Network Dynamical Process: SDS III

15.1 Definitions

1. A **Sequential Dynamical System (SDS)** is a mathematical object used extensively in modeling problems in computation. Denoted by S, it is a discrete system defined as a graph over three components: (i) $G = (V, E)$, the underlying graph over a set of nodes V and a set of edges E, where $|V| = n$ and $|E| = m$; (ii) $F = (f_v)$, a set of local functions defined for each node v in V, $f : D^{\delta_v+1} \rightarrow D$, where domain $D = \{0, 1\}$ denotes state of each vertex. $N(v)$ denotes the set of neighbors of v, $d_v = |N(v)|$ denotes the number of neighbors of v (degree of node v); and, π specifies the order in which the node states are to be updated by applying the appropriate function in F. Each node v computes its next state by applying the function f_v on the current states of itself and its neighbors.

2. A **configuration** C of an SDS S is a vector $(s_1, s_2, ..., s_n)$ of length n, where s_i in D denotes the state of each node i in V.

3. The **update function** H denotes a single update of the system configuration by applying the node functions in an order defined in π. $H : D^n \xrightarrow{\pi} D^n$.

4. Phase space PS of S is a directed graph $G_1 = (V, E)$, where each node denotes a configuration X of S and an edge denotes a directed transition between configurations X and $Y = H(X)$.

5. If $Y = H(X)$, then X is called the **predecessor** of Y.

6. If $X = H(X)$, then X is called a **fixed point**.

7. A **Garden of Eden** configuration is one that has no predecessors.

8. A **threshold-SDS** is one in which all the local functions are simple Boolean threshold functions.

9. S is said to **cycle** through a sequence of configurations $\langle C_1, C_2, ..., C_r\rangle$ if $H(C_1) = C_2$, $H(C_1) = C_2$, $H(C_2) = C_3$, ..., $H(C_{r-1}) = C_r$, and $H(C_r) = C_1$.

10. A cycle involving more than one configuration is called a **limit cycle**.

15.2 Computational Theory for SDS: Research Questions

15.2.1 Structure and computation of phase space properties

(a) Computational complexity and provable algorithms.
(b) Structure theorems.
(c) Unified proof techniques.

15.2.2 Computational problems

(a) The reachability problem
Given system states x and y of an SDS-map $\phi = [F_Y, \pi]$, does there exist an integer $r > 0$ such that $\phi^r(x) = y$? In other words, starting from state x, can we reach state y? The worst-case scenario is when all states of the map ϕ are on a simple path, and x and y are the extreme points, implying a worst-case analysis of $2^k - 1$ combinations of all k node in ϕ. The reachability problem is PSPACE-complete.

(b) The permutation-existence problem
In this situation we are given states x and y, a graph Y, and vertex functions $(f_v)_v$. Does there exist a permutation (i.e., update order) p such that $[FY, p]$ maps x to y in one step? That is, does there exist an SDS update order p such that $[FY, p](x) = y$? The answer to this problem is similar to the answer for the reachability problem. For SDS with Boolean threshold vertex functions, the problem is NP-complete, but for nor vertex functions it can be answered efficiently. Note that the reachability problem can be posed for many other types of dynamical systems than SDS, but the permutation existence problem is unique to SDS.

(c) The predecessor-existence problem
Given a system state x and an SDS-map $\phi = [F_Y, \pi]$, does there exist a system state z such that $\phi(z) = x$? The predecessor existence problem is NP-complete in the general case, but can be solved efficiently for restricted classes of vertex functions and/or graphs. Examples include SDS where the vertex functions are given by logical **And** functions and SDS where the graphs have bounded tree-width. Locating the combined function/graph complexity boundary for when such a problem goes from being polynomially solvable to NP-complete is an interesting research question.

15.2.3 Structure theorems

Example: Structure of nor graphs- have only 1 step transients.

15.2.4 Formal specifications

Formal modeling of real world systems and verification of computational codes- does the code faithfully represent the model, efficient mapping of system models on distributed systems.

15.3 Examples of models and computational power of SDS

15.3.1 As universal computing devices

(a) Hopfield networks.
(b) Cellular Automata.
(c) Communicating FSMs.

15.3.2

Can simulate Turing machines for all "natural" complexity classes, i.e., SDS can mimic Turing machines.

15.3.3

Are simulations optional: No, often no shorter computation possible.

15.3.4

Simple interactions with simple local functions are intractable (Global algorithms are intractable, e.g.: reachability problems).

15.4 Polynomial time algorithm for Boolean+threshold SDS

Theorem 1 *Reachability Problem (RP) for SDS is in P when each local function is*
(a) Boolean symmetric and monotone (threshold),
(b) individual nodes need not have same local function, and
(c) ordering should be fair.

Proof:
Let
$T_1(v) \equiv$ threshold value required for node v to become 1, i.e., the smallest integer such that s_v must be assigned 1 if $T_1(v)$ of the inputs to f_v have value 1.
$T_0(v) \equiv$ threshold value required for node v to become 0, i.e., the smallest integer such that s_v must be assigned 0 if $T_0(v)$ of the inputs to f_v have value 0.
The potential $P(C, v)$ of a node v with respect to configuration C is defined as follows:

$$\begin{aligned} P(C, v) &= T_1(v), if C(v) = 1 \\ &= T_0(v), if C(v) = 0 \end{aligned}$$

The potential $P(C, e)$ of an edge $e = \{u, v\}$ with respect to configuration C is defined as follows:

$$\begin{aligned} P(C, e) &= 1, if e = \{u, v\} and C(u) \neq C(v) \\ &= 0, otherwise \end{aligned}$$

For a configuration C, the potential of S is defined by

$$P(C, S) = \sum_{v \in V} P(C, v) + \sum_{e \in E} P(C, e)$$

We first need to consider the following bounds on the potential of S:

Lemma 1 *For any configuration C, $P(C, S) \geq 1$.*

Lemma 2 *For any configuration C, $P(C, S) \leq 3m + 2n$.*

It is subsequently shown that number of steps is of the order $O(m + n) \Rightarrow$ modified RP is in P. ■

15.5 References

[1] C. Barrett, H. Hunt III, M. Marathe, S. Ravi, D. Rosenkrantz, R. Stearns, Analysis problems for sequential dynamical systems and communicating state machines, in: Proc. International Symposium on Mathematical Foundations of Computer Science, MFCS 01, Czech Republic, 2001, pp. 159172.

[2] Henning S. Mortveit , Christian M. Reidys, An Introduction to Sequential Dynamical Systems, Springer-Verlag New York, Inc., Secaucus, NJ, 2007

[3] Christopher L. Barrett , Harry B. Hunt, III , Madhav V. Marathe , S. S. Ravi , Daniel J. Rosenkrantz , Richard E. Stearns, Complexity of reachability problems for finite discrete dynamical systems, Journal of Computer and System Sciences, v.72 n.8, p.1317-1345, December, 2006