FUNMANAbstraction

Dan Bryce

September 27, 2024

1 Background

Definition 1 *A Petrinet Ω is a directed graph (V, E) with vertices $V = (V_x, V_z)$ partitioned into sets V_x of state vertices and V_z of transition vertices, and edges $E = (E_{in}, E_{out})$ partitioned into sets E_{out} of flow-out and E_{in} flow-in edges.*

Definition 2 *A flow-out edge $e \in E_{out}$ comprises a pair of vertices (v_x, v_z), where $v_x \in V_x$ is a state vertex, $v_z \in V_z$ is a transition vertex, and the flow is directed from v_x to v_z.*

Definition 3 *A flow-in edge $e \in E_{in}$ comprises a pair of vertices (v_z, v_x), similar to a flow-out edge, except that the flow is directed from v_z to v_x.*

Definition 4 *The ODE semantics Θ of the Petrinet Ω defines a tuple $(P, X, Z, \mathcal{I}, \mathcal{P}, \mathcal{X}, \mathcal{Z}, \mathcal{R})$ where*

- *P is a set of parameters;*
- *X is a set of state variables;*
- *Z is a set of transitions;*
- *$\mathcal{I} : S \to \mathbb{R}$ assigns the initial value of state variables to a real number;*
- *$\mathcal{P} : P \to \mathbb{R} \cup \mathbb{R} \times \mathbb{R}$ assigns parameters to a real number, or a pair of real numbers defining an interval;*
- *$\mathcal{X} : X \to V_x$ assigns state variables to state vertices;*
- *$\mathcal{Z} : Z \to V_z$ assigns transtions to transition vertices; and*
- *$\mathcal{R} : \mathbf{P} \times \mathbf{X} \times Z \to \mathbb{R}$ defines the rate of each transition in $x \in X$ in terms of the set of parameter vectors $\mathbf{P}$ and state variable vectors $\mathbf{X}$.*

The elements of the Petrinet Ω and semantics Θ define the partial derivative $\frac{d\mathbf{x}}{dt}$, so that for each state variable $x \in X$:

$$\frac{dx}{dt} = \sum_{v_z \in V_z^{in(x)}} \mathcal{R}(\mathbf{p}, \mathbf{x}, z) - \sum_{v_z \in V_z^{out(x)}} \mathcal{R}(\mathbf{p}, \mathbf{x}, z) \tag{1}$$

where $V_z^{in(x)} = \{v_z \in V_z | (v_z, v_x) \in E_{in}\}$ and $V_z^{out(x)} = \{v_z \in V_z | (v_x, v_z) \in E_{out}\}$ are the transition vertices that flow in and out of the vertex v_x, respectively. We denote by $\nabla_{\Omega,\Theta}(\mathbf{p}, \mathbf{x}, t) = (\frac{dx_1}{dt}, \frac{dx_2}{dt}, \ldots)^T$, the gradient comprised of components in Equation (1).

Using the partial derivatives defined by the Petrinet graph and semantics, we can define the state vector at given time $t + dt$ with the forward Euler method as:

$$\begin{aligned}
\frac{d\mathbf{x}}{dt} &= \nabla_{\Omega,\Theta}(\mathbf{p}, \mathbf{x}, t) \\
\frac{\mathbf{x}(t+dt) - \mathbf{x}(t)}{dt} &= f_{\Omega,\Theta}(\mathbf{p}, \mathbf{x}, t) \\
\mathbf{x}(t+dt) &= f_{\Omega,\Theta}(\mathbf{p}, \mathbf{x}, t)dt + \mathbf{x}(t)
\end{aligned}$$

Definition 5 *An abstraction (Θ', Ω') of a Petrinet and the associated semantics (Θ, Ω) that is produced by the abstraction operator A has the following properties:*

