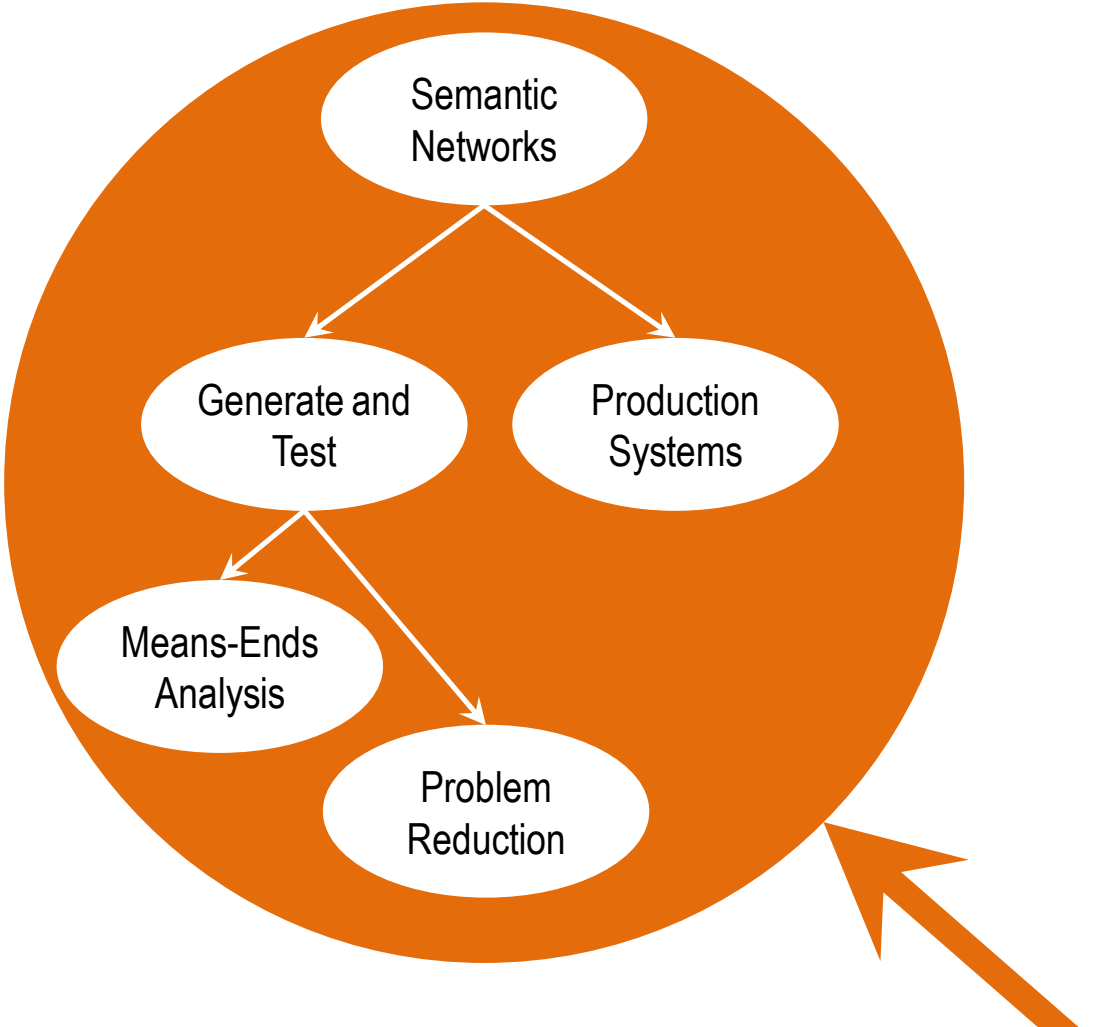
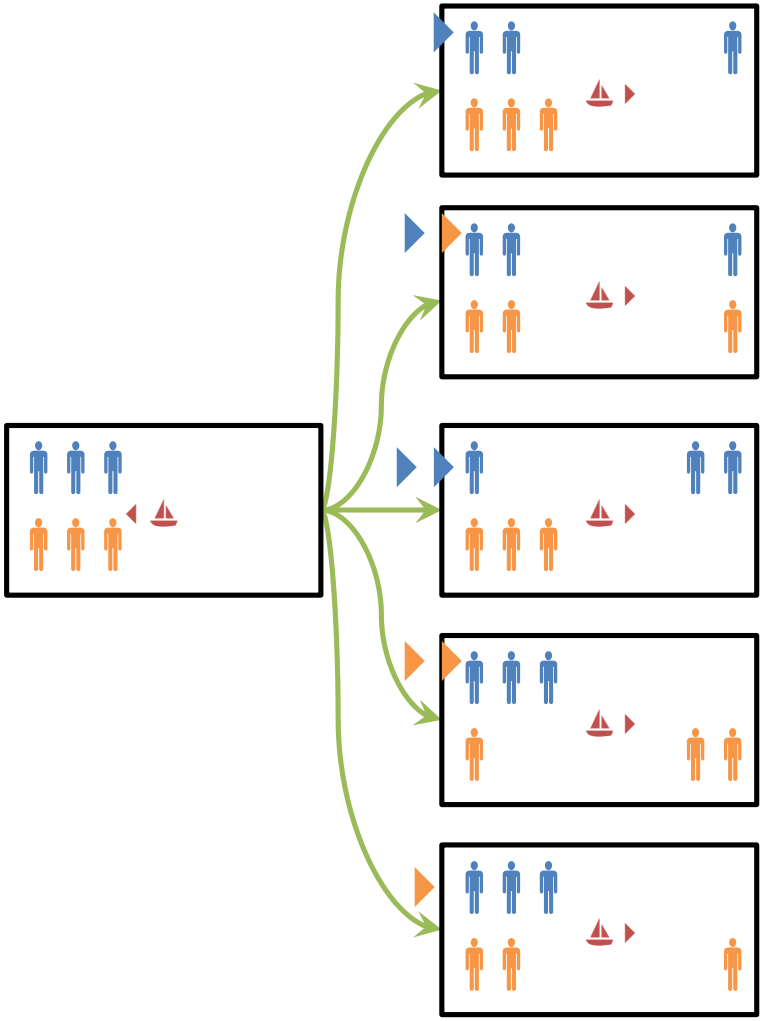

