CS 4804 Homework 2
Solution Sketches

---

1. **(10 points)** As the textbook states, the original cryptarithmetic problem can be thought of as having the following constraints:

$$\begin{aligned} F &\neq O \\ F &\neq U \\ &\cdots \\ U &\neq W \\ 2O &= R + 10X_1 \\ X_1 + 2W &= U + 10X_2 \\ X_2 + 2T &= O + 10F \end{aligned}$$

where X_1 and X_2 are variables capturing the carryovers in the addition. Of these constraints only the last three need to be re-expressed as binary constraints. We illustrate the basic idea using the constraint

$$2O = R + 10X_1$$

Introduce a new variable (say V_1) whose domain consists of the cartesian product of the domains of all variables in the given constraint (i.e., O, R, and X_1). The (common) domain of O and R is $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and the domain of X_1 is $(0, 1)$. Thus the domain of V_1 is the set of 200 tuples:

$$\begin{gathered} \{(0,0,0), \\ (0,0,1), \\ (0,1,0), \\ (0,1,1), \\ \cdots \\ (9,8,0), \\ (9,8,1), \\ (9,9,0), \\ (9,9,1)\} \end{gathered}$$

Then the constraint $2O = R + 10X_1$ can really be captured as

$$2(\text{first attribute of } V_1) = \text{second attribute of } V_1 + 10(\text{third attribute of } V_1)$$

In other words, this becomes a unary constraint! Meaning we can actually simplify the domain of V_1 to contain only the ten legal values:

$$\begin{gathered} \{(0,0,0), \\ (1,2,0), \\ (2,4,0), \\ (3,6,0), \\ (4,8,0), \\ (5,0,1), \\ (6,2,1), \\ (7,4,1), \\ (8,6,1), \\ (9,8,1)\} \end{gathered}$$

The only task left is to relate this new variable to the existing variables. We introduce a binary constraint between V_1 and O with the annotation '$O =$ first attribute of V_1'; a binary constraint between V_1 and R with the annotation '$R =$ second attribute of V_1'; and a binary constraint between V_1 and X_1 with the annotation '$X_1 =$ third attribute of V_1'. Similarly, we can model the other two non-binary constraints.

2. **(10 points)** Define the propositional variable $x_{i,j}$ that is true when pigeon i is placed in hole j and false otherwise. Then the pigeonhole problem is to find a satisfying assignment of the CNF:

$$
\begin{array}{lll}
 & (x_{1,1} \vee x_{1,2} \vee \cdots \vee x_{1,n}) & \text{// pigeon 1 is in at least one of the holes} \\
\wedge & (x_{2,1} \vee x_{2,2} \vee \cdots \vee x_{2,n}) & \text{// pigeon 2 is in at least one of the holes} \\
\wedge & \cdots & \\
\wedge & (x_{n+1,1} \vee x_{n+1,2} \vee \cdots \vee x_{n+1,n}) & \text{// pigeon n+1 is in at least one of the holes} \\
\wedge & (\neg x_{1,1} \vee \neg x_{2,1}) & \text{// hole 1 cannot have both pigeons 1 and 2} \\
\wedge & (\neg x_{1,1} \vee \neg x_{3,1}) & \text{// hole 1 cannot have both pigeons 1 and 3} \\
\wedge & \cdots & \\
\wedge & (\neg x_{2,1} \vee \neg x_{3,1}) & \text{// hole 1 cannot have both pigeons 2 and 3} \\
\wedge & (\neg x_{2,1} \vee \neg x_{4,1}) & \text{// hole 1 cannot have both pigeons 2 and 4} \\
\wedge & \cdots & \\
\wedge & (\neg x_{n,1} \vee \neg x_{n+1,1}) & \text{// hole 1 cannot have both pigeons } n \text{ and } n+1 \\
\wedge & (\neg x_{1,2} \vee \neg x_{2,2}) & \text{// hole 2 cannot have both pigeons 1 and 2} \\
\wedge & (\neg x_{1,2} \vee \neg x_{3,2}) & \text{// hole 2 cannot have both pigeons 1 and 3} \\
\wedge & \cdots & \\
\wedge & (\neg x_{n,2} \vee \neg x_{n+1,2}) & \text{// hole 2 cannot have both pigeons } n \text{ and } n+1 \\
\wedge & \cdots & \\
\wedge & (\neg x_{1,n} \vee \neg x_{2,n}) & \text{// hole } n \text{ cannot have both pigeons 1 and 2} \\
\wedge & \cdots & \\
\wedge & (\neg x_{n,n} \vee \neg x_{n+1,n}) & \text{// hole } n \text{ cannot have both pigeons } n \text{ and } n+1
\end{array}
$$

This gives a total of $(n + 1 + \frac{n^2}{2}(n + 1))$ clauses involving $n(n + 1)$ variables. The constraints declaring that a pigeon can be in only one hole are not modeled here as they fall out indirectly from the given constraints. You could have been more pedantic and included $(n + 1)\frac{n(n-1)}{2}$ more clauses. These clauses will have the form:

$$
\begin{array}{lll}
 & (\neg x_{1,1} \vee \neg x_{1,2}) & \text{// pigeon 1 cannot be in both holes 1 and 2} \\
\wedge & (\neg x_{1,1} \vee \neg x_{1,3}) & \text{// pigeon 1 cannot be in both holes 1 and 3} \\
\wedge & \cdots & \\
\wedge & (\neg x_{1,1} \vee \neg x_{1,n}) & \text{// pigeon 1 cannot be in both holes 1 and } n \\
\wedge & \cdots & \\
\wedge & (\neg x_{1,n-1} \vee \neg x_{1,n}) & \text{// pigeon 1 cannot be in both holes } n-1 \text{ and } n \\
\wedge & \cdots & \\
\wedge & (\neg x_{n+1,1} \vee \neg x_{n+1,2}) & \text{// pigeon n+1 cannot be in both holes 1 and 2} \\
\wedge & \cdots & \\
\wedge & (\neg x_{n+1,n-1} \vee \neg x_{n+1,n}) & \text{// pigeon n+1 cannot be in both holes } n-1 \text{ and } n
\end{array}
$$

