Session 3

EE 382V: Social Computing **Fall 2018**

Lecture 8 Session 3: November 10

Lecturer: Vijay Garg *Scribe: Haleigh Walker*

8.1 RSA Underlying Mathematics

Proof: We want to demonstrate the correctness of our RSA algorithm and Euler's Theorem for any integer M which is relatively prime to n.

$M^{\Phi(n)} \equiv 1 \ mod \ n$

Here $\Phi(n)$ is the Euler totient function giving numbers of positive integers less than n which are relatively prime to n. For prime numbers p:

$\Phi(p)$ = p - 1

Therefore:

$\Phi(n) = \Phi(p) \cdot \Phi(q)$
= (p - 1) $\cdot$ (q - 1)
= n - $(p + q)$ + 1

Since d is relatively prime to $\Phi(n)$, its inverse e is in the ring of integers modulo $\Phi(n)$:

e $\cdot$ d $\equiv 1 \ mod \ \Phi(n)$

We now want to prove that our equations for encryption and decryption hold true when e and d are chosen correctly.

D(E(M)) $\equiv (E(M))^d \equiv (M^e)^d \bmod n = M^{ed} \bmod n$
$E(D(M)) \equiv (D(M))^e \equiv (M^d)^e \bmod n = M^{ed} \bmod n$
$M^{ed} \equiv$M$^{k\Phi(n)+1}$ mod n (for some integer k)

For all M such that p does not divide M

M$^{p-1} \equiv 1 \bmod p$

We also know that:

$\Phi(n) = (p-1) \cdot (q-1)$
$k\Phi(n) = k(p-1) \cdot (q-1)$

Therefore:

$M^{k\Phi(n)+1} = (M^{p-1})^{k(q-1)} \cdot M = 1 \cdot M$

Since (p - 1) divides $\Phi(n)$, from here we can conclude:

$M^{k\Phi(n)+1} \equiv M \ mod \ p$

Similarly
$M^{k\Phi(n)+1} \equiv M \bmod q$

Since $n = pq$, it is therefore implied that for all M
M^{ed} ≡M$^{k\Phi(n)+1} \equiv M \bmod n$

Therefore E and D are inverse permutations. ■

8.2 Finite Distributive Lattice (FDL)

The lattice is used to model the search space of the combinatorial optimization problem. We want a parallel algorithm to solve problems with n processes. The goal of the combinatorial optimization problem is to find the smallest element in the lattice which satisfies the predicate (feasible solutions). We will assume the bollean predicate B is lattice linear.

```
Algorithm: getLeastFeasible
    if predicate is true
        you are done
    else
        there is at least one place you must advance to make the predicate true (at least 1 forbidden)
```

We will model the current state of the system with vector G, the global state, where G[i], the state of the process, is maintained at process i.

Definition 8.1 *Forbidden State: Given a predicate B, and a vector G ∈ L, a state G[i] is forbidden if for any vector H ∈ L, where G ≤ H, if H[i] equals G[i], then B is false for H.*

forbidden(G,i,B) $\forall H \in L$: $G \leq H$: *(G[i] = H[i])*$\Longrightarrow \neg B(H)$.

Definition 8.2 *Lattice Linear Predicates: We define a predicate B to be lattice-linear with respect to a lattice L if for any global state G, B is false in G implies that G contains a forbidden state. A boolean predicate B Is lattice-linear with respect to a lattice L iff*

$\forall G \in L : \neg B(G) \Longrightarrow \exists\ i : forbidden(G, i, B)$

Theorem 8.3 *B_1 is lattice linear $\bigwedge$ B_2 is lattice linear $\Longrightarrow$ $B_1 \bigwedge B_2$ is lattice linear*

Proof: Global state G where B_1 or B_2 is false (If B_2 can never be true, then $B_1 \bigwedge B_2$ can never be true)

We need to show ¬B(G) $\Longrightarrow \exists i$: forbidden(G, i, B)

Suppose ¬(B$_1$ $\bigwedge B_2$). This implies that one of the predicates is false and therefore from lattice-linearity, a forbidden state exists. ■

Ex: Smallest possible cut = optimal answer

The predicate is true in the blue circles
If the lattice is linear, you can advance to find the optimal solution
We need to make our cut so that we have P1, P2, and P3 true. See Figure 3.1 below.

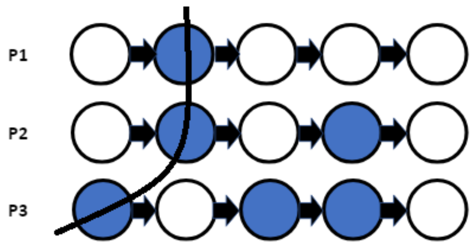

Figure 8.1: Optimal Solution for Lattice Linear Example

8.2.1 Stable Marriage

We now illustrate that the lattice-linear predicate detection technique is applicable when the search space is any distributive lattice by applying predicate detection to the stable marriage problem.