Generate
and Test

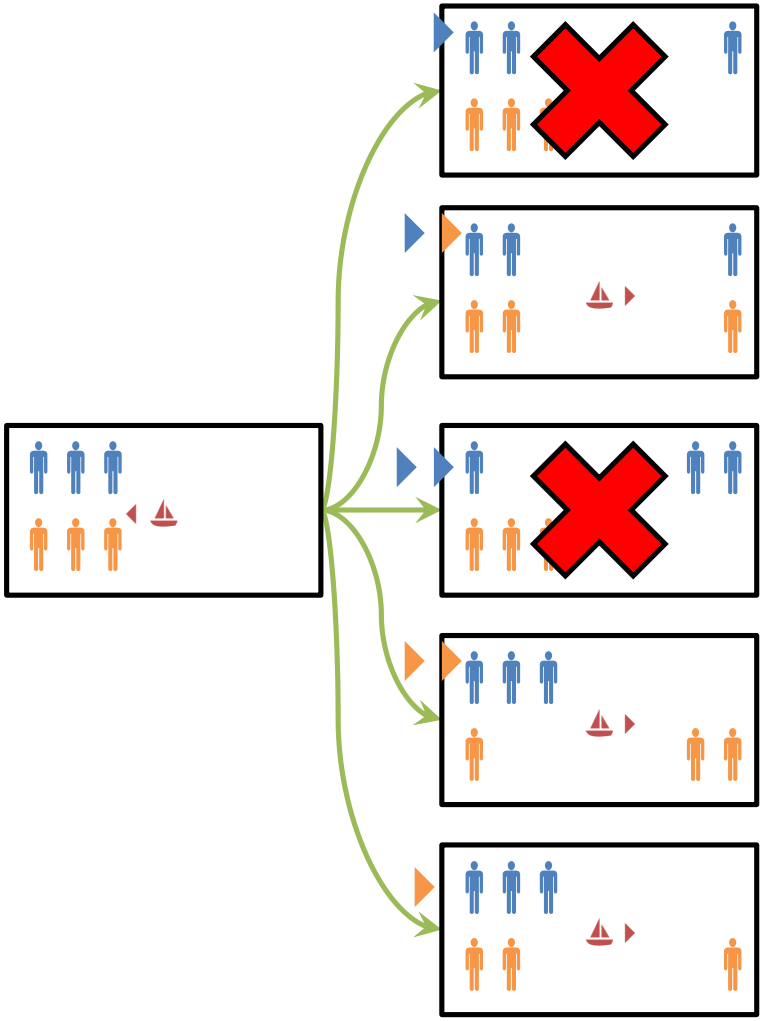
# Fundamentals

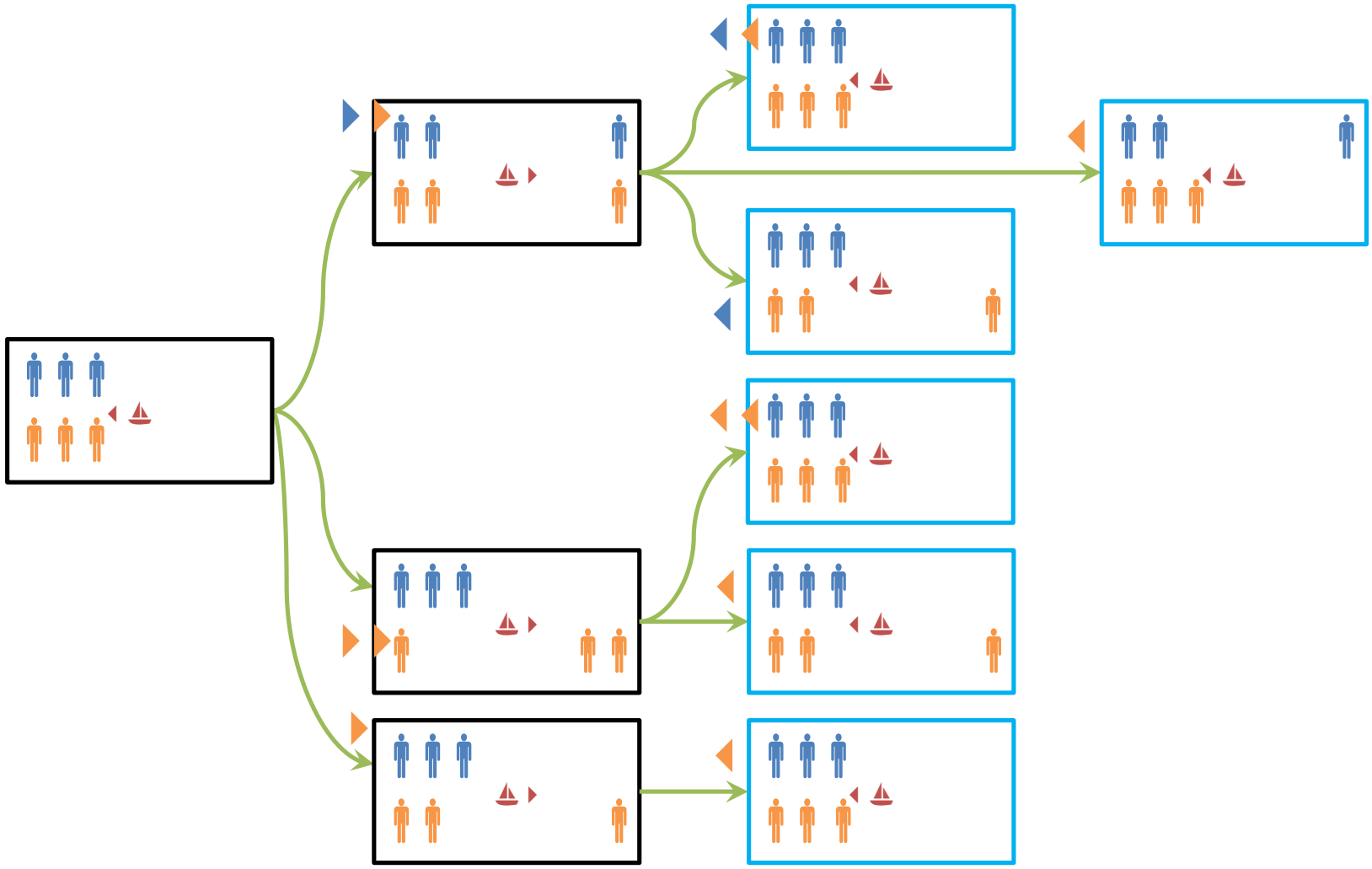


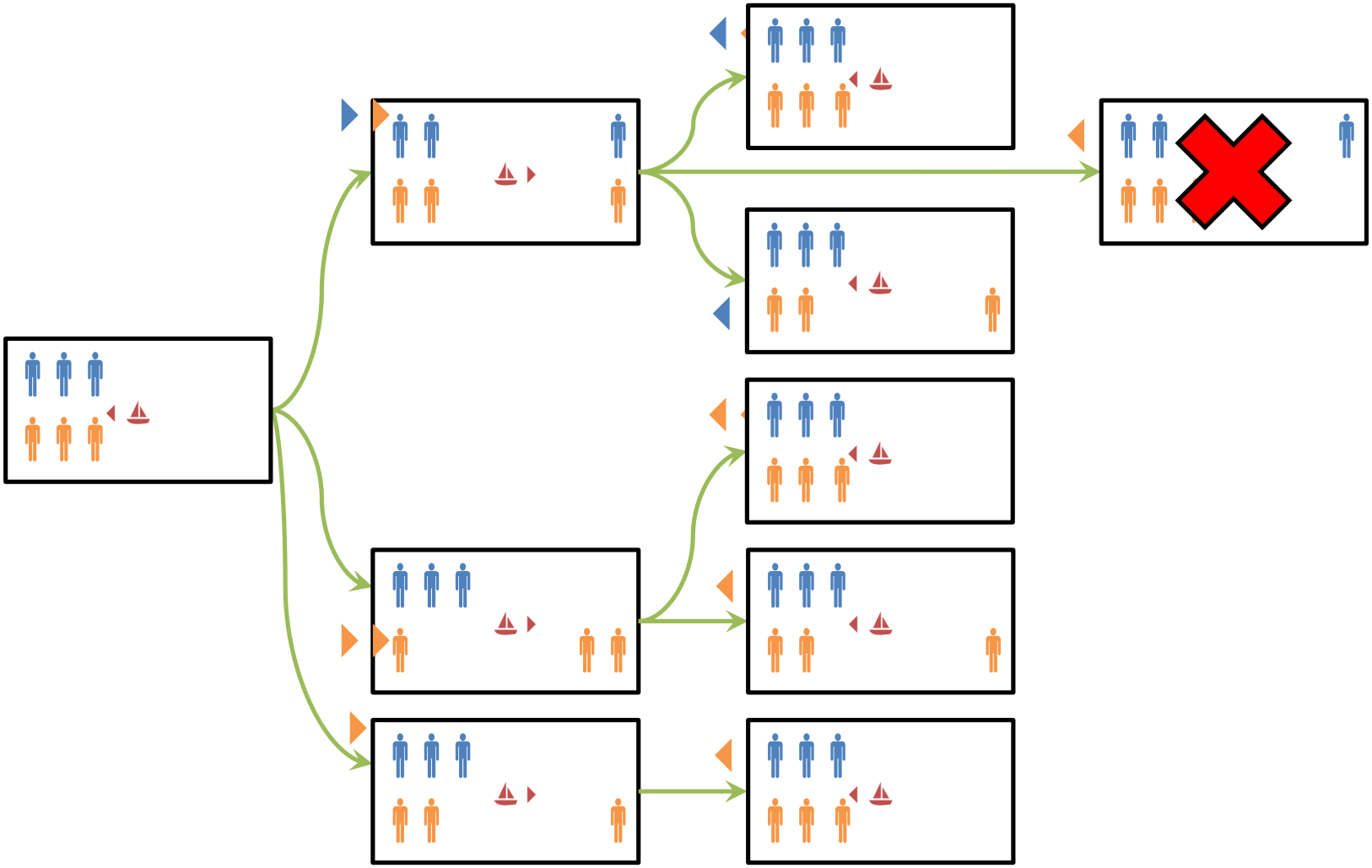
# Lesson Preview

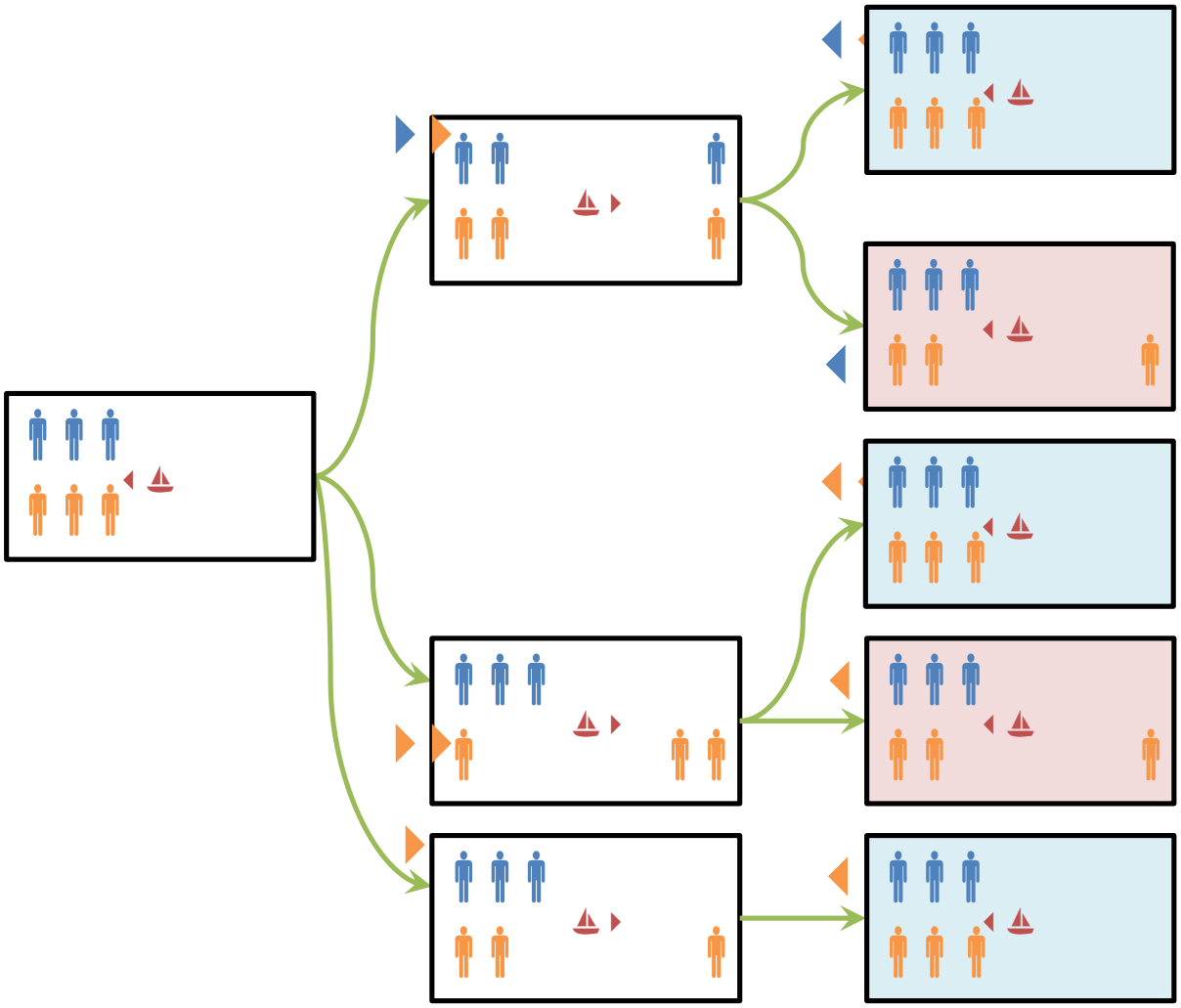
- Generate and test method
- Smart testers
- Smart generators
- Generate and test for Raven's Progressive Matrices