- *State: For each $x \in X$, $A(x) = x'$, where $x' \in X'$. For each vertex $v_x \in V_x$, $A(v_x) = v'_x$ where $v'_x \in V'_x$. For each $x \in X$ where $\mathcal{X}(x) = V_x$, $A(x) = x'$, and $A(v_x) = v'_x$, then $\mathcal{X}'(x') = v'_{x'}$. For each $x' \in X'$, $\mathcal{X}'(x') = \sum_{x \in X : A(x) = x'} \mathcal{X}(x)$.*
- *Parameters: For each $p \in P$, $A(p) = p'$, where $p' \in P'$. For each $p' \in P'$, $\mathcal{P}'(p') = \sum_{p \in P : A(p) = p'} \mathcal{P}(p)$.*
- *Transitions: For each $z \in Z$, $A(z) = z'$, where $z' \in Z'$. For each vertex $v_z \in V_z$, $A(v_z) = v'_z$, where $v'_z \in V'_z$. For each $z \in Z$, where $\mathcal{Z}(z) = v_z$, $A(z) = z'$, and $A(v_z) = v'_z$, then $\mathcal{Z}'(z') = v'_{z'}$.*
- *In Edges: For each edge $(v_z, v_x) \in E_{in}$, $A((v_z, v_x)) = (v'_z, v'_x)$, $A(v_x) = v'_x$, and $A(v_z) = v'_z$, where $(v'_z, v'_x) \in E'_{in}$;*
- *Out Edges: For each edge $(v_x, v_z) \in E_{out}$, $A((v_x, v_z)) = (v'_x, v'_z)$; $A(v_x) = v'_x$, and $A(v_z) = v'_z$, where $(v'_x, v'_z) \in E'_{out}$;*
- *Transition Rates: For each $z' \in Z'$, $\mathcal{R}'(\mathbf{p}', \mathbf{x}', z') = \sum_{z \in Z : A(z) = z'} \mathcal{R}(\mathbf{p}, \mathbf{x}, z)$.*

The abstraction (Θ', Ω') similarly defines the gradient $\nabla_{\Omega',\Theta'}(\mathbf{p}', \mathbf{x}', t) = (\frac{dx'_1}{dt}, \frac{dx'_2}{dt}, \ldots)^T$, in terms of Equation 1. The abstraction thus expresses the gradient by aggregating terms from the base Petrinet and semantics. It preserves the flow on transitions, but expresses the transition rates in terms of the base states. As such, the abstraction compresses the Petrinet graph structure, but at the cost of expanding the expressions for transition rates. Moreover, the transition rates refer to state variables and parameters that are not expressed directly by the Petrinet and semantics, and by extension, the gradient.

We modify the abstraction in what we call a *bounded abstraction*, so that it refers to the abstract, and not the base, Petrinet and semantics. This bounded abstraction replaces base elements with corresponding bounded elements. For example, if $A(x_1) = x'$ and $A(x_2) = x'$ (x_1 and x_2 are base variables represented by x' in the abstraction), a possible transition rate could be of the form $\mathcal{R}'(\mathbf{p}', \mathbf{x}', z') = p_1 x_1 + p_2 x_2$. By construction, we know that $x_1 + x_2 = x'$. However, in general $p_1 \neq p_2$, and we cannot say that $p_1 x_1 + p_2 x_2 = p' x'$ for some p'. Yet, if we replace p_1 and p_2 by $p^{ub} = \max(p_1, p_2)$, then $p^{ub} x_1 + p^{ub} x_2 \geq p' x'$. Simplifying, we get $p^{ub} x_1 + p^{ub} x_2 = p^{ub}(x_1 + x_2) = p^{ub} x' \geq p' x'$. A similar argument can be made where $p^{lb} = \min(p_1, p_2)$ and we find that $p^{lb} x' \leq p' x'$.

