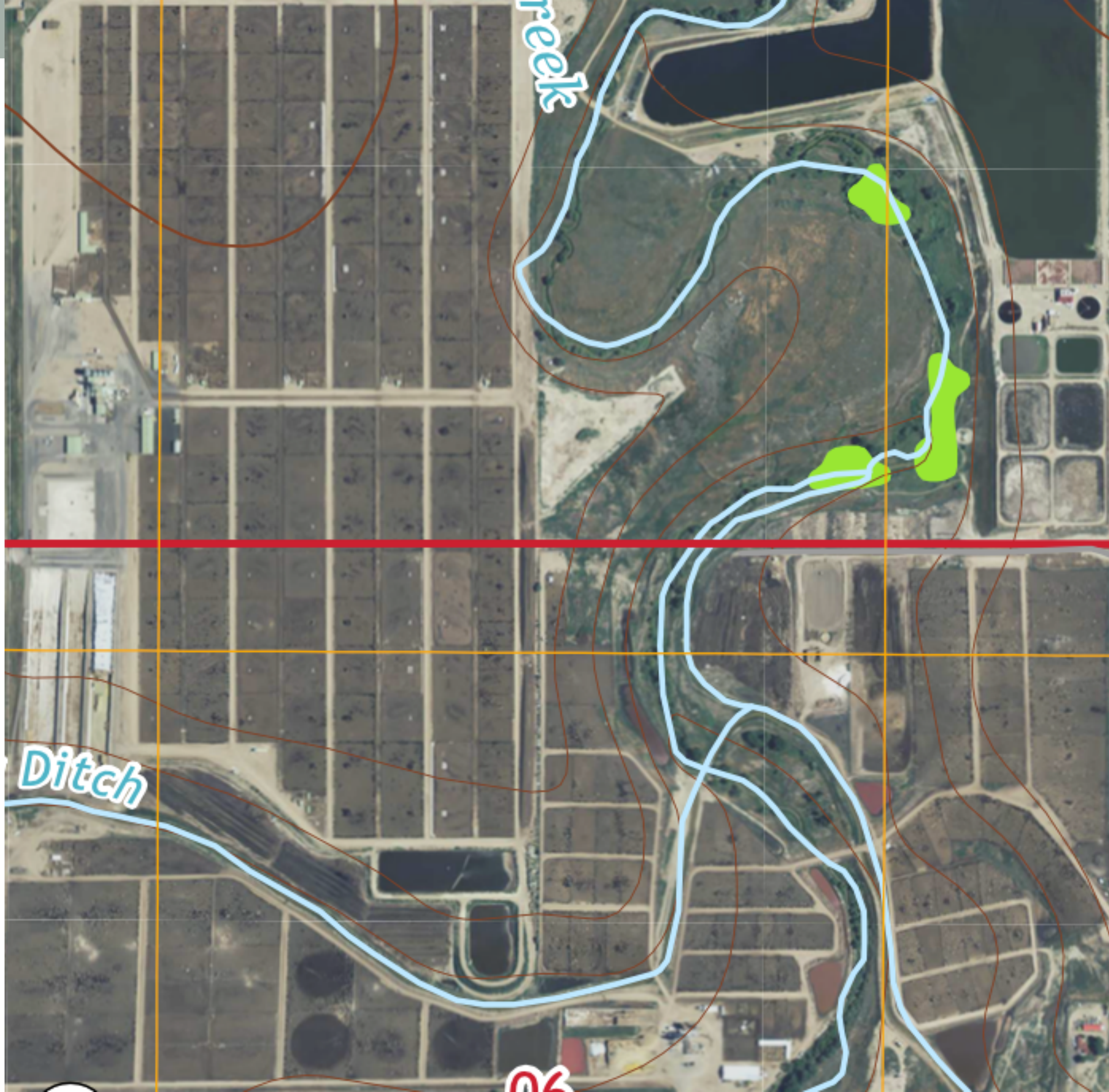
FEEDLOT MODEL

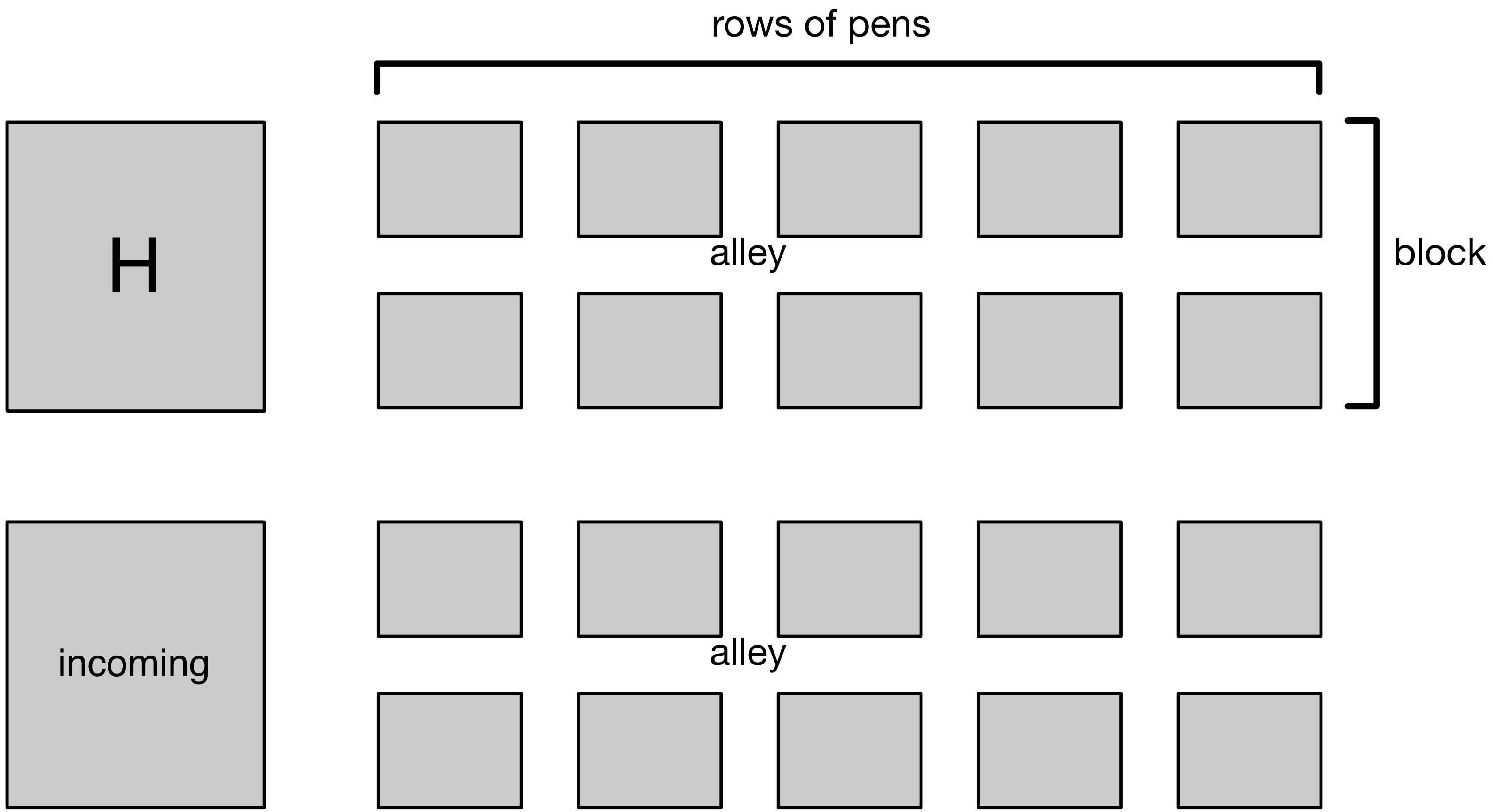
Incremental Development

AFIDD, Cornell, 6 November 2015

Goals

- Construct a feedlot model starting with a theoretical model.
- Track changes in complexity of the model.
- Track where we incorporate uncertainty.


Ditch


rows of pens
H
alley
block
incoming
alley

We Need a Few Things

- Disease state of individuals
- Infection by contact or through vectors