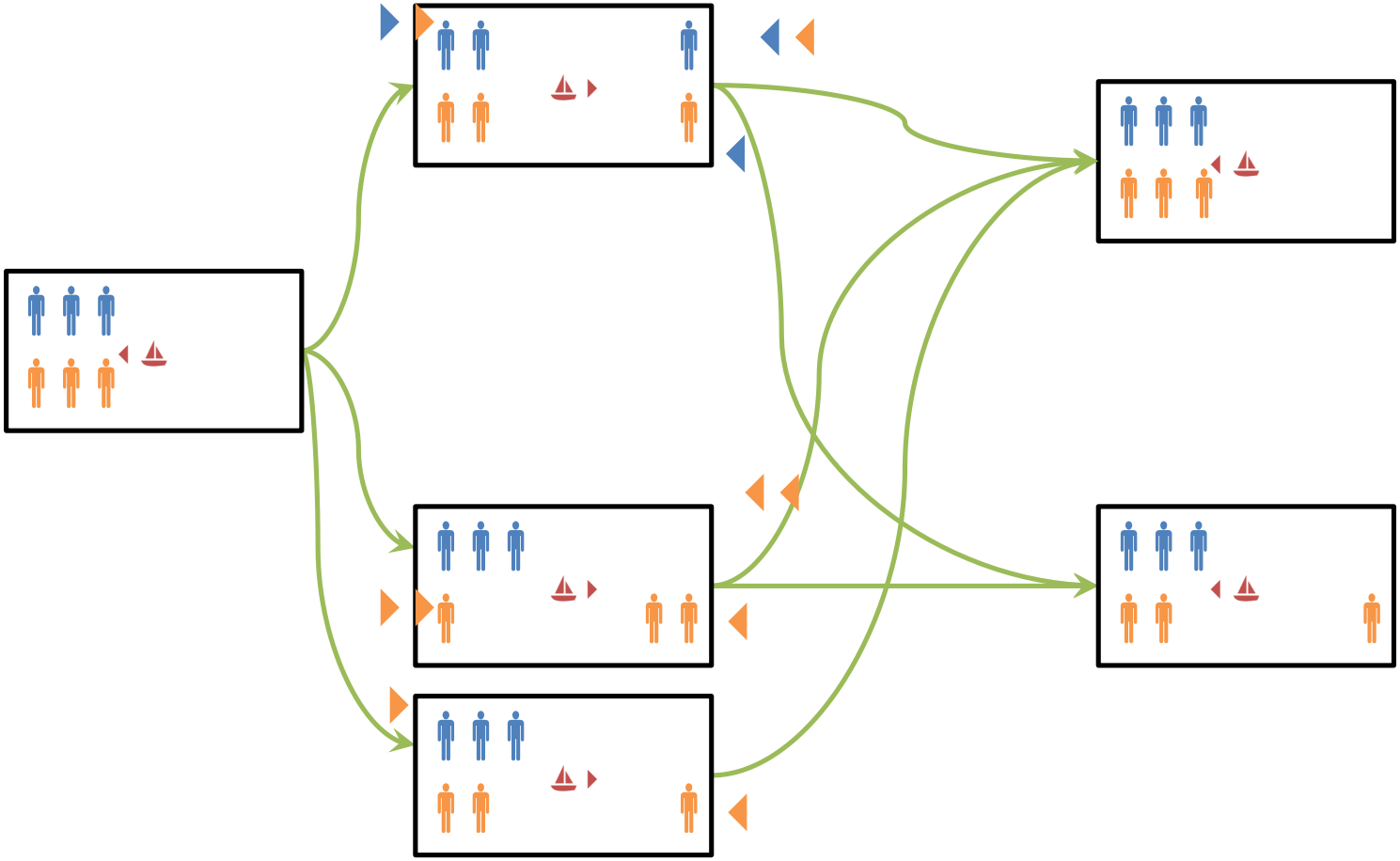


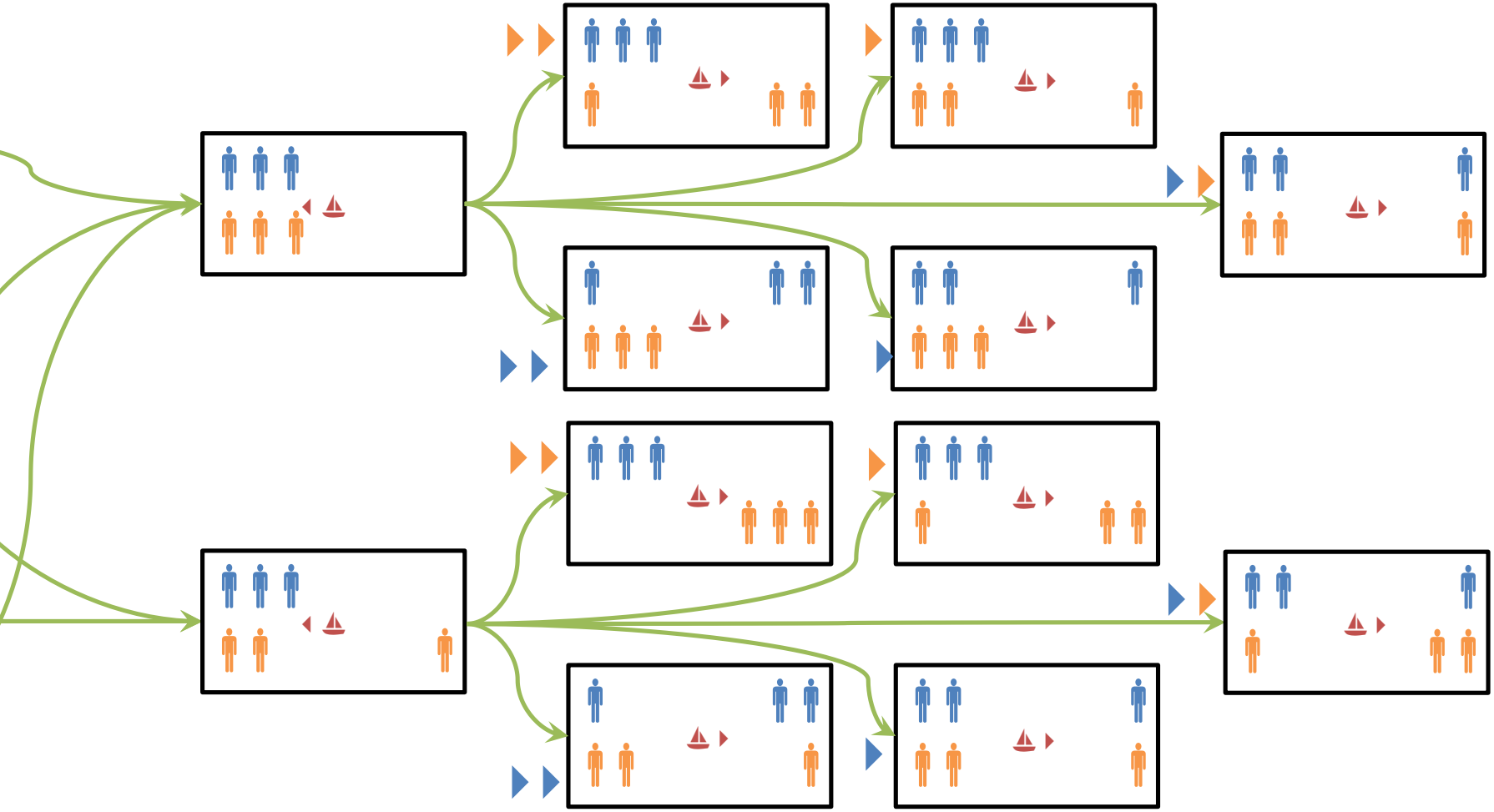


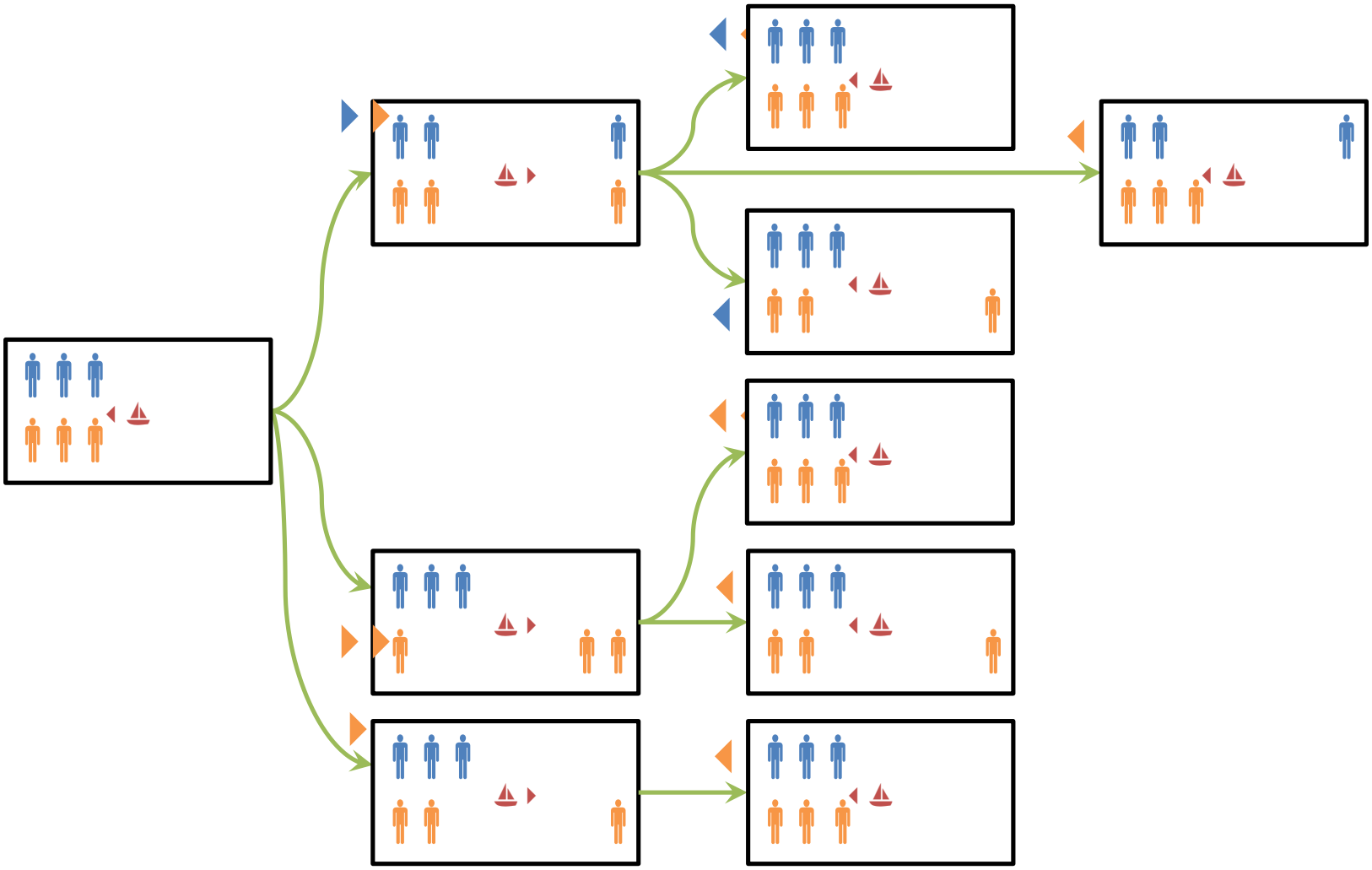


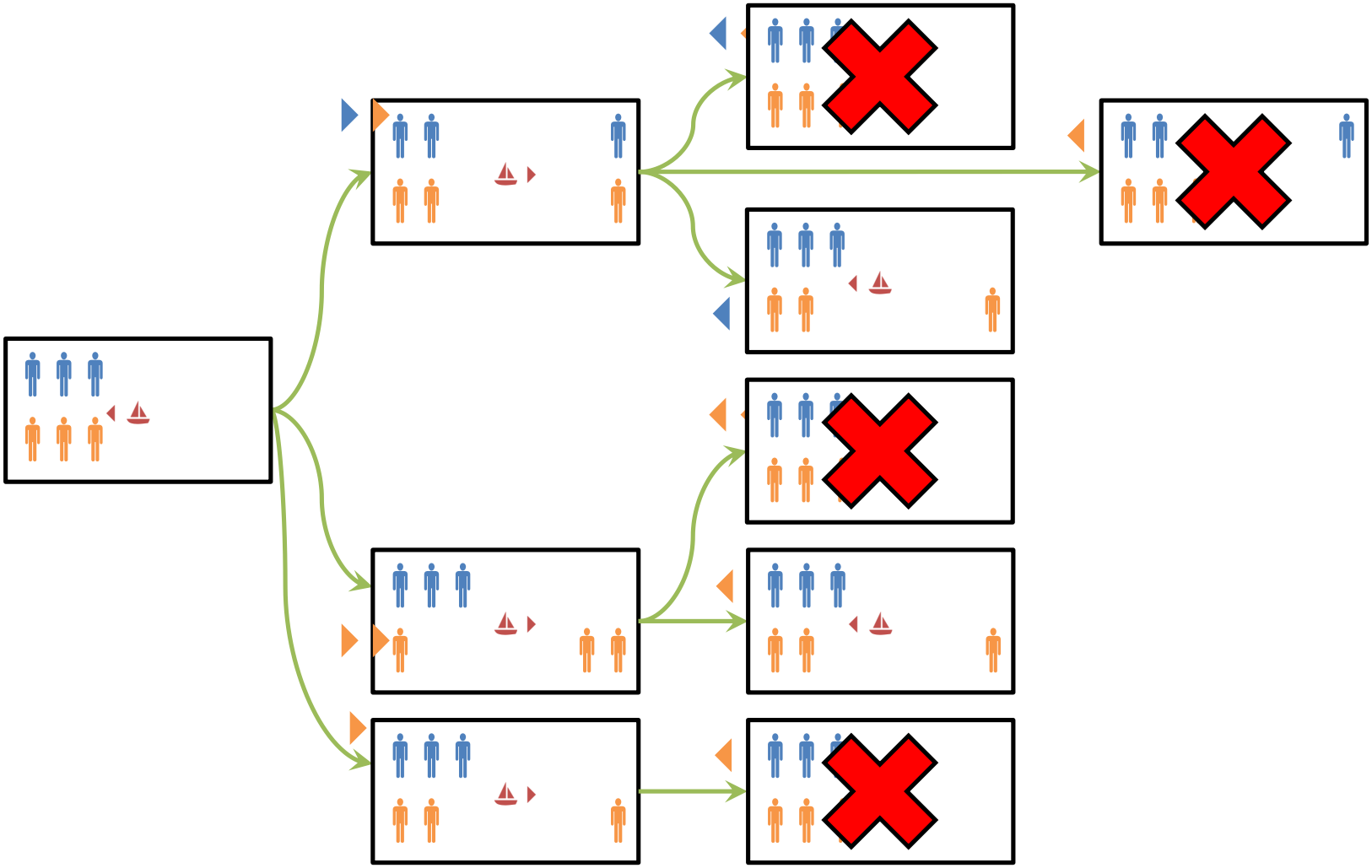


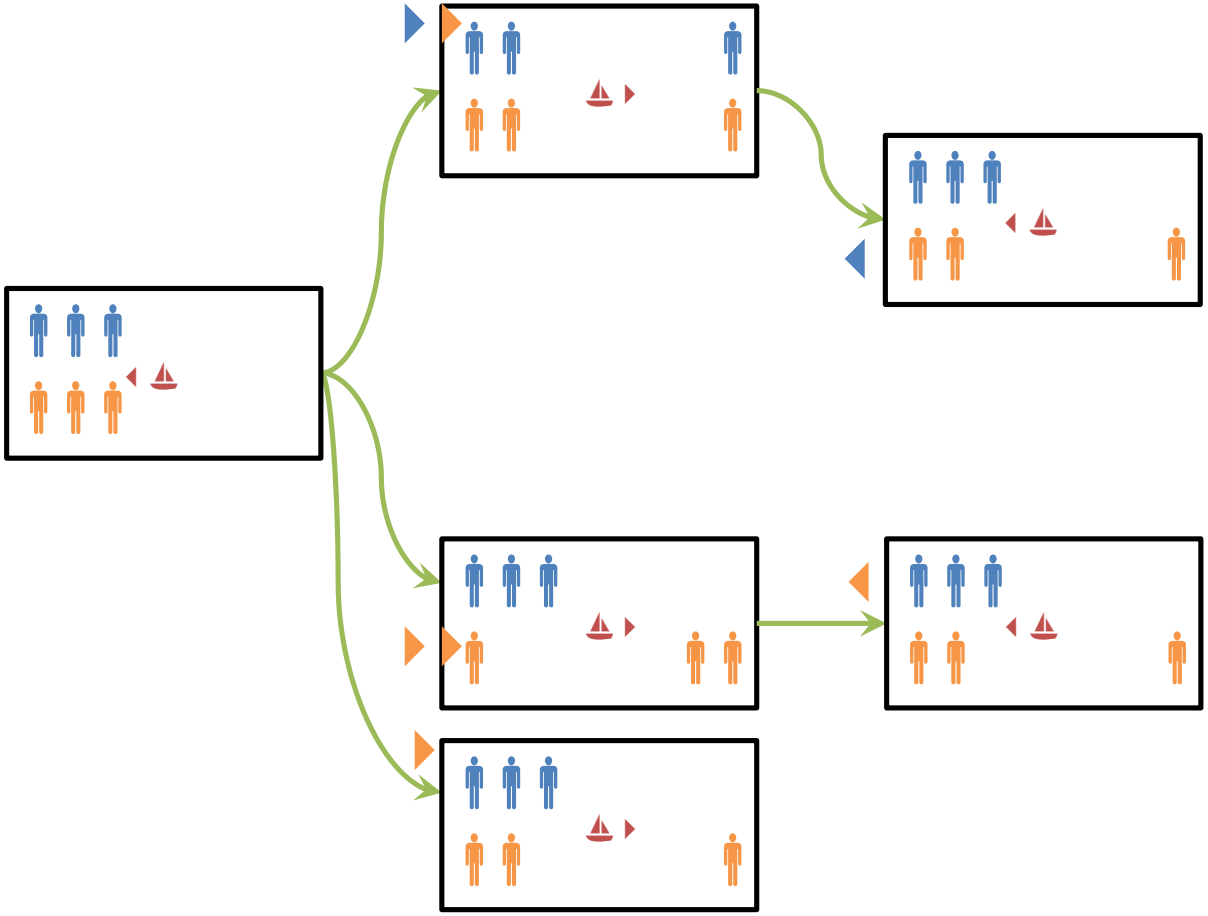


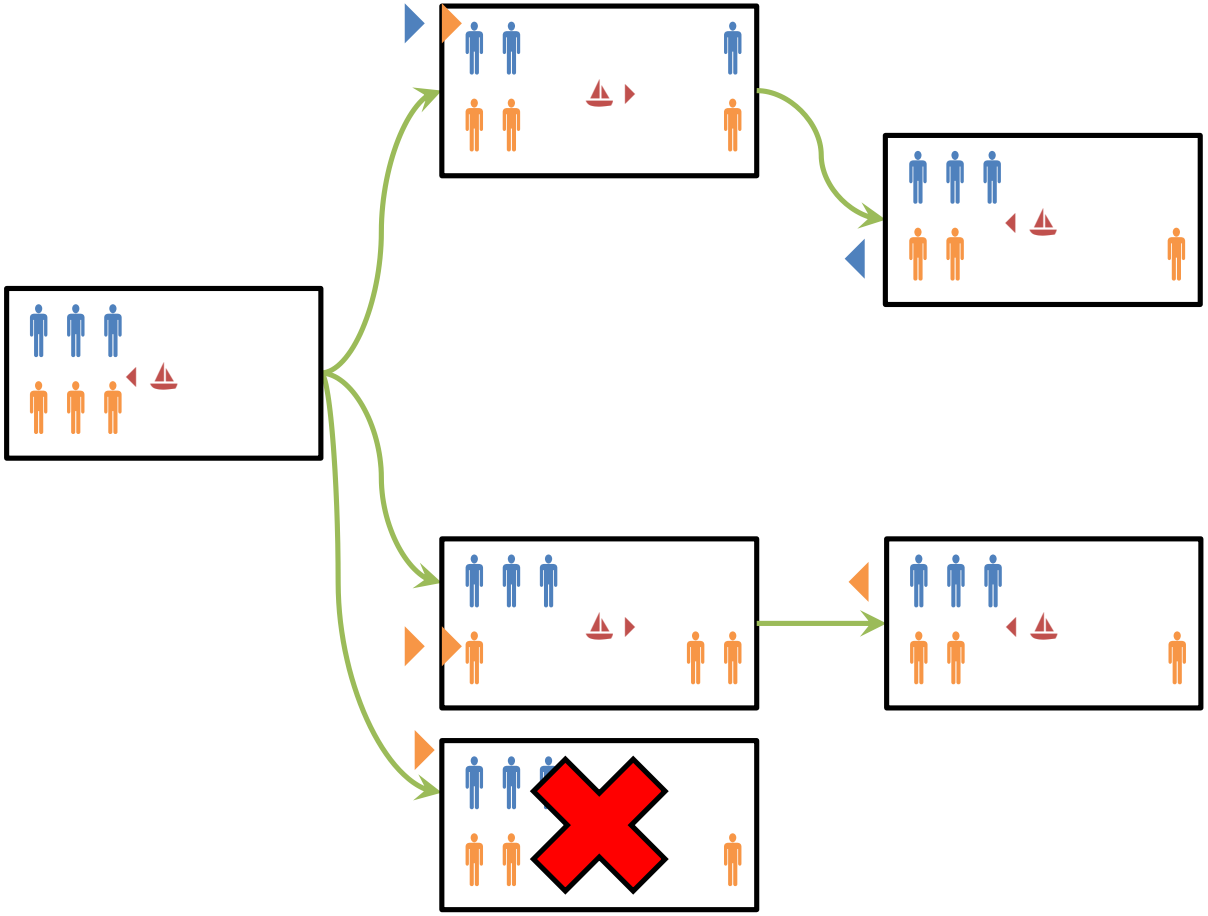


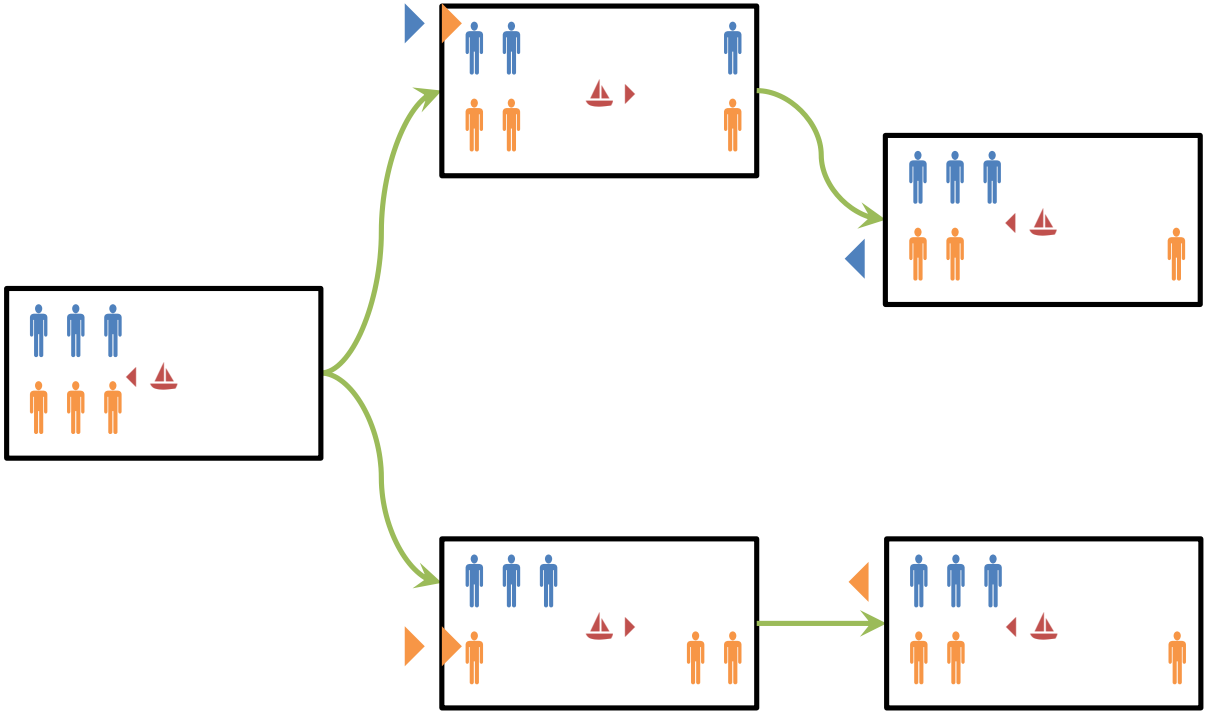


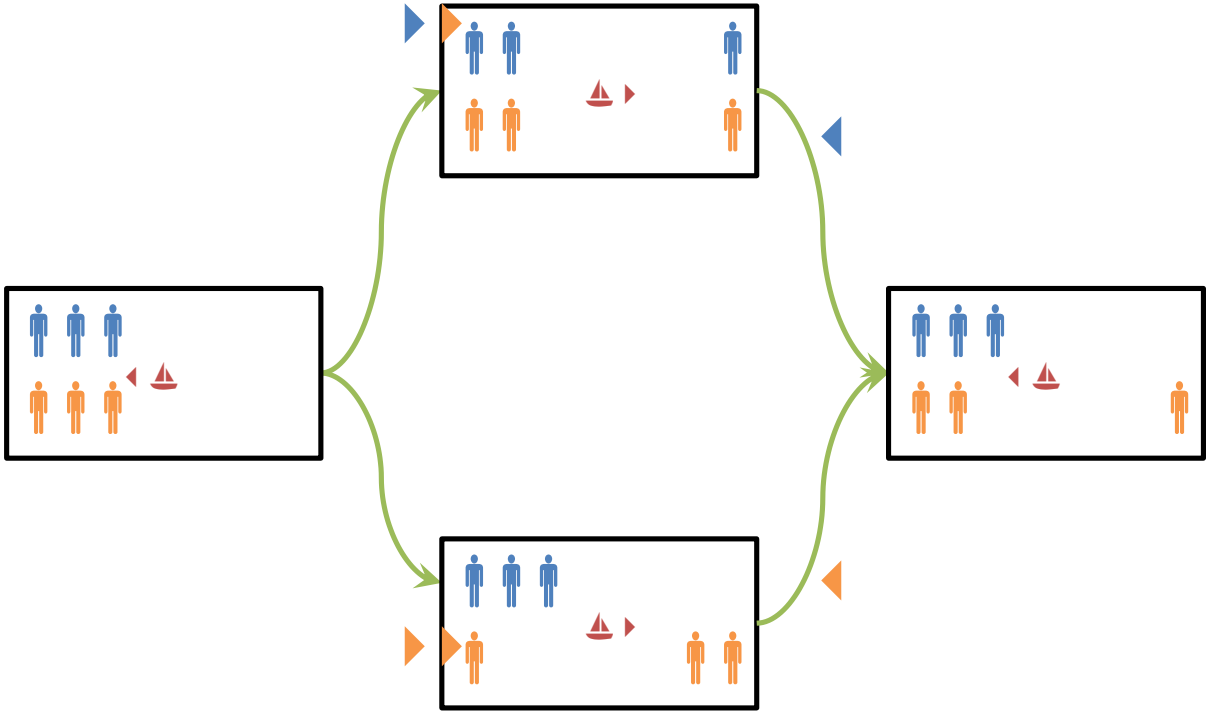


