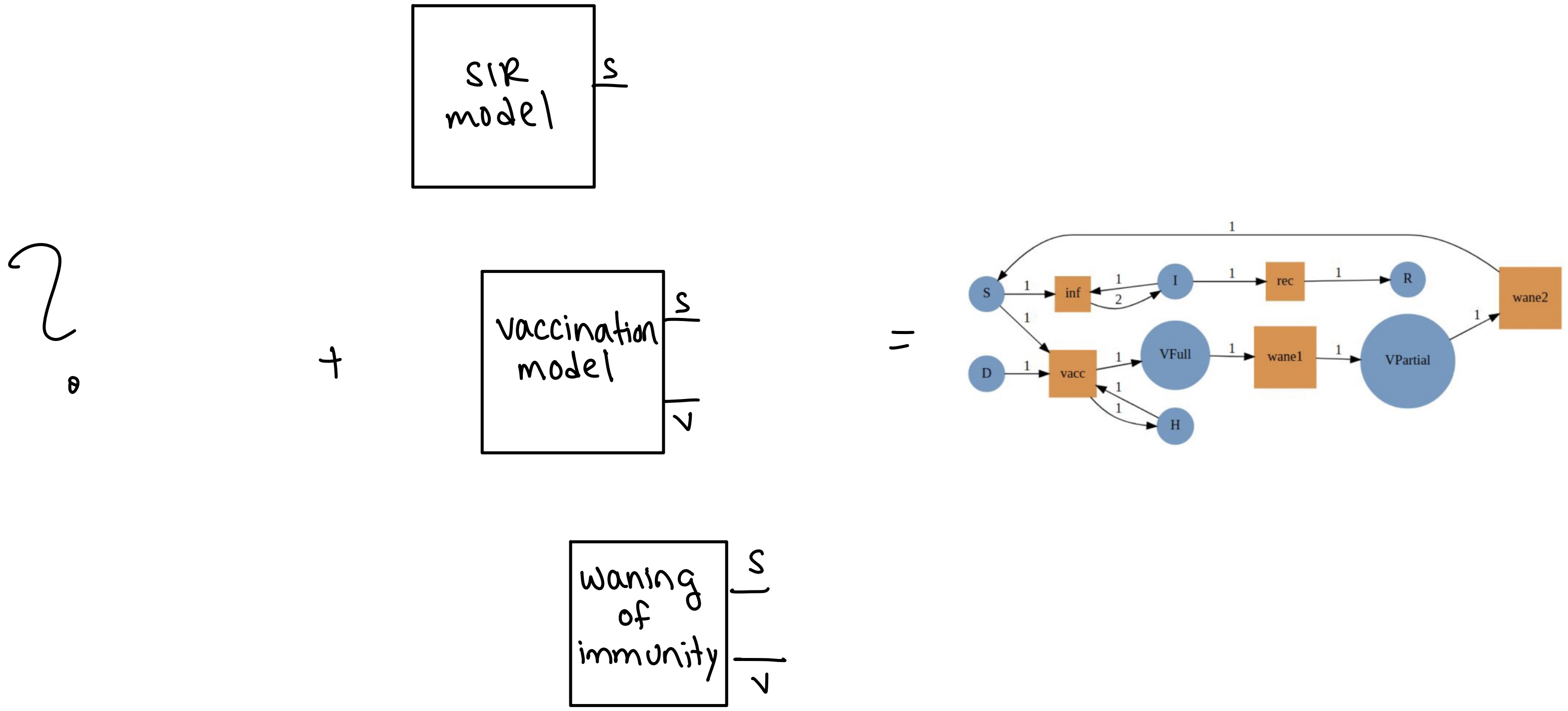
interaction

component models

composite model

More Examples

your turn! Infection + vaccination model



interaction

component models

composite model