- Turnover of pens (management decisions)
- Movement

- How do we express these things?

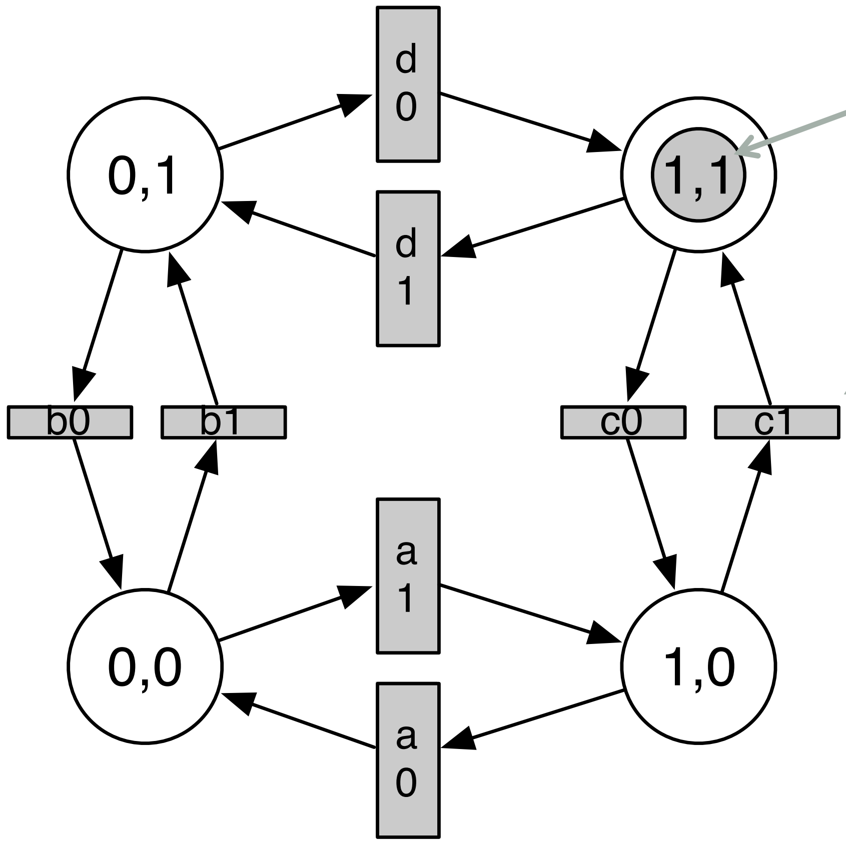
Generalized Stochastic Petri Nets!

- Define Places
- Define Transitions which move Tokens among places when the Transition fires.
- Define Rates for transitions.

- That's your model.

Simple Movement Model

- Places have unique names.
- Only one Transition can fire at any time, the *next* one.