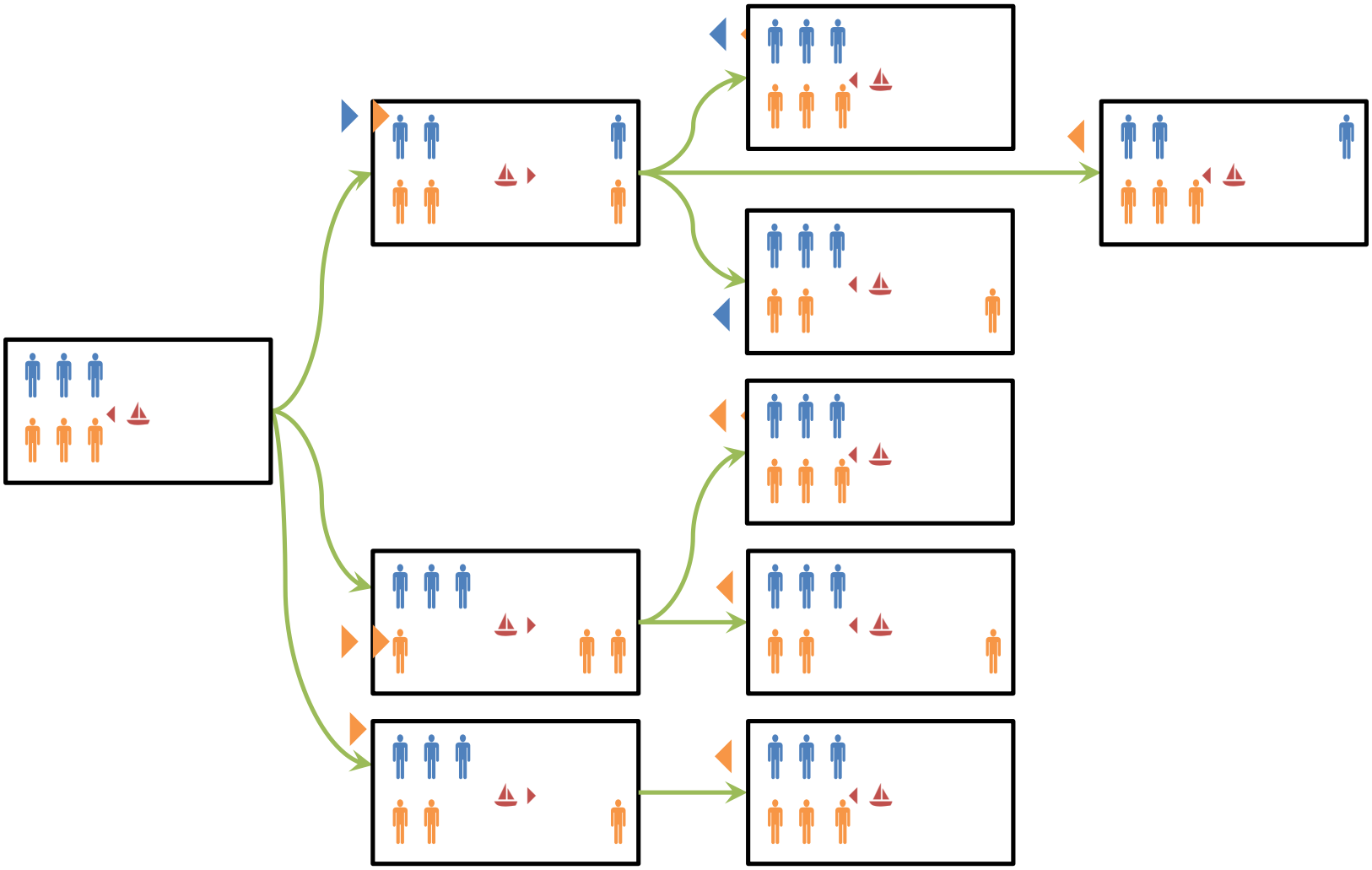


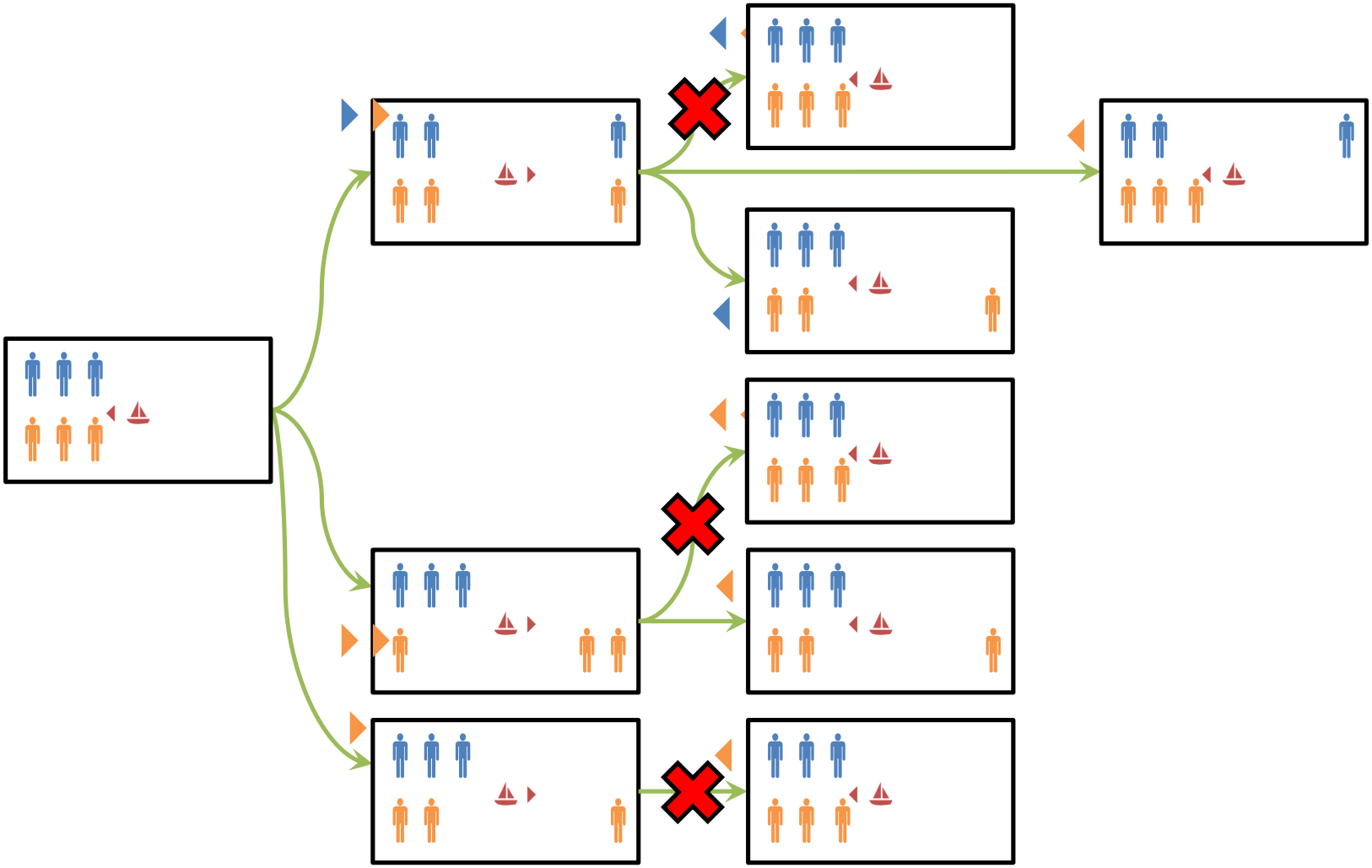


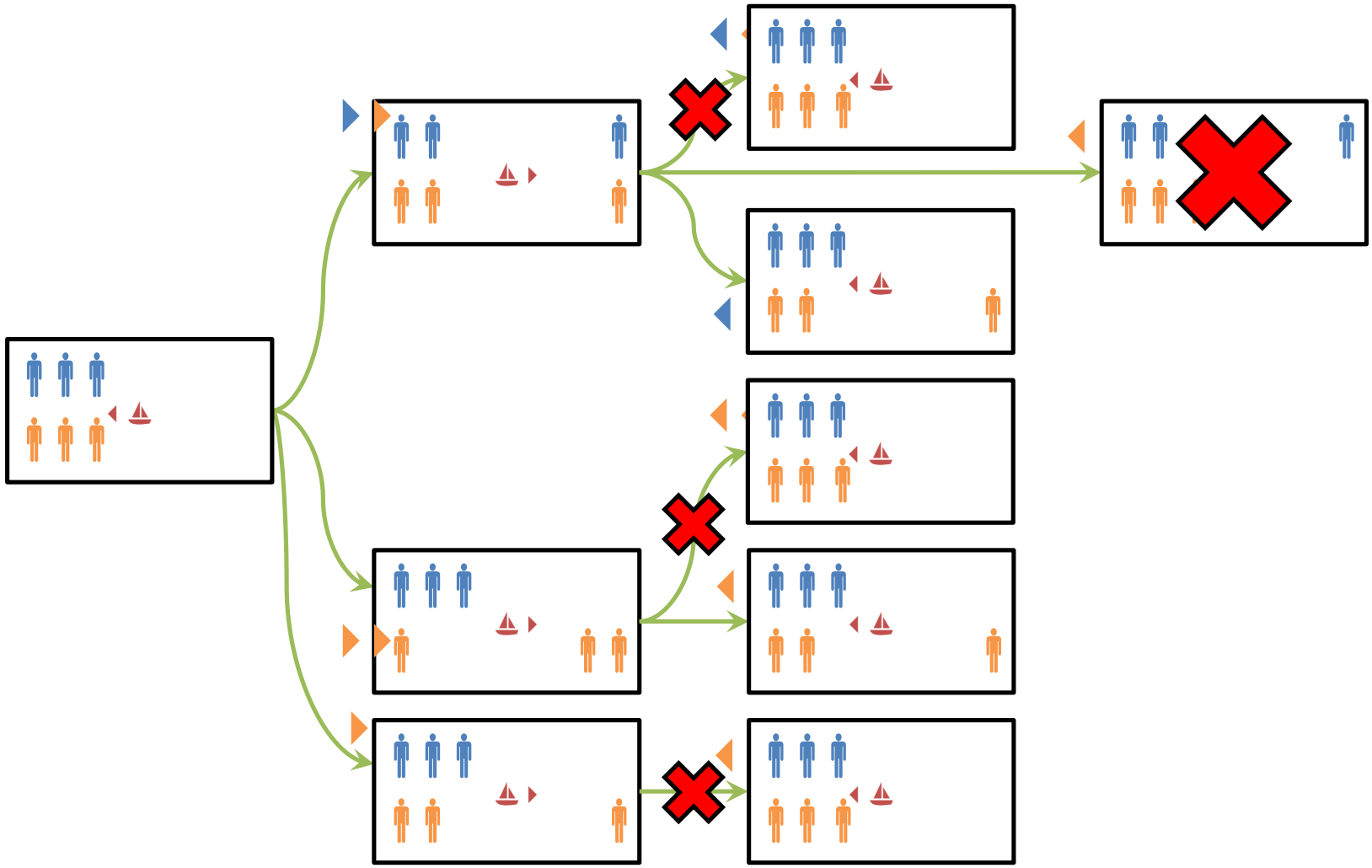


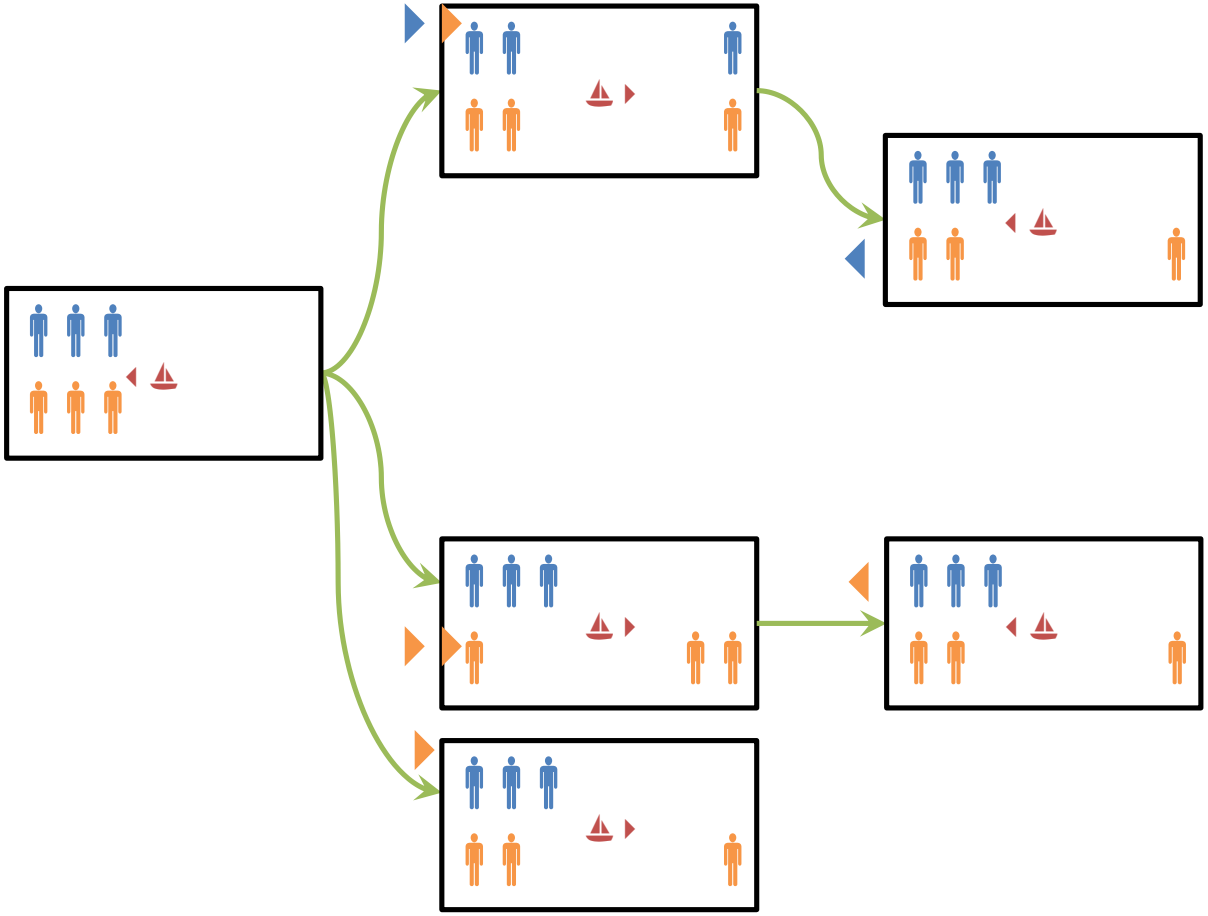


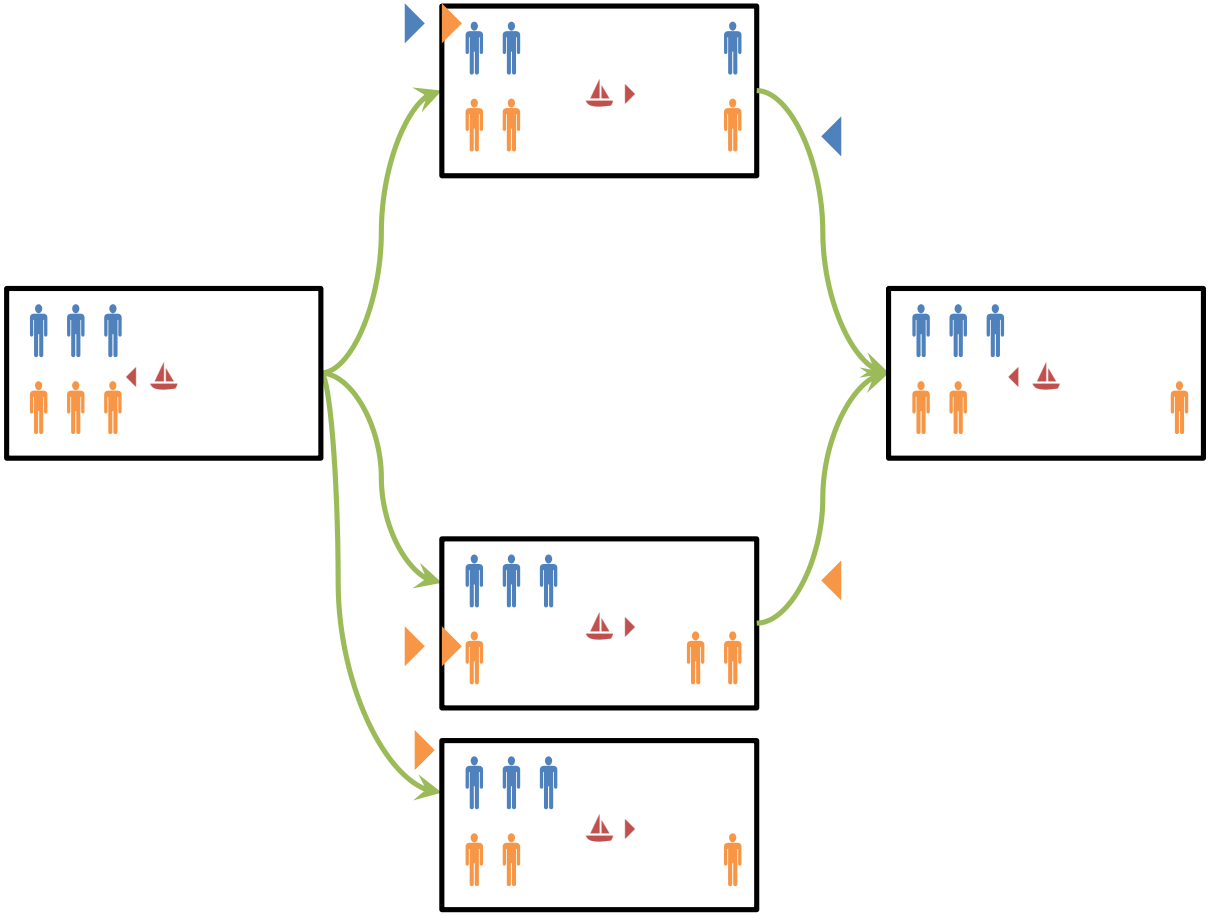


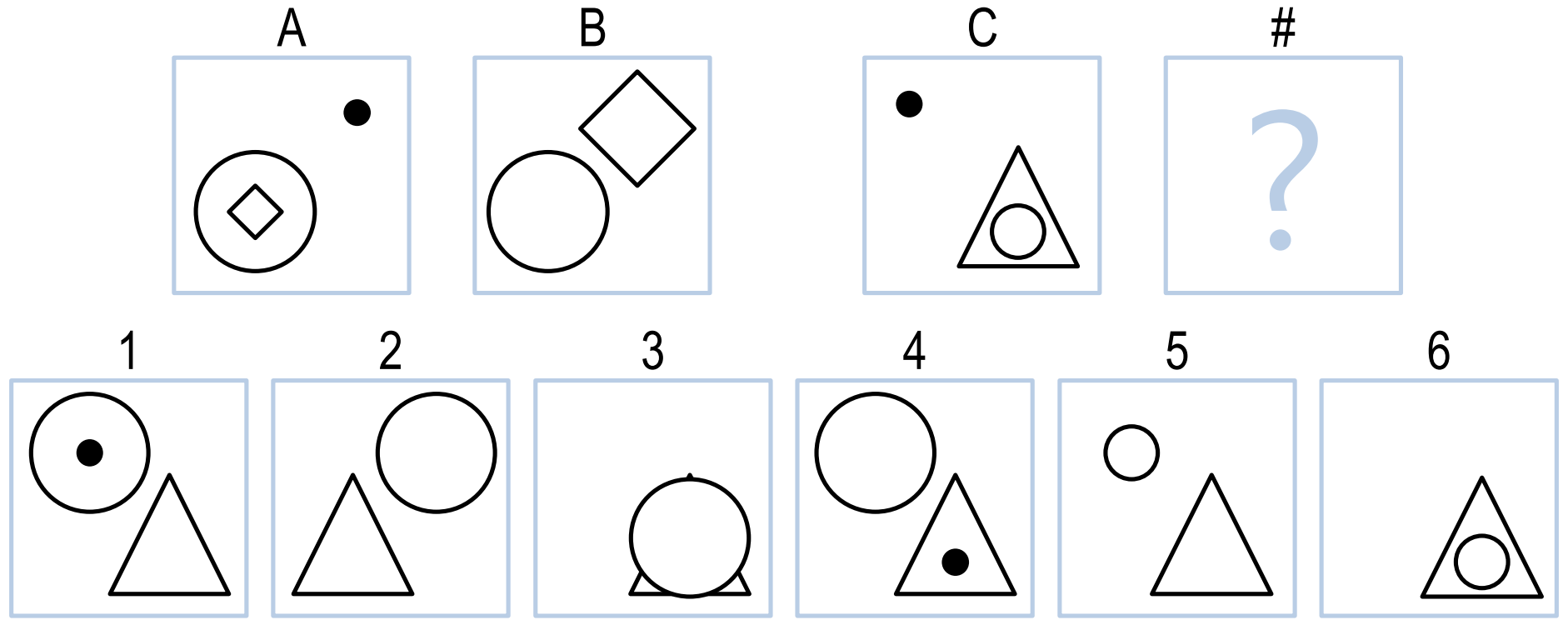


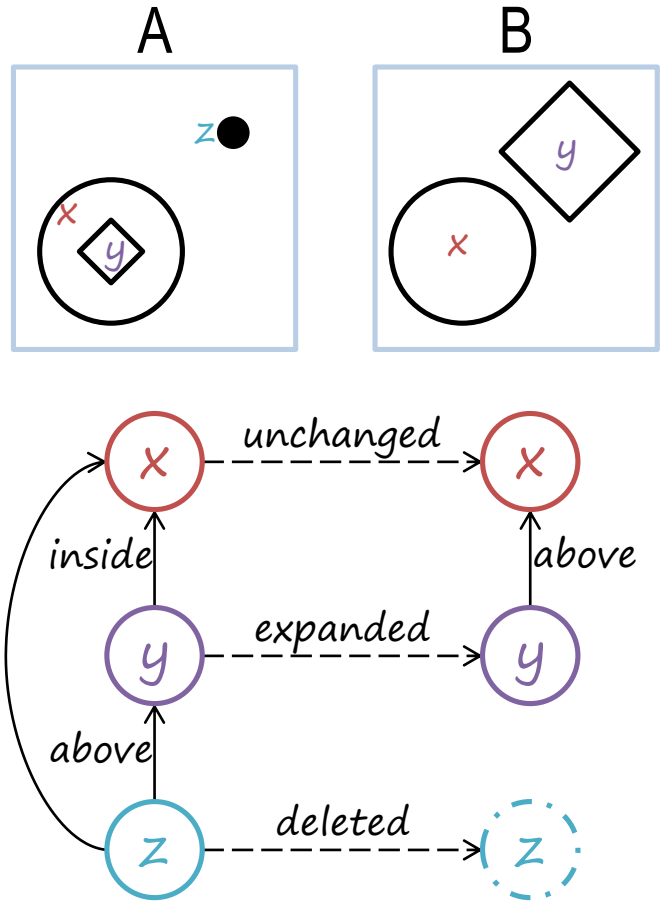


