Any N-state GNA can be simulated on a stateless GNA

Benjamin James Bush

Some assumptions and notation:

We use a slightly different set of notation than that which is used in [Sayama & Laramee, 2009].

All networks and GNAs are assumed to be undirected, weightless and deterministic.

The term “N-state” refers to the state space {0, 2, 3, ..., N-1}.

An N-state network is represented by the triple (V, C, L), where V is a set of nodes, C is a function mapping nodes to states, and L is a set of links, where each link is an unordered pair, such as {node1, node2}.

A stateless network is represented by the pair (V, L), where V and L are defined in the same way as above.

A GNA is represented by a triple (E, R, I), where E is the extraction mechanism, R is the replacement mechanism, and I is the initial configuration.

The extraction mechanism E is a function which takes a graph G as input and returns subgraph of G.

The replacement mechanism R is a function which takes the subgraph G’ as input and returns a pair (G’’, m), where G’’ is the replacement graph and m is the correspondence function which maps the nodes of G’ to nodes of G’’.

The temporal dynamics of a GNA is can be represented by a sequence of graphs $\{G_0, G_1, G_2, ...\}$ where G_0 is the initial configuration of the GNA and G_n is the configuration of the GNA after n iterations.

Given a configuration graph G_n, an extracted subgraph G_n', a replacement graph G_n'', and a correspondence function m, the next configuration G_{n+1} is calculated via the insertion function Ins, so that

$$Ins(G_n, G_n', (G_n'', m_n)) = G_{n+1}$$

The details of how the Ins function works are “fairly obvious” [Sayama and Laramee, 2009], so we will not discuss it here.

The iteration of a GNA A is carried out through the use of the iteration function Itr_A, so that

$$Itr_A(G_n) = Ins(G_n, E(G_n), R(E(G_n)) = G_{n+1}$$

for all n in {0, 1, 2, ...}

and

$$Itr_A^k(G_n) = G_{n+k}$$

for all n in {0, 1, 2, ...} and k in {1, 2, 3, ...}

Claim:

For any **N-state** GNA A there is a **stateless** GNA B which simulates A, so that there exists a function f such that

$$Itr_A^k(G_{A0}) = f^{-1}(\ Itr_B^k(\ f(G_{A0})\ )$$
$$\text{for all k in } \{1, 2, 3, \ldots\}$$

Our first task is to find a way to represent an N-state network as a stateless network. We will be using Bush networks, as illustrated in figure 2 below. Before giving the details of this representation, we define Bush networks.

Definition: Bush network

A stateless network G = (V, L) is a **Bush network** if it meets several conditions, which give here. Firstly, every connected component of G must contain at least one node of degree ≥ 3. Secondly, for every node p in V of degree ≥ 3 we can partition the neighbors of p into the following mutually exclusive subsets: H_p, T_p, D_p, ("**H**ead", "**T**ail", and "**D**istal" respectively). These subsets have the following properties:

- $H_p = \{ h_{p1}, h_{p2} \}$, where h_{p1} and h_{p2} each has degree 1 (i.e. they have no neighbors other than p).
- $T_p = \{ t_{p0} \}$, and t_{p0} has degree 1 or 2. Moreover, all paths which begin at t_{p0} but do not pass through p must pass only through nodes of degree 1 or 2.
- D_p may be empty or nonepty. If D_p is nonepty, then every member of D_p has degree ≥ 4.

Definition: Primary node, BushState, and BushDegree.

Each node p in a Bush network that has degree ≥ 3 is called a **primary node** of the Bush network**.** For each primary node p, **BushState(p)** is defined as one less than the length of the largest path that begins at p that passes only through nodes of degree 1 or 2. See Fig. 1.