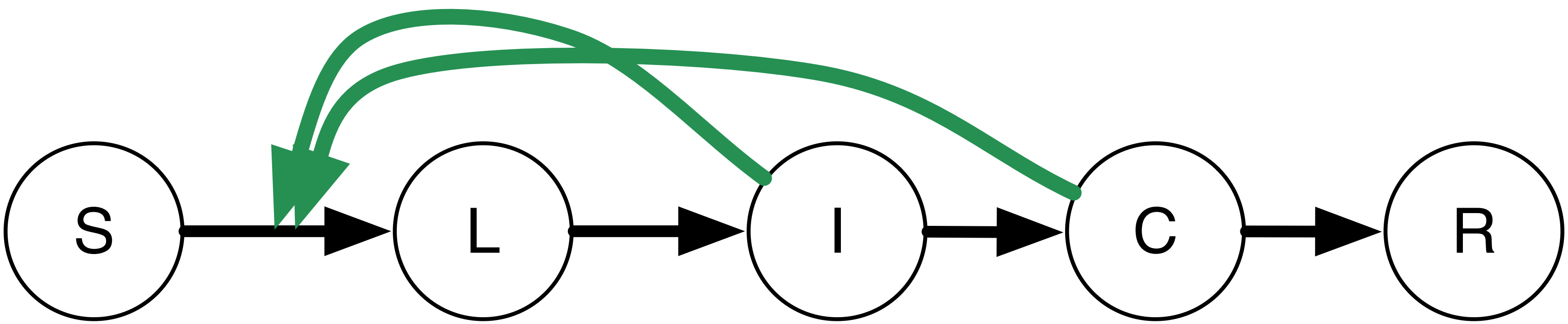


This token is at place (1,1). The location of all Tokens is called the Marking.

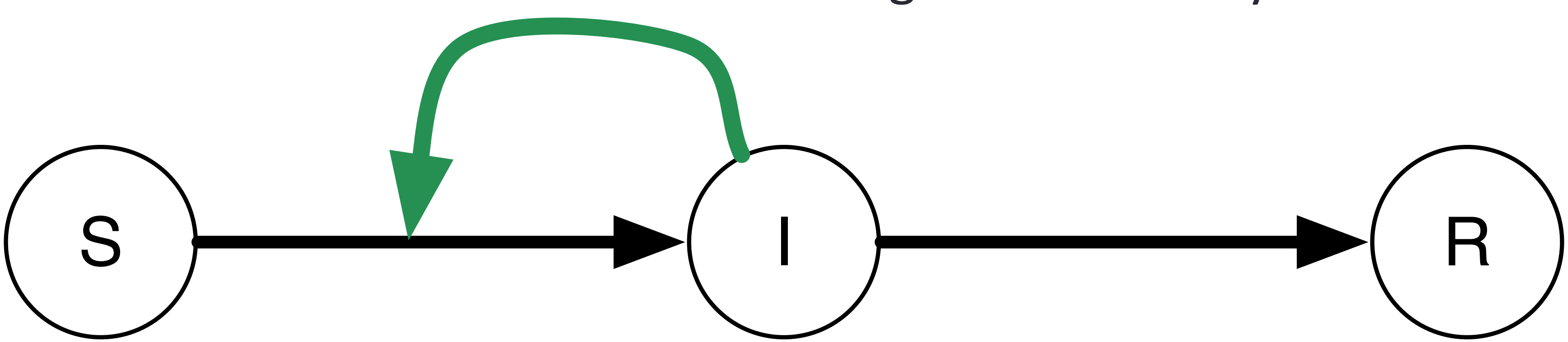
Each transition has three properties:

1) When is it enabled?
2) Once enabled, at what rate does it fire?
3) When it fires, how does it modify the marking?