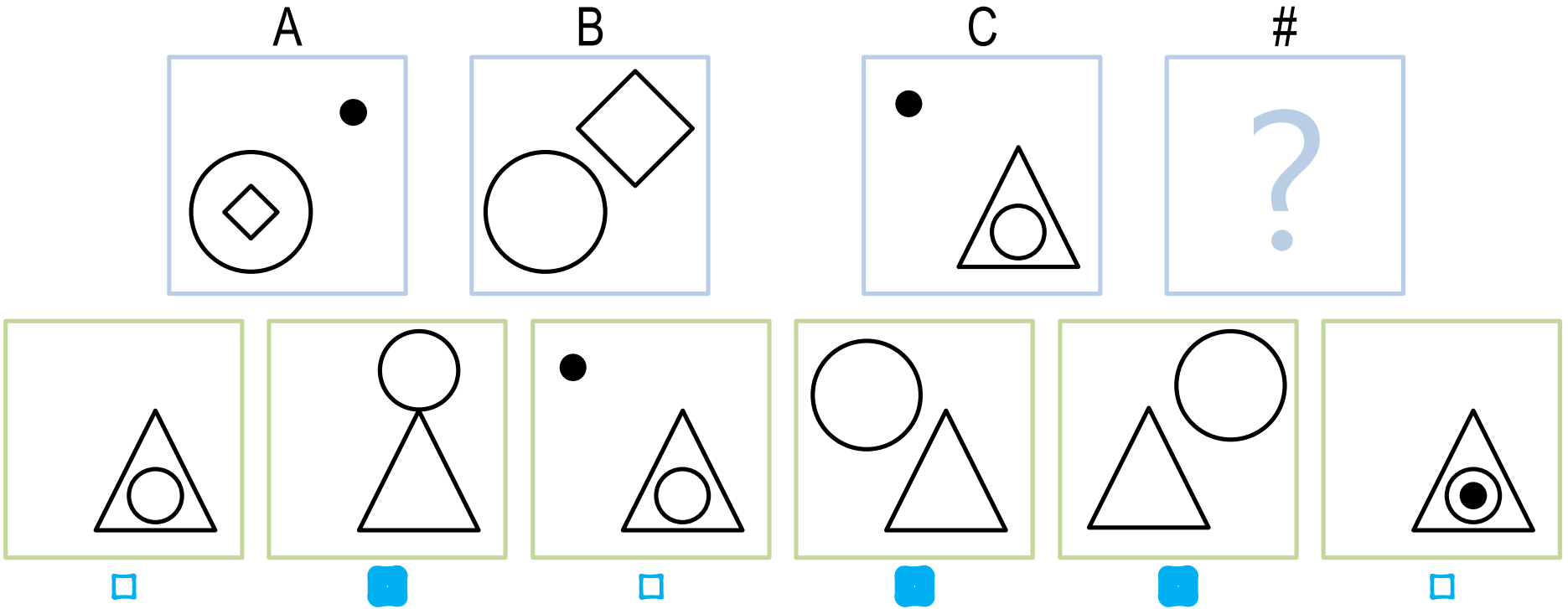






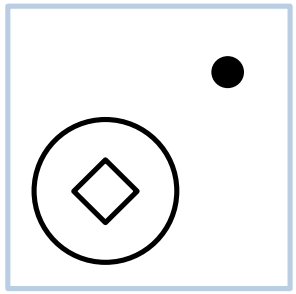

A
B
C
#
?
1
2
3
4
5
6


A
B
z
x
y
x
y
unchanged
x
x
inside
above
above
y
expanded
y
above
z
deleted
z

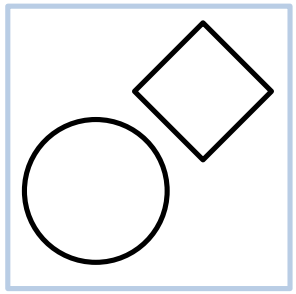



Which of these states might a good generator generate?

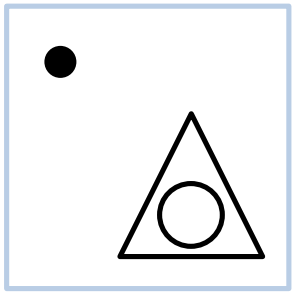
A



