

Abstraction Assignment

Choose 5 problems.

Due April 18

1. Write a procedure that copies a stack according to the following specifications:
Oper Copy_Stack(**updates** S, T: Stack);
ensures S = @T **and** T = @T;
2. Write a procedure to reverse a stack according to:
Oper Flip(**updates** S: Stack);
requires true;
ensures S = @S^{Rev};
3. Write a procedure to check 2 stacks for equality:
Oper Are_Equal_Stacks(**restores** S, T: Stack): Boolean;
ensures Are_Equal_Stacks = (S = T);
4. Suppose you have the One_Way_List component available. Show how to implement the Bounded Queue_Template.
5. Write a procedure to copy a queue:
Oper Replica(**restores** Q: Queue): Queue;
ensures Replica = (Q);
6. Write a procedure to merge 2 queues:
Oper Merge_Int_Q(**clears** P, Q: Int_Q_Fac.Queue;
replaces R: Int_Q_Fac.Queue);
requires In_Ascending_Order(P) **and** In_Ascending_Order(Q) **and**
|P| + |Q| ≤ Max_Length;
ensures R ≅ (@P, @Q) **and** In_Ascending_Order(R) **and** P = Λ **and** Q = Λ;
Note: [(≅ is the "is a shuffle" predicate]
7. Describe at least 3 implementations for the Coalescable_Equivalence_Relation_Template Template.
8. Describe at least 3 implementations for the Sorting Machine.
9. Explain 2 implementations for the Spanning Forest.
10. Using the Coalescable_Equivalence_Relation and the Sorting_Machine, implement the Spanning_Forest.