The Basic Disease Model

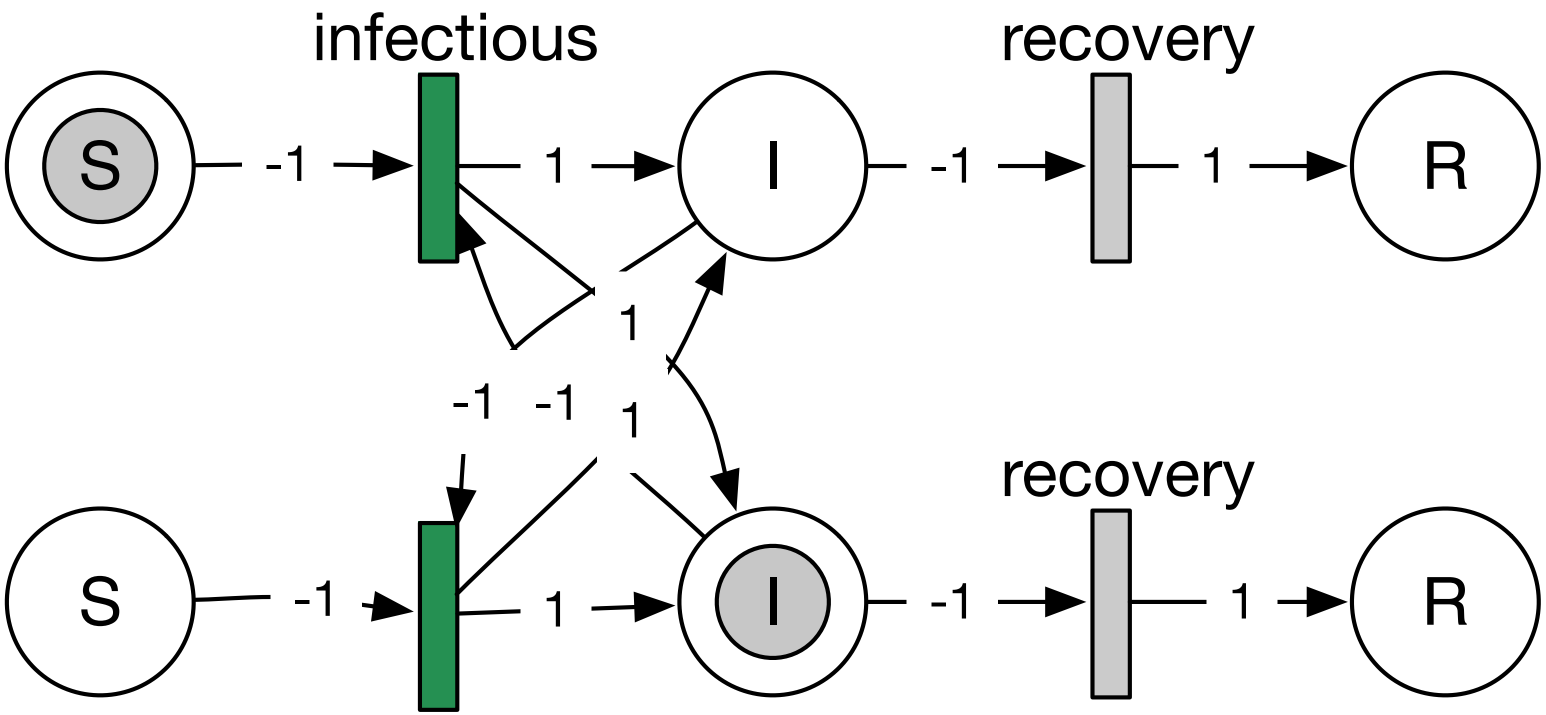


- Susceptible, Latent, Infectious-Subclinical, Infectious-Clinical, Recovered
- Taken from Reeves's WH, taken from Mardones.
- Will draw SIR in GSPN because diagrams look busy.



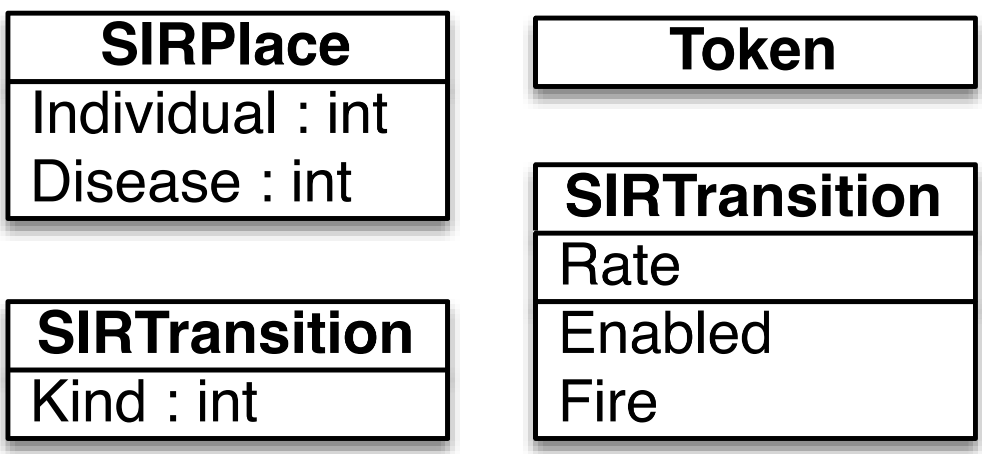
SIR with GSPN

- "-1" means take a token and "1" means give a token.
- Beta=Hazard for infection, Lambda=Hazard for recovery.
- R0=Beta/Lambda
- Note the tokens.



SIR With GSPN

- https://github.com/afidd/Semi-Markov/blob/master/example/sir_graph.cpp
- Place defined by the pair (disease state, individual id).
- Transition has a “kind” used to identify it in post-processing results.



Creating the GSPN

- For [each individual]
  For [each disease state]
  Create Place as a node in the graph.
- For [each individual]
  For [each neighbor]
  Add an infection transition between
  the (individual, infectious) and (neighbor, susceptible)