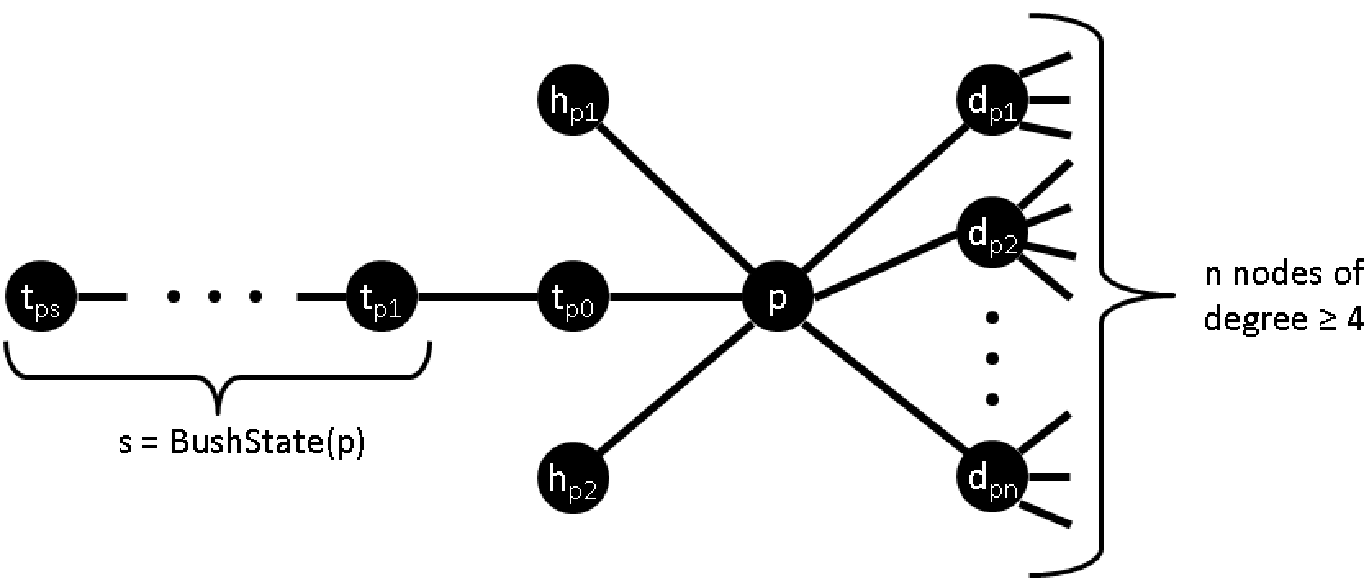

Figure 1: Primary node p, its neighbors, and its state tail.

We will now define the function **Bushify**, which takes an N-state network and maps it to its corresponding Bush network.

Definition: Bushify function

Given an N-state network G = (V_G, C, L_G), we construct Bushify(G) = B = (V_B, L_B) as follows:

For each node p in V_G, we put nodes p, h_{p1}, and h_{p2} into V_B.
We also put t_{pi} into t_p for each i in {0, 1, 2, ..., C(p) }.
Every link { a, b } in L_G is also placed into L_B. In addition, we create the following links for every node p in V_G:

- {p, h_{p1}}, {p, h_{p2}}, and {p, t_{p0}}
- { t_{pi} , $t_{p(i+1)}$ } for all i in {0, 1, ..., C(p)-1}. Note that if C(p) = 0, then the set {0, 1, ..., C(p)-1} is empty.

An example is shown in the figure shown below:

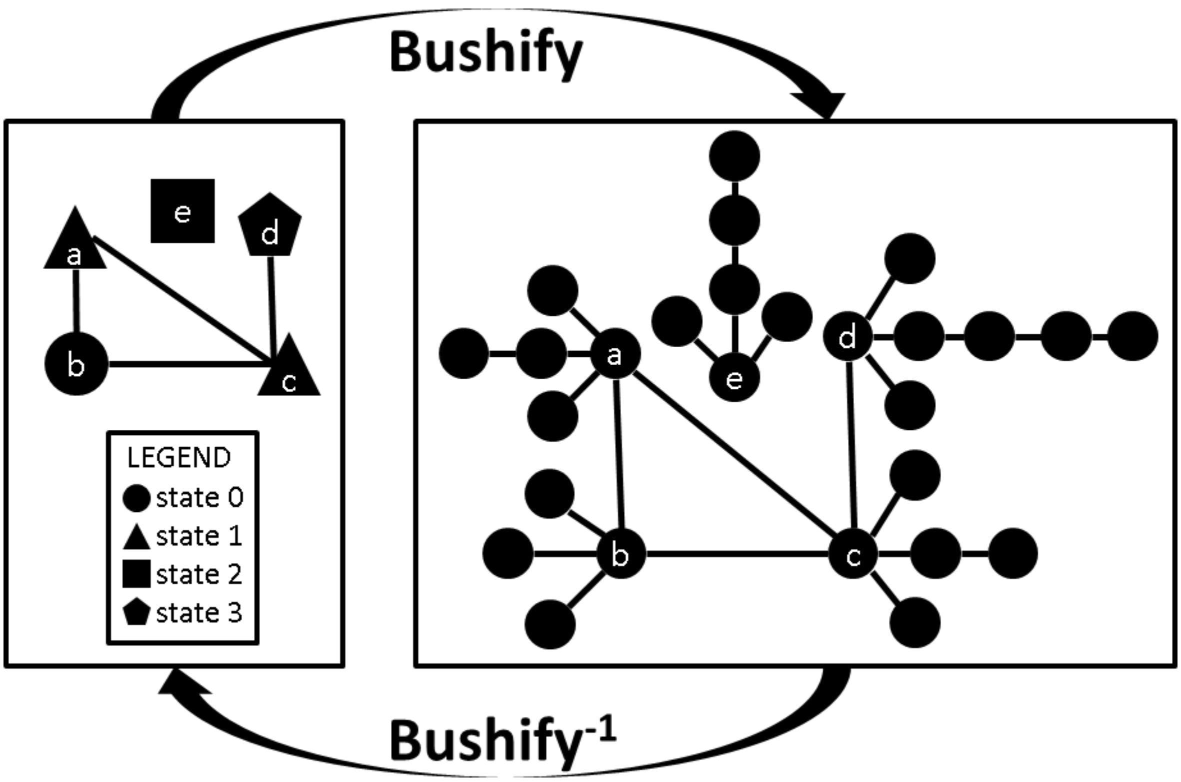

Figure 2: Representing a 3 state network as a stateless Bush network by using the Bushify function.

As the above figure suggests, the Bushify function is invertible, as we now show:
Given any Bush network B = (V_B, L_B), $Bushify^{-1}$(B) = G = (V_G, C, L_G) is an N-state network, where $N = \max_{p \epsilon V_B} BushState(p)$. V_G = {p in V_B s.t. p has degree ≥ 3} so that V_G is the set of primary nodes in B. For any node p in V_G, C(p) = BushState(p). For any two nodes p and q in V_G, { p, q } is in L_G iff { p, q } is in L_B.

Simulating an N-state GNA on a stateless GNA

Suppose we have an N-state GNA A with extraction mechanism E_A, replacement mechanism R_A and initial configuration $I_A = (V_A, C_A, L_A)$. We will define a new stateless GNA B with extraction mechanism E_B, replacement mechanism R_B, and initial configuration I_B. GNA B will simulate GNA A.

The initial configuration for B is I_B = Bushify(I_A)

The extraction mechanism for B is simply

$$E_B = \text{Bushify} \circ E_A \circ \text{Bushify}^{-1}$$

where ∘ is the function composition operator.

The extraction mechanism E_B is illustrated by the **blue path** in figures 3 and 5 below.

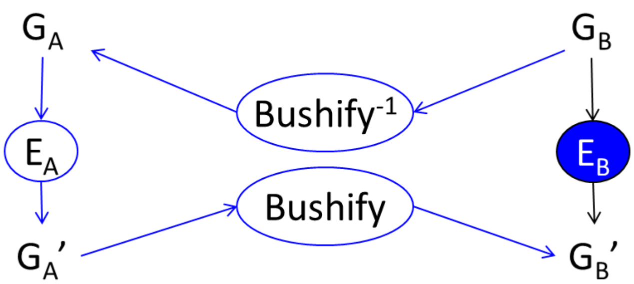

Figure 3 Functional diagram for the new extraction mechanism E_B

The replacement mechanism for B works as follows:

Suppose we have an extracted Bush subgraph G_B'.

Suppose further that $R_A(\text{Bushify}^{-1}(G_B')) = R_A(G_A') = (G_A'', m)$

Then $R_B(G_B') = (\text{Bushify}(G_A''), m) = (G_B'', m)$

Note that the correspondence map m remains unchanged. Only correspondence information about the primary nodes of the Bush network is required, and this information is already in m.

The replacement mechanism R_B is illustrated by the **red paths** in figures 4 and 5 below.

