1)
- Reflexive: If the diagonal has only ones, the relation is reflexive.
  Justification: If for every x, x R x, so the items on the diagonal must be ones. Similarly, for every one on the diagonal, we have x R x, so we have reflexivity.
  All other justifications follow in a similar manner.
- Irreflexive: If the diagonal has only zeros, the relation is irreflexive.
- Symmetric: If the matrix $A = A^T$, the relation is symetric.
- Anti-symmetric: If outside the diagonal no two symmetric cells are both ones.
- Transitivity is more complicated, we've seen the Warshall algorithm. Alternatives exist with comparing the matrix A to $A \cdot A$.
- Existence: Every row has at least one "one"
- Uniqueness: At most one "one" in every row
- Injective: At most one "one" in every column
- Surjective: At least one "one" in every column
- Being an order relies on transitivity, rest can be checked as above

2)

$$M = (\{s, p, l, t, r\}, \{0, 1\}, \{\vdash, 0, 1\sqcup\}, \vdash, \sqcup, \delta, s, t, r)$$

	⊢	0	1	⊔
s	(s,⊢,R)	(p,1,R)	(s,1,R)	(t,⊔,L)
p		(p,0,R)	(p,1,R)	(l,1,L)
l	(s,⊢,R)	(l,0,L)	(l,1,L)	

The state s finds a first 0 in the string, replaces it with a 1 and changes to state p. The state p goes to the end of the string, adds a 1, and changes to state l. The state l goes to the beginning of the string and changes to the state s. Therefore the machine will replace each zero by a one and add a one at the end, replacing all the zeros by two times more ones.

3) The partitions are:

$$\{\{a, b, c, d\}\}$$

$$\{\{a\}, \{b, c, d\}\}$$

$$\{\{a, c, d\}, \{b\}\}$$

$$\{\{a, b, d\}, \{c\}\}$$

$$\{\{a, b, c\}, \{d\}\}$$

$$\{\{a, b\}, \{c, d\}\}$$

$$\{\{a, c\}, \{b, d\}\}$$

$$\{\{a, d\}, \{b, c\}\}$$

$$\{\{a, b\}, \{c\}, \{d\}\}$$

$$\{\{a, c\}, \{b\}, \{d\}\}$$

$$\{\{a, d\}, \{b\}, \{c\}\}$$

$$\{\{a\}, \{b, c\}, \{d\}\}$$

$$\{\{a\}, \{b, d\}, \{c\}\}$$

$$\{\{a\}, \{b\}, \{c, d\}\}$$

$$\{\{a\}, \{b\}, \{c\}, \{d\}\}\}$$

With the first one being the coarsest and the last one being the finest.

4$*$) Given a DFA with n states $s_1 \ldots s_n$ with s_s being the starting state we transform it to a Turing machine as follows:

- Add a TM starting state s and a TM transition

$$\delta(s, \vdash) = (m_s, \vdash, R)$$

- For each DFA transition $s_i \to c \to s_j$ add a TM transition

$$\delta(m_i, c) = (m_j, c, R)$$

- For each accepting state of the DFA s_i add a TM transition

$$\delta(m_i, \sqcup) = (t, \sqcup, L)$$

- For each rejecting state of the DFA s_i add a TM transition

$$\delta(m_i, \sqcup) = (r, \sqcup, L)$$

It is easy to see that the Turing machine traverses its input only once from left to right without changing it and stops at the end of input. It is an easy induction to show that at every position of the input i the state in which the Turing machine is m_k corresponds to the state at which the DFA is s_k. As such, when the word ends and the the DFA terminate, the machine encounters the blank. If the DFA state was accepting the machine will go to the accepting state and otherwise.