By introducing the bounded parameters, we no longer rely upon the base state variables or parameters. However, in tracking the effect of the bounded parameters, the bounded abstraction must also track bounded rates and bounded state variables. The resulting bounded abstraction thus over-approximates the abstraction and base model, wherein we can derive bounds on the state variables at each time, which may correspond to a larger (hence over-approximation) set of state trajectories.

Definition 6 *A bounded abstraction (Θ^B, Ω^B) of an abstraction (Θ', Ω') of (Θ, Ω) replaces each element of (Θ', Ω') by a pair of elements denoting the lower and upper bound of that element (and referred to with the "lb" and "ub" superscripts). The bounded abstraction defines:*

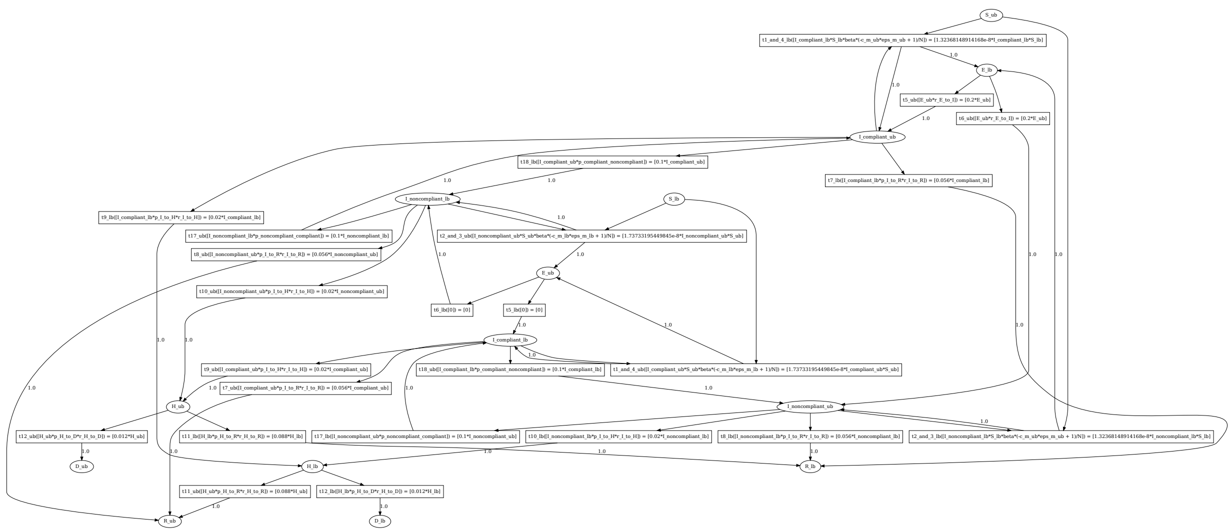

Figure 1: SEIRHD Model Petrinet

- *State: For each* $x' \in X'$, $x^{lb}, x^{ub} \in X^B$. *For each* $v'_{x'} \in V'_x$, $\mathcal{X}^B(x^{lb}) = v^B_{x^{lb}}$ *and* $\mathcal{X}^B(x^{ub}) = v^{ub}_x$. *For each* $x^{lb}, x^{ub} \in X^B$, $\mathcal{I}^B(x^{lb}) = \mathcal{I}^B(x^{ub}) = \mathcal{I}'(x')$.
- *Parameters: For each* $p' \in P'$, *let* $\mathcal{P}^B(p^{lb}) = \min_{p \in P : A(p) = p'} \mathcal{P}(p)$ *and* $\mathcal{P}^B(p^{ub}) = \max_{p \in P : A(p) = p'} \mathcal{P}(p)$.
- *Transitions: For each* $z' \in Z'$, $z^{lb}, z^{ub} \in Z^B$. *For each vertex* $v_z \in V_z$, $v_{z^{lb}}, v_{z^{ub}} \in V^B_z$.
- *In Edges: For each edge* $(v_{z'}, v_{x'}) \in E'_{in}$, $(v_{z^{lb}}, v_{x^{lb}}), (v_{z^{ub}}, v_{x^{ub}}) \in E^B_{in}$.
- *Out Edges: For each edge* $(v_{x'}, v_{z'}) \in E'_{out}$, $(v_{x^{ub}}, v_{z^{lb}}), (v_{x^{lb}}, v_{z^{ub}}) \in E^B_{out}$.
- *Transition Rates: For each* $z^{lb} \in Z^B$, $\mathcal{R}^B(\mathbf{p}^B, \mathbf{x}^B, z^{lb}) = \min_{z \in Z : A(z) = z'} \mathcal{R}(\mathbf{p}, \mathbf{x}, z)$ *(replacing* $\mathbf{p}$ *and* $[\mathbf{x}]$ *of the minimal rate by the elements in* $\mathbf{p}^B$ *and* $\mathbf{x}^B$ *respectively, which minimize