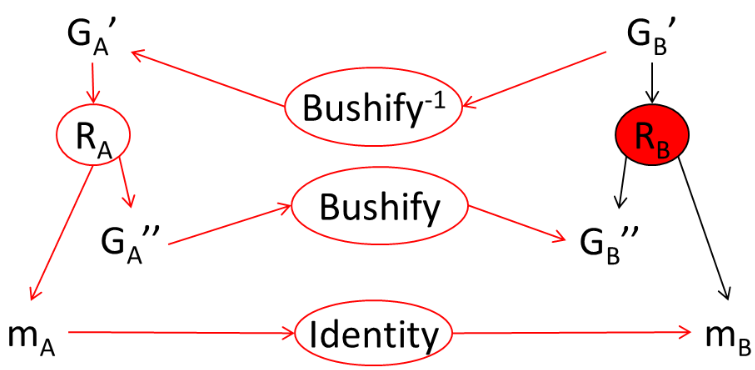

Figure 4 Functional diagram for the new replacement mechanism R_B

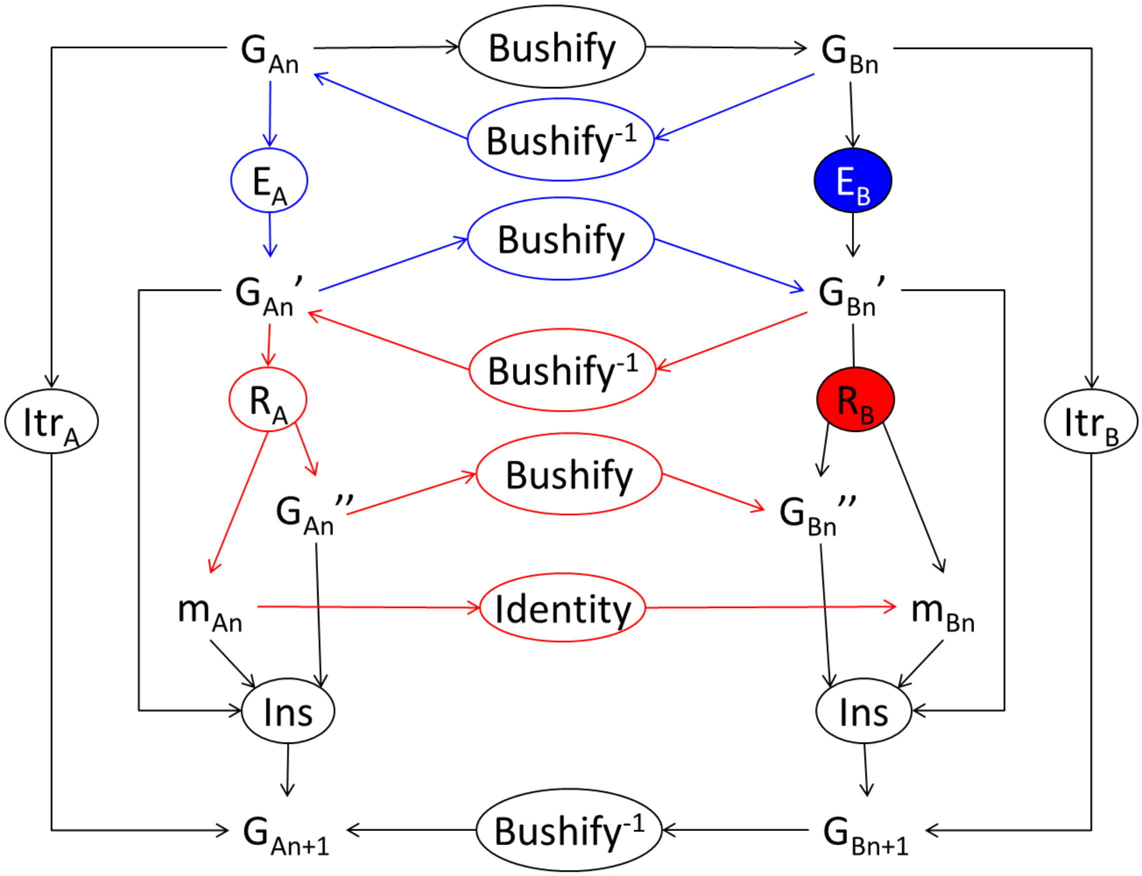

Figure 5: Functional diagram showing the simulation of GNA A by GNA B.

Suppose Itr_A is the iteration function for GNA A and Itr B is the iteration function for GNA B. From the above diagram it is clear that Itr_A is in fact the same function as the composition

$$Bushify^{-1} \circ Itr_B \circ Bushify$$

So that B simulates A as desired. ∎