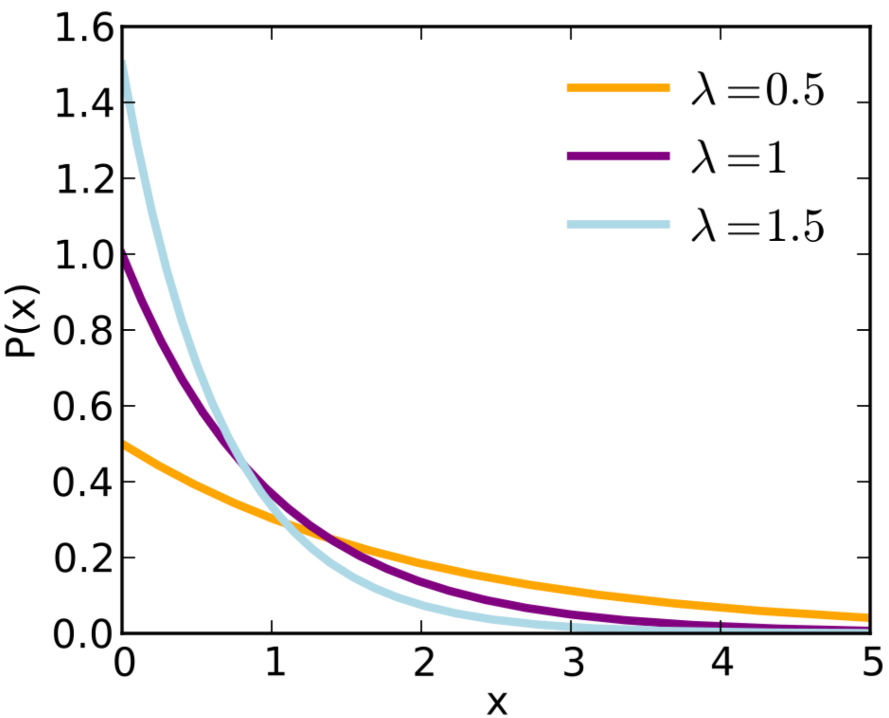
Code for Place

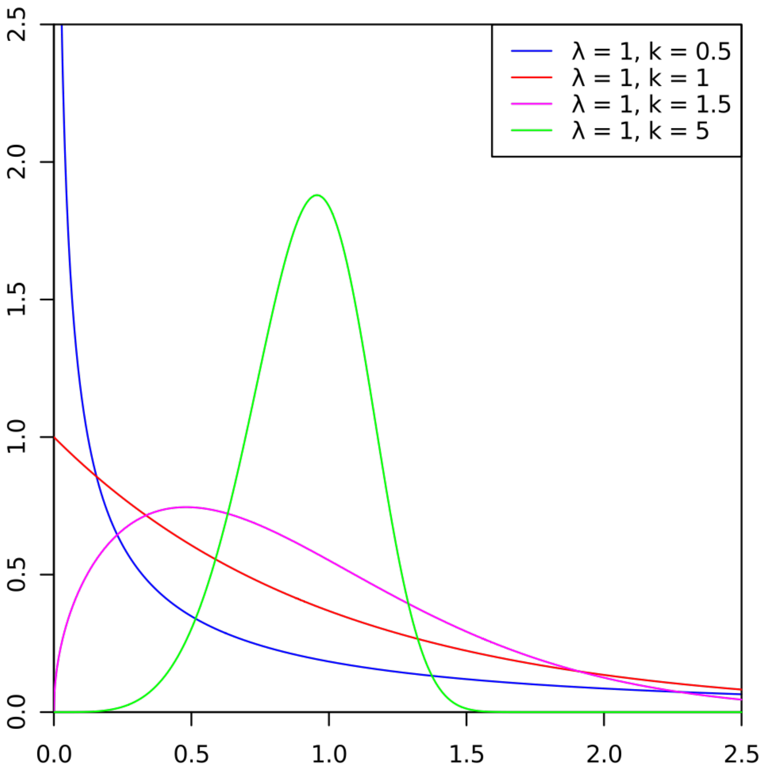
- It's just an identifier for a unique node in a graph.

```
struct SIRPlace
{
  int64_t disease;
  int64_t individual;
  // Some C++ drivel excised.
};
```

Same SIR, Transitions Include Complexity

- Exponential Rate P(t)=1-exp(-beta*t)
- Weibull Rate P(t)=1-exp[(x/\lambda)^k]





From Wikipedia on Exponential and Weibull distributions.

Code for a Transition

This looks at the arrows in the GSPN to see if there is a Token at each arrow that points to this Transition.

```
class InfectNeighbor : public SIRTransition
{
public:
  virtual std::pair<bool, std::unique_ptr<Dist>>
  Enabled(const UserState& s, const Local& lm,
    double te, double t0) override {
    if (lm.template InputTokensSufficient<0>()) {
      return {true, std::unique_ptr<ExpDist>(new
ExpDist(s.params.at(0), te))};
    } else {
      return {false, std::unique_ptr<Dist>(nullptr)};
    }
  }

  virtual void Fire(UserState& s, Local& lm, double t0,
      RandGen& rng) override {
    BOOST_LOG_TRIVIAL(trace) << "Fire infection " << lm;
    lm.template TransferByStochiometricCoefficient<0>(rng);
  }
};
```

Some Transitions do more complicated calculations to determine when they are enabled or what they do when they fire.

Making It Go

- A Marking is a map from Place->Tokens at that place.
- Add tokens to the marking to make the initial state.
- For [all individuals]
  Add a token to the marking at (individual, susceptible).
- For [random individual]
  Move their token from (individual, susceptible)
  to (individual, infectious).
- Hand it to the continuous-time dynamics.

Continuous-Time Main Loop

- “state” is the marking and a list of parameters.
- The output_function looks at the marking every time any event happens, in order to record whatever we want.
- This is the actual code.

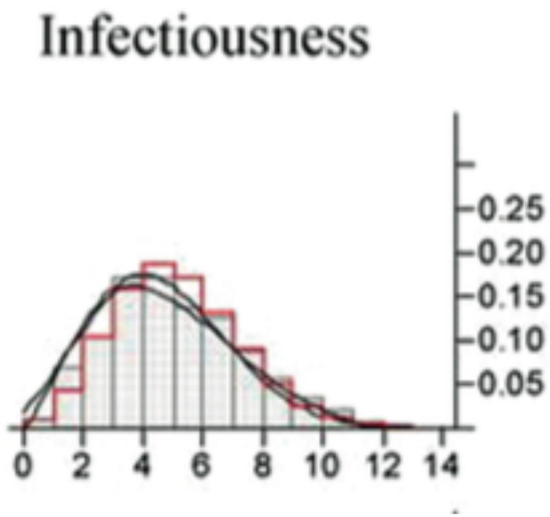
```
SIROutput<SIRState,SIRGSPN> output_function(gspn);

  dynamics.Initialize(&state, &rng);

  bool running=true;
  auto nothing=[](SIRState&)->void {};
  while (running) {
    running=dynamics(state);
    if (running) output_function(state);
  }
  output_function.final(state);
```

Recovery Transition – Thinking about FMDV

- Observations define moments of transition.
- Entry and exit from Infectious state by serum, clinical observation.
- A histogram of observations is non-Exponential.
- Mardones et al tries to provide those rates.
- The more we know about progress of disease by breed, sex, age, the more precise these curves are.