B



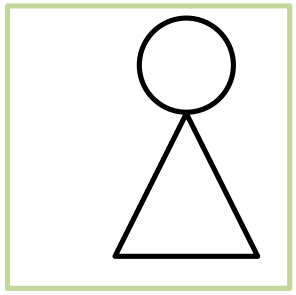
C



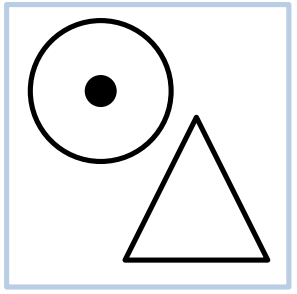
#



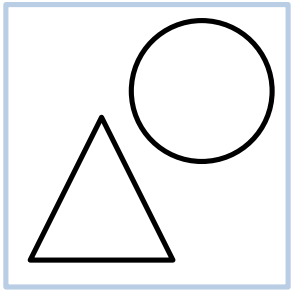

Which answer should our tester select?



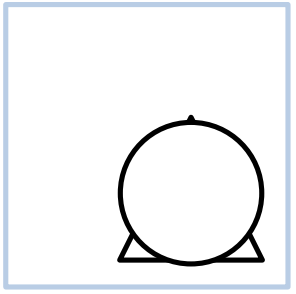
1 o



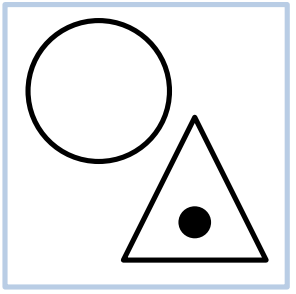
2 ●



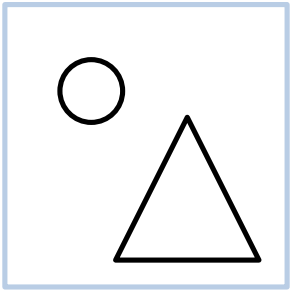
3 o