An example of an **incorrect answer**: Create variables p_i for the pigeons and h_i for the holes (giving a total of $2n + 1$ variables). This is a dubious formulation because the variables are always true! i.e., there are indeed $n + 1$ pigeons and n holes. There is no point creating a

CNF formula when you know for sure the truth assignments of all variables. Any clause that you create after that is going to be true! e.g., $p_1 \vee h_2 \vee p_3$ is true because the two pigeons and the one hole are indeed present. Meaning, your final formula is most likely true (which is another bug, because the pigeonhole problem is not satisfiable).

3. **(20 points)** Let us think of the puzzle from the viewpoint of the five houses. To each house, we need to assign a color, situate a person, associate a cigarette, drink, and pet. Let us first order the houses, colors, people, cigarettes, drinks, and pets in some manner. Define propositional variables c_{ij} that is true when house i has color j and false otherwise. Similarly, p_{ij}, t_{ij}, d_{ij}, and p_{ij} capture the association from house to people, cigarettes, drinks, and pets. Then the portion of the CNF dealing with colors can be given by:

$$
\begin{array}{lll}
 & (c_{11} \vee c_{12} \vee c_{13} \vee c_{14} \vee c_{15}) & \text{// house 1 has at least one color} \\
\wedge & (c_{21} \vee c_{22} \vee c_{23} \vee c_{24} \vee c_{25}) & \text{// house 2 has at least one color} \\
\wedge & (c_{31} \vee c_{32} \vee c_{33} \vee c_{34} \vee c_{35}) & \text{// house 3 has at least one color} \\
\wedge & (c_{41} \vee c_{42} \vee c_{43} \vee c_{44} \vee c_{45}) & \text{// house 4 has at least one color} \\
\wedge & (c_{51} \vee c_{52} \vee c_{53} \vee c_{54} \vee c_{55}) & \text{// house 5 has at least one color} \\
\wedge & (\neg c_{11} \vee \neg c_{12}) & \text{// house 1 cannot have both colors 1 and 2} \\
\wedge & (\neg c_{11} \vee \neg c_{13}) & \text{// house 1 cannot have both colors 1 and 3} \\
\wedge & \cdots & \\
\wedge & (\neg c_{12} \vee \neg c_{13}) & \text{// house 1 cannot have both colors 2 and 3} \\
\wedge & \cdots & \\
\wedge & (\neg c_{54} \vee \neg c_{55}) & \text{// house 5 cannot have both colors 4 and 5}
\end{array}
$$

Similarly, we can add clauses to represent uniqueness of cigarette, drink, people, and pets.

You could also be more adventurous and attempt to encode the specific facts given in the question. For instance, the constraint 'the Ukrainian drinks tea' can be modeled by first restating it in terms of houses, i.e., 'the house the Ukranian lives in is the same as the house in which tea is consumed.' We then take the index of the Ukranian (say she is the third person in our list of people), the index of tea (say it is the fourth drink in our list of drinks), and add the constraint:

$$
\begin{array}{lll}
 & (p_{13} \Leftrightarrow d_{14}) & \text{// house 1 has both Ukranian and tea, or neither} \\
\wedge & (p_{23} \Leftrightarrow d_{24}) & \text{// house 2 has both Ukranian and tea, or neither} \\
\wedge & (p_{33} \Leftrightarrow d_{34}) & \text{// house 3 has both Ukranian and tea, or neither} \\
\wedge & (p_{43} \Leftrightarrow d_{44}) & \text{// house 4 has both Ukranian and tea, or neither} \\
\wedge & (p_{53} \Leftrightarrow d_{54}) & \text{// house 5 has both Ukranian and tea, or neither}
\end{array}
$$

Notice that each $\Leftrightarrow$ can be restated in terms of a conjunction of implications, which in turn can be reformulated as disjunctions; the net effect is that you can restate the entire formula in CNF.

An example of an **incorrect answer**: Create variables n, e, s, u, j for the Norwegian, Englishman, Spaniard, Ukranian, and the Japanese respectively. These are always true!!!! It is given in the problem that these people are present. What you need are variables representing the truth of 'the Norwegian is in the first hourse,' 'the Norwegian lives in the blue house,' etc. which have the potential to be false.

4. **(60 points)** If you coded the three algorithms correctly, you will find that the arc-consistency algorithm generates the fewest number of nodes, followed by the forward propagation algorithm, followed by backtracking. However, depending on your implementation, arc-consistency

might end up spending more time 'thinking' than 'doing' and in this respect, forward propagation will appear as the best of the three. The difficulty of problems is also not monotonous as n increases, and you should notice 'bumps' in the plots of nodes generated versus n. 30 points for coding, 10 points for correct execution, 10 points each for the above observations about (i) nodes generated versus n, for the three algorithms, and (ii) time spent thinking versus doing, for the three algorithms.