Mardones et al. 2010

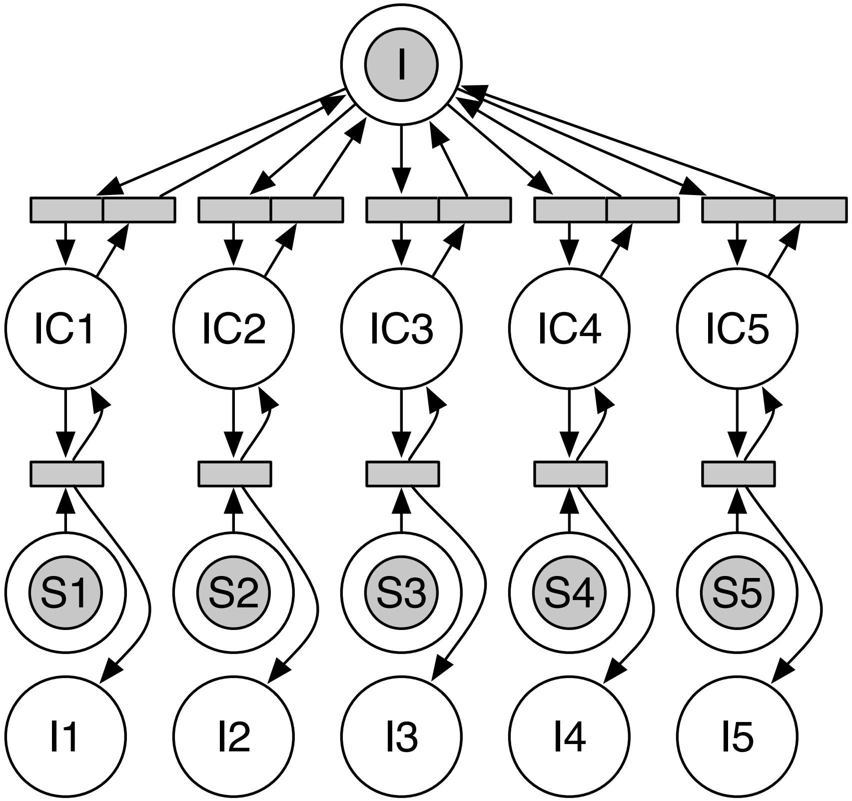
- Stochastic simulation averages over this uncertainty.

Infection Transition

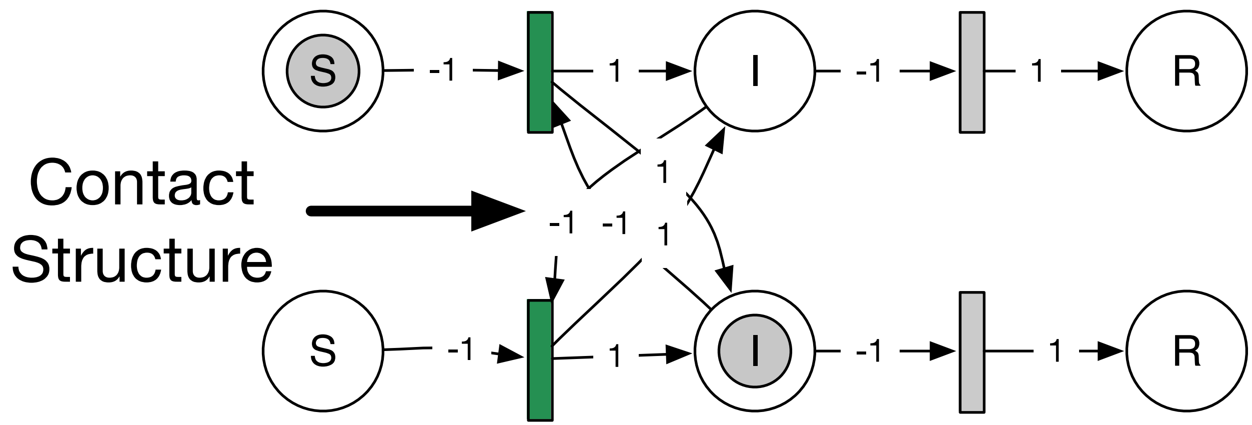
- Often refer to “beta,” the hazard rate, or propensity, for infecting a neighbor.
- It’s the rate parameter of an exponential transition.
- We encapsulate in beta both *how the herd moves* and *how infectious the disease is.*
- Product of (contact rate) x (-log of probability that transmission doesn’t happen).

What if Contact Were Explicit?

- Product of (contact rate) x (-log of probability that transmission doesn't happen).
- Our mental picture is:
  - Contact
  - Possibly infect
  - Contact another
  - Possibly infect
- SIR averages over this as a rate of contact.
- Similar to adequate contacts representation.



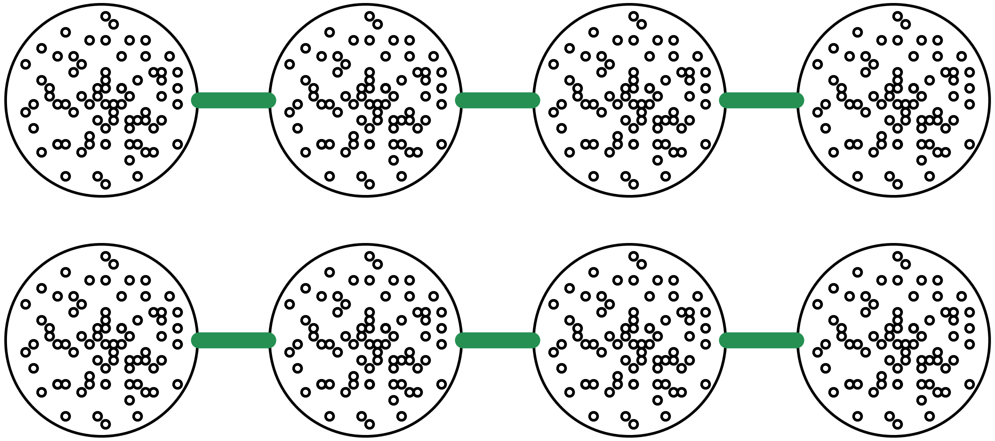
Contact Structure is Connectivity in GSPN



- Structure of GSPN establishes what *can* happen.
- Rates define *how quickly* it might happen.
- We choose for the model structure and rates for infection.
- Pen rider, airborne, wandering varmints, 4H club visits, hospital pens.