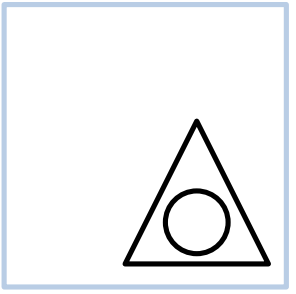
4 o



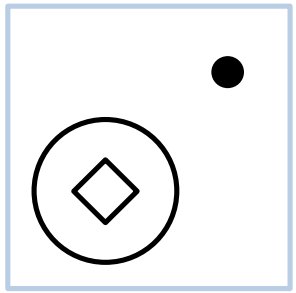
5 o



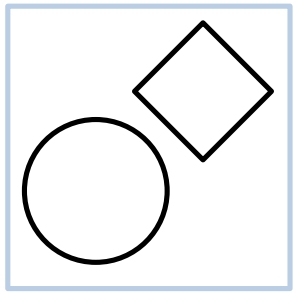
6 o



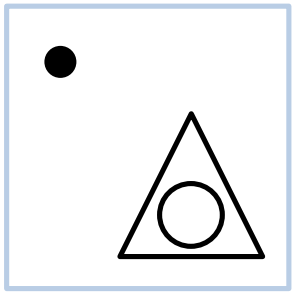
A



B



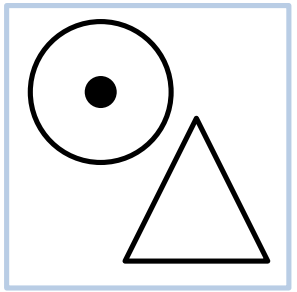
C



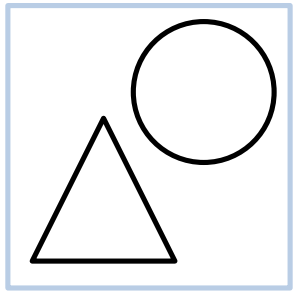
#



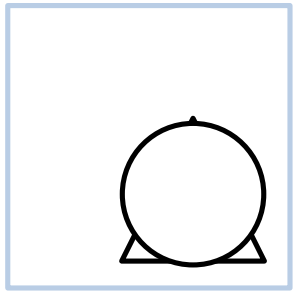

1