the rate), and* $\mathcal{R}^B(\mathbf{p}^B, \mathbf{x}^B, z^{ub}) = \max_{z \in Z : A(z) = z'} \mathcal{R}(\mathbf{p}, \mathbf{x}, z)$ *(similarly replacing* $\mathbf{p}$ *and* $\mathbf{x}$ *of the maximal rate by the elements in* $\mathbf{p}^B$ *and* $\mathbf{x}^B$ *respectively, which maximize the rate).*

2 Stratification Abstraction

The SIERHD model from the August monthly demo uses the model summarized by the Petrinet diagram in Figure 1.

The following transitions connect the variables S_c, S_{nc}, E_c, E_{nc}, I_c, and I_{nc}, R, H, D:

$$
\begin{aligned}
t_1 : (I_c, S_c) &\xrightarrow{r_1} (I_c, E_c) \\
t_2 : (I_{nc}, S_c) &\xrightarrow{r_2} (I_{nc}, E_c) \\
t_3 : (I_{nc}, S_{nc}) &\xrightarrow{r_3} (I_{nc}, E_{nc}) \\
t_4 : (I_c, S_{nc}) &\xrightarrow{r_4} (I_c, E_{nc}) \\
t_5 : (E_c) &\xrightarrow{r_5} (I_c) \\
t_6 : (E_{nc}) &\xrightarrow{r_6} (I_{nc}) \\
t_7 : (I_c) &\xrightarrow{r_7} (R) \\
t_8 : (I_{nc}) &\xrightarrow{r_8} (R) \\
t_9 : (I_c) &\xrightarrow{r_9} (H) \\
t_{10} : (I_{nc}) &\xrightarrow{r_{10}} (H) \\
t_{11} : (H) &\xrightarrow{r_{11}} (R) \\
t_{12} : (H) &\xrightarrow{r_{12}} (D) \\
t_{13} : (S_{nc}) &\xrightarrow{r_{13}} (S_c) \\
t_{14} : (S_c) &\xrightarrow{r_{14}} (S_{nc}) \\
t_{15} : (E_{nc}) &\xrightarrow{r_{15}} (E_c) \\
t_{16} : (E_c) &\xrightarrow{r_{16}} (E_{nc}) \\
t_{17} : (I_{nc}) &\xrightarrow{r_{17}} (I_c) \\
t_{18} : (I_c) &\xrightarrow{r_{18}} (I_{nc})
\end{aligned}
$$

Abstracting this base model involves merging variables, such as S_c and S_{nc} into a composite variable S where $S = S_c + S_{nc}$. In this approach, a composite variable can represent any number of stratified copies of a variable (e.g., S stratified by ten age groups). We also abstract the base model transitions so that their source and target variables are composite variables. For example, if we define composite variables S, I, and E for the corresponding stratified variables in the base model, then transitions t_1 to t_4 become the composite transition $t_{1:4}$:

$$
t_{1:4} : (I, S) \xrightarrow{r_{1:4}} (I, E)
$$

where $r_{1:4}$ is the composite rate of flow and $r_{1:4} = \sum_{i=1}^{4} r_i$. The composite rate of flow $r_{1:4}$ between the base variables due to the base transitions becomes difficult to track because we no longer separately model each base variable. For example $r_1 = \frac{I_c S_c \beta(-c_{m_0} * \epsilon_{m_0} + 1)}{N}$, depends upon I_c and S_c, and they are no longer variables in the abstract model. We express $r_{1:4}$ in terms of the abstract variables by bounding it. Bounding the rate expressions implies that we must also bound each of the variables. We denote the upper and lower bounds with the *ub* and *lb* superscripts.

For example, we express the rate $r_{1:4}$ with a pair of rates $r_{1:4}^{lb}$ and $r_{1:4}^{ub}$. If we assume that the composite rate preserves the total flow between base model variables, we define bounds on the rate as follows:

$$
\begin{aligned}
S(t+dt) &= S_c(t+dt) + S_{nc}(t+dt)\\
&= S_c(t) - (r_1+r_2)dt + S_{nc}(t) - (r_3+r_4)dt\\
&= S(t) - (r_1+r_2+r_3+r_4)dt\\
&= S(t) - \Bigg(\frac{I_cS_c\beta(1-c_{m_0}\epsilon_{m_0})}{N} + \frac{I_{nc}S_c\beta(1-c_{m_1}\epsilon_{m_1})}{N} +\\
&\qquad \frac{I_{nc}S_{nc}\beta(1-c_{m_2}\epsilon_{m_2})}{N} + \frac{I_cS_{nc}\beta(1-c_{m_3}\epsilon_{m_3})}{N}\Bigg)dt\\
&\leq S(t) - \Bigg(\frac{I_cS_c\beta(1-c_m^{ub}\epsilon_m^{ub})}{N} + \frac{I_{nc}S_c\beta(1-c_m^{ub}\epsilon_m^{ub})}{N} +\\
&\qquad \frac{I_{nc}S_{nc}\beta(1-c_m^{ub}\epsilon_m^{ub})}{N} + \frac{I_cS_{nc}\beta(1-c_m^{ub}\epsilon_m^{ub})}{N}\Bigg)dt\\
&= S(t) - \frac{\beta(1-c_m^{ub}\epsilon_m^{ub})}{N}(I_cS_c + I_{nc}S_c + I_{nc}S_{nc} + I_cS_{nc})dt\\
&= S(t) - \frac{\beta(1-c_m^{ub}\epsilon_m^{ub})}{N}((I_c+I_{nc})(S_c+S_{nc}))dt\\
&= S(t) - \frac{\beta(1-c_m^{ub}\epsilon_m^{ub})}{N}(IS)dt\\
&\leq S^{ub}(t) - \frac{I^{lb}S^{lb}\beta(1-c_m^{ub}\epsilon_m^{ub})}{N}dt\\
&= S^{ub}(t+dt)
\end{aligned}
$$

where the upper bound $S^{ub}(t+dt)$ assumes that the negative rate terms have minimal magnitude (i.e., the upper bound decreases by the least amount). The terms are minimal when they are replaced by the appropriate bounds I^{lb}, S^{lb}, c_m^{ub}, ϵ_m^{ub}. The lower bound $S^{ub}(t+dt)$ uses a similar approach, instead selecting bounds with a maximum magnitude and decreasing the lower bound by greatest amount. Positive rate terms are handled similarly so that they are maximal when used to compute upper bounds and minimal for lower bounds.

The abstract model that de-stratifies the base model defines 12 compartments (lower and upper bound for each variable after defining the composite variables), and 12 transitions (lower and upper bound for each composite transition). While the resulting model has fewer transitions, it has more compartments. However, we would have constructed the same size abstraction for a stratified model with an arbitrary number of levels. For example, if the base model used ten levels instead of two, then it would have 33 compartments and significantly more transitions.