Pen Separation by Concrete or Fenceline

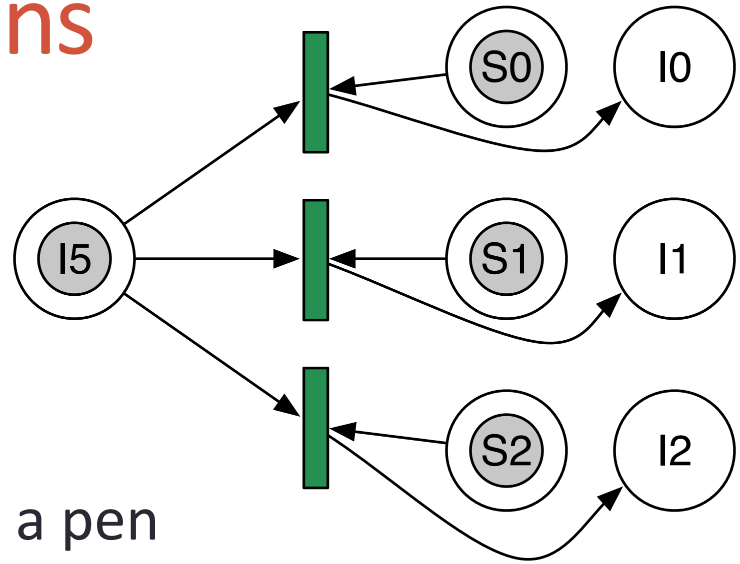
- Concrete wouldn't be all-to-all contact among animals.
- Fenceline would have same infectivity per contact but smaller contact rate, leading to a different beta.



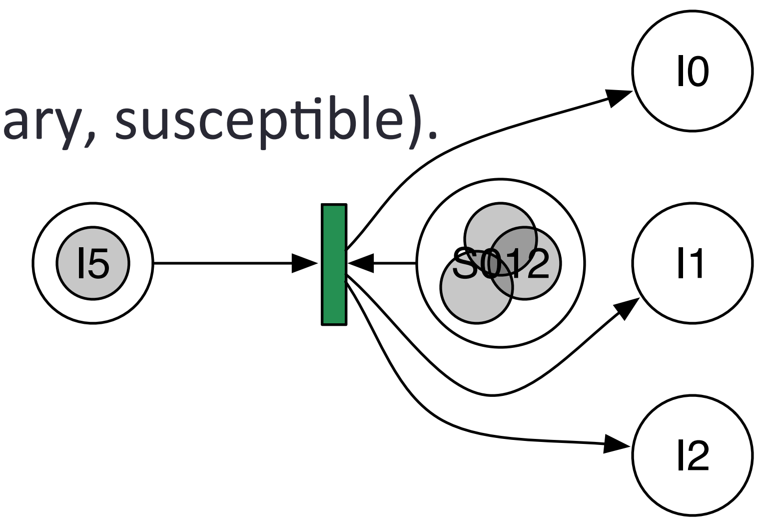
Fenceline Contact in Code

- For [each pen]
  For [each other pen]
  if [pens share a fenceline]
  Create an infection transition for each pair of individuals.
- Code is at
- https://github.com/adolgert/feedlot/blob/master/src/seir_exp.cpp#L365
- How do you account for reduced spread across fenceline compared to within-pen?
- How do concrete walls change the model?