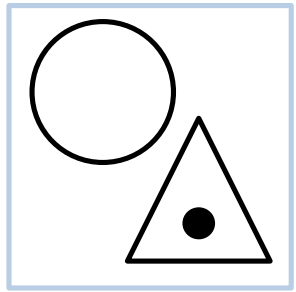
2



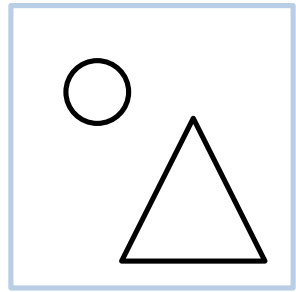
3



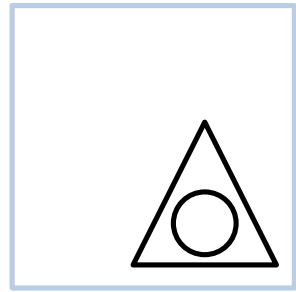
4



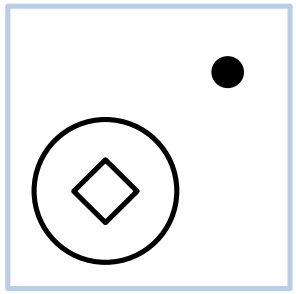
5



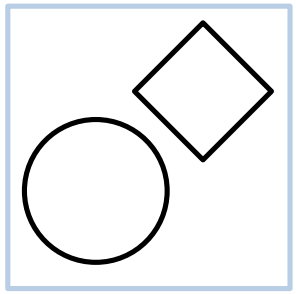
6



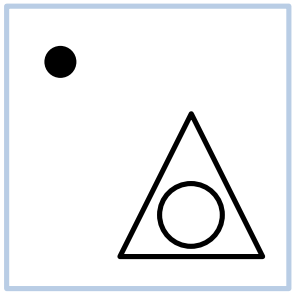
A



B



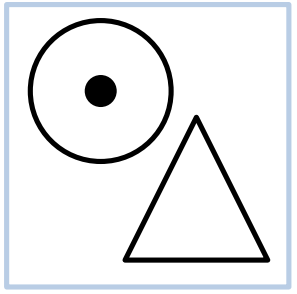
C