The vertices are a set of proposals E that can be made by men to women. We refer to these proposals as events which are executed by n processes P1, P2 P_n corresponding to n men. When there are no constraints, we impose a partial order $\rightarrow p$ on E with n disjoint chains to model the order in which these proposals can be made. Similar to the Gale-Shapely algorithm, each process makes proposals to women in decreasing order of male preferences. In Figure 3.2, black arrows indicate male preference in decreasing order, while red arrows indicate female preferences in increasing order.

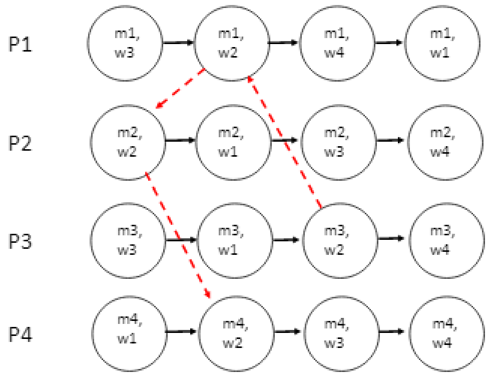

Figure 8.2: Male and Female Preferences

w2 preferences in order: m4, m2, m1, m3

Feasible Predicate: Assignment that gives a stable marriage.

E: set of proposals that can be made by men to women
$\rightarrow p$: partial order imposed

Definition 8.4 *A global state $G \subseteq E$ is consistent if*
$\forall e, f \in E : (e \rightarrow p\ f) \bigwedge (f \in G) \implies (e \in G)$.

Looking for male optimal solution that satisfies the feasible solution.

G = Global State
$B_{stablemarriage} \equiv B_{sin}(G)$
$\equiv$ frontier of consistent global state G
$\equiv$ last state in G
$\equiv$ G satisfies stable marriage predicate iff there is no red arrow pointing from frontier of G to any state in G AND G is consistent

Note: Red arrows pointing backwards from G indicate a blocking pair. A red arrow pointing within G also violates matching because it indicates the same woman for two men.

8.2.2 Prove Lattice Linear Predicate

Claim 8.5 *Backwards red arrows are forbidden.*

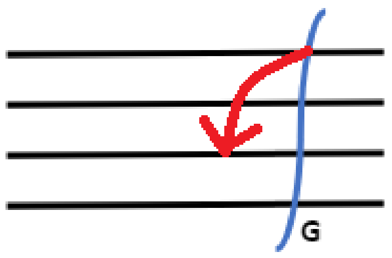

Figure 8.3: Backwards Red Arrows

The predicate can only be true if you advance on this process. You must advance at forbidden states to find the optimal solution.

This algorithm is more parallel than Gale-Shapely because multiple forbidden states can advance at the same time.

8.2.3 Constrained Stable Marriage

For the constrained stable marriage problem, we have arrows that relate proposals of different processes. In the diagram below, the black arrows indicate the order of proposal. If the process proposes to a circle an arrow is pointing to, it must also propose to the circle from which the arrow originated (See Figures 3.4 and 3.5 below).

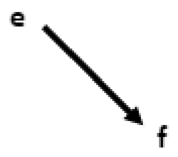

Figure 8.4: If You Include f, You Must Include e

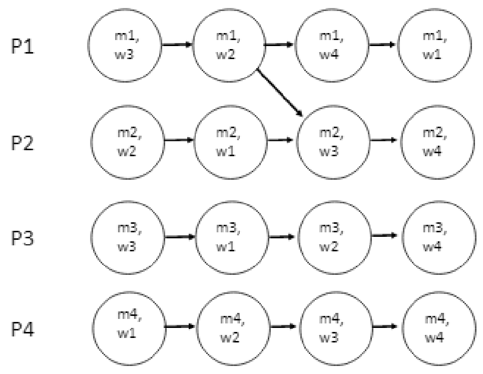

Figure 8.5: Constrained Stable Marriage

Backwards black arrows indicate that G is not consistent (future to past). See Figure 3.6 below.

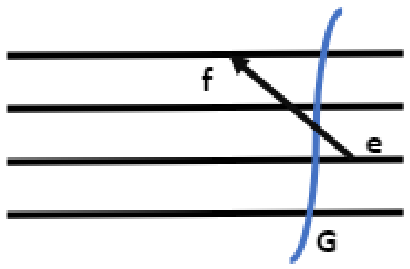

Figure 8.6: G is not Consistent

8.2.4 Algorithm Applications

The same algorithm can be applied to multiple problems, such as Market Clearing Price (VCG) and Housing Allocation problems, but the predicate continues to change.

8.2.4.1 Market Clearing Price

For VCG, if you are given a price vector, prices must increase, therefore decreasing prices are forbidden. You can then construct a preferred seller graph (bipartite graph).
Predicate: Market clearing at perfect matching.
We know that without perfect matching, a constricted set exists.
In the minimal set of constricted items, all items are forbidden.

8.2.4.2 Housing Allocation

In the Housing Allocation problem, subset matchings create cycles. Not being part of a cycle is forbidden.