Coalesced Transitions

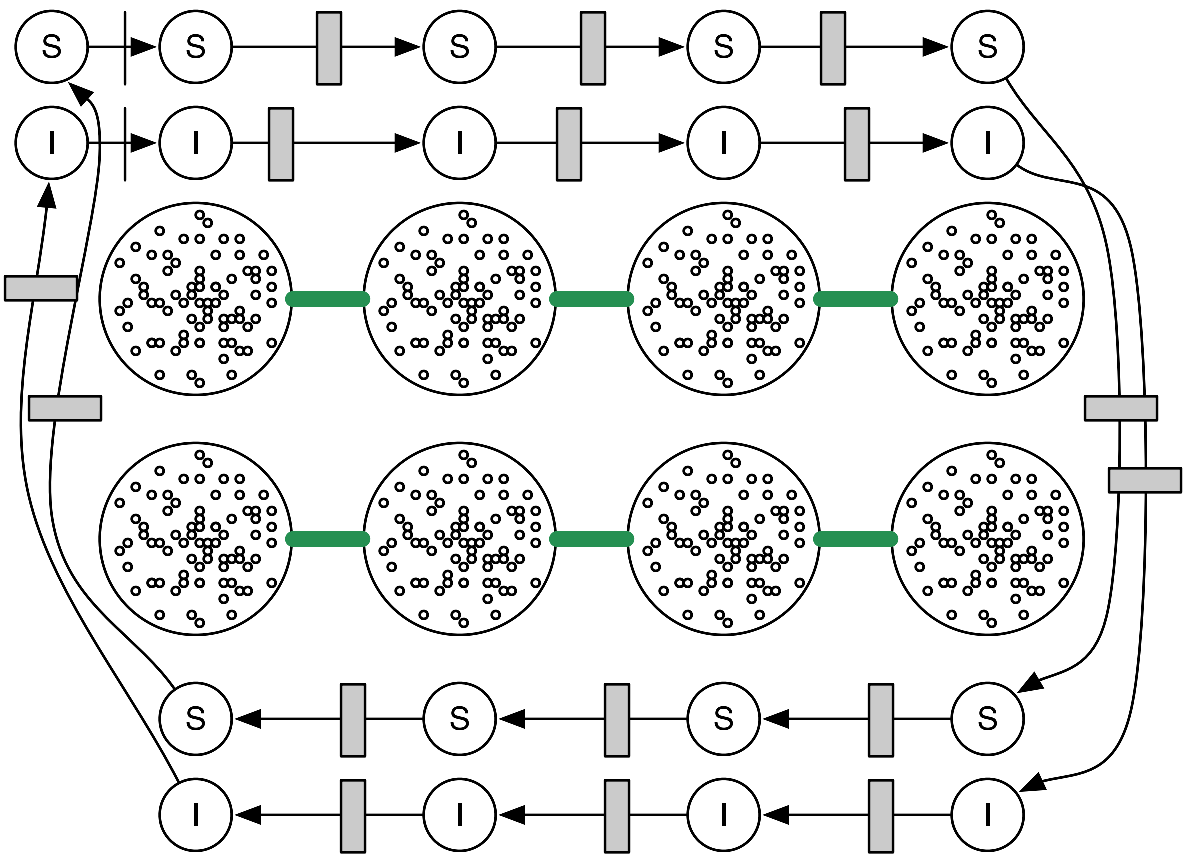


- Susceptible->Exposed transitions can be combined.
- Infectious->Recovered must be separate.
- Put all Susceptible tokens for a pen at the same place.
- Label it as (Pen id, pen summary, susceptible).
- This is the “rider” code.
- Only an optimization.



Pen rider/walker

- Move rider by moving token among places.
- Sick animals deposit on rider, which deposits on animals in same or next pen.
- Rider stays becomes uninfectious at some rate.



Transitions for the Pen Rider

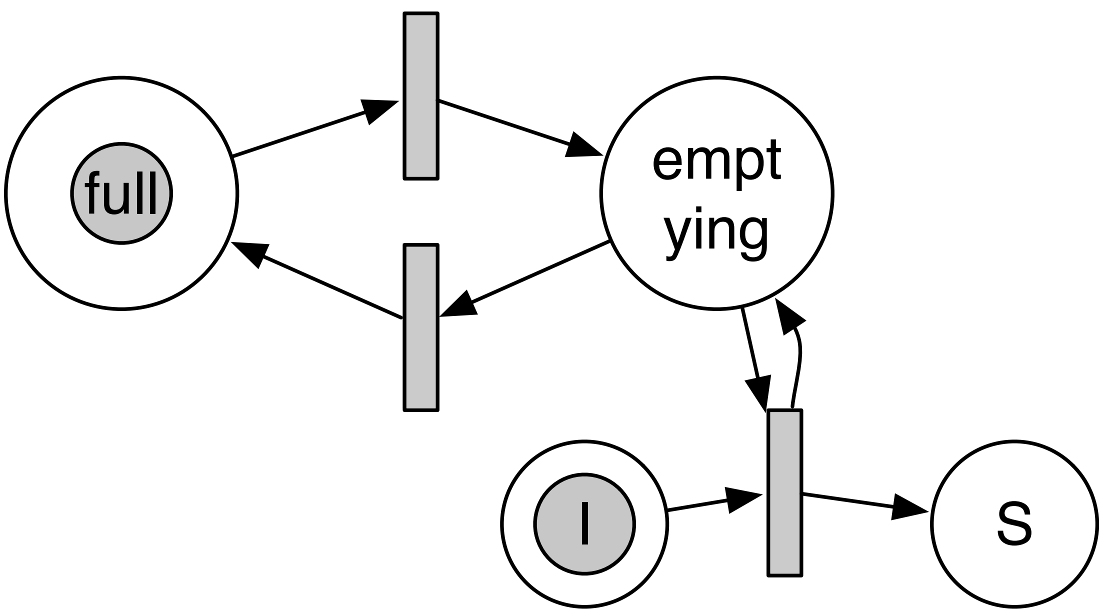
- The structure of the GSPN graph defines what interactions can happen.
- How do we make a rider that goes once a day?
- What choices do we have to make and where do they go in the code?
  - Rider moves.
  - Rider picks up infectious particles from infectious animals.
  - Rider contacts susceptible animals.
  - Rider's infectiousness wanes.

The Rider Exerts Force of Infection

- We can calculate R0 with the rider as a function of the rate of pickup, the rate of movement, and rate of infection.
- Either by measurement or analytically.
- The rider is a controlled contact rate. It represents the most ordered kind of mixing in the population.
- We expect a linear spread of disease to be the slowest kind of spread (structurally).

Demographics

- Replace cattlebeasts in pen every 133 days.
- Simple model triggers transition from any disease state to susceptible every 133 days, on a per-pen basis.
- This graph shows Full and Emptying places for a single pen and *just one* disease state transition for *just one* animal.



Transition Rates for Pen Replacement

- Every 133 days for each pen.
- Pen replacement is distributed through time.
- Create an offset at the start of the simulation.
- The transition rate says the pen will replace in some six-hour interval on the 133rd day.
- Uniform distribution with an offset, which you put in a Transition, which you hook to the Pen Replacement places.

Places for the Full Model

- Each place is a triple.
- (individual id, individual-type, disease state)
- (pen id, pen-summary-type, disease state)
- (pen id, pen-rider, disease-carrying state)
- (pen id, demographics-type, filling/emptying)

- When the Token represents an individual, it carries the individual’s ID within the pen.

Initial State Choices

- How many animals in S, E, I, C, R?
- Are they all in the same pen? https://github.com/adolgert/feedlot/blob/master/src/seir_exp.cpp#L557
- Do demographics introduce new sources of infection?
- Turn the rider on/off just by putting a Token in a Rider place (or not).
- Turn demographics on/off similarly.
- Empty pens have no Tokens in the marking associated with their places.