We developed two abstract models from the base model. The first, as described above, de-stratifies the S, E, and I variables. The second, de-stratifies only S and E, allowing I to remain stratified.

Figure 2 illustrates the bounds computed by simulating the base and abstract models with FUNMAN. Each subplot is one compartment variable, and each series is one of the bounds or base model value. For example, the second plot illustrates I, which includes:

- Base model: I_compliant and I_noncompliant,
- De-stratified S, E, and I: I_lb and I_ub,
- De-stratified S, and E: I_compliant_lb, I_compliant_ub,I_noncompliant_lb and I_noncompliant_ub,

The abstractions provide different bounds for each variable. In general, as abstraction increases, the model will provide looser bounds, but with a smaller model. Using abstraction refinement techniques, it is possible to start with an abstract model and only refine the relevant variables. Selectively refining models will trade off multiple abstract model simulations against a single, potentially large model simulation. In cases where

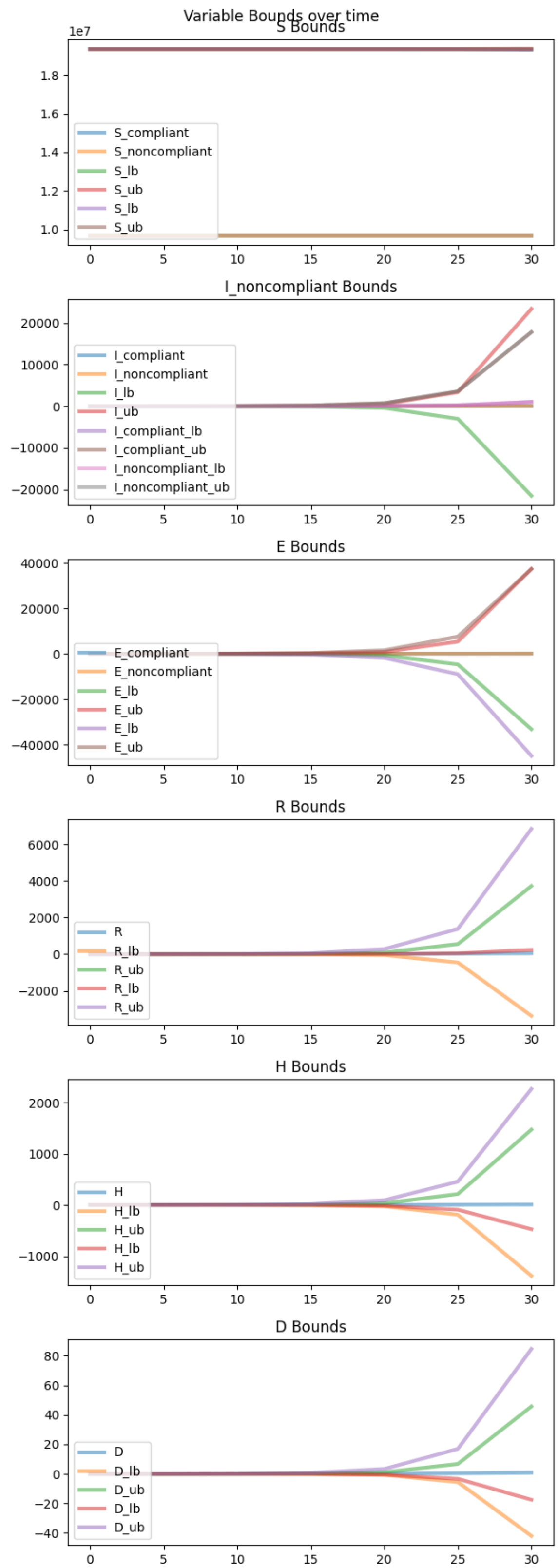

Figure 2: SEIRHD Model Bounds

the bounds are enough to answer a query (or check a constraint), the abstract model simulation can lead to significant scale up.