|  |  |
| --- | --- |
| Question | Answer |
| 5.1 | It is possible to rule out the invalid transition scenario through meta-modeling. By defining the relationship between states and transitions within a Machine object, the meta-model enforces a structure that disallows transitions crossing machine boundaries. |
| 5.3 | First-Order Formula for the Meta-Model in Figure 5.1   1. Parent Constraint:   ∀𝑥(Person(𝑥)→count({𝑦:Parent(𝑦,𝑥)})≤2  Verification of the Two Rightmost Instances in Figure 5.2   * The provided instances:   + Alice, Bob, and E cycle: Each has no more than two parents.   + D and C mutual parent-child relationship: Each has no more than two parents.   Both instances adhere to the constraints specified in the first-order formula. |
| 5.4 | ∀*x*,*y*,*s*(Transition(*s*,*x*)∧Transition(*s*,*y*)∧*x !*=  *y*→∃*l*1​,*l*2​(InputLabel(*x*,*l*1​)∧InputLabel(*y*,*l*2​)∧*l*1​=*l*2​)) |
| 5.5 | Base Case (Path Length 1):   * If s1 = sn, the equation trivially holds (direct successor).   Inductive Case (Path Length k+1):   * Assume the equation holds for k-length paths. * If successor\*(s1, sn), there exists a path with intermediate state s2:   + successor(s1, s2)   + successor\*(s2, sn) (path length k) * By induction hypothesis, the equation holds for the sub-path (s2, sn): (s2 = sn) ∨ ∃s3. successor(s2, s3) ∧ successor\*(s3, sn) * Combining: We get the original equation for the k+1 path length.   By induction, successor\*(s1, sn) ≡ true implies the equation for all path lengths, proving it satisfies the definition of the transitive closure. |
| 5.16.a | inv[Person] { \_.getChild.size <=2 }   * Ecore Feature to Use: Upper Bound of a Reference   + Reference: children   + Type: Person   + Containment: True   + Multiplicity: 0..2 (meaning 0 to 2 children are allowed) |
| 5.16.b | invariant: self.name.notEmpty().   * Ecore Feature to Use: EAttribute Properties   + Attribute: name   + Type: EString   + Additional Specification: Use an OCL invariant in the model: name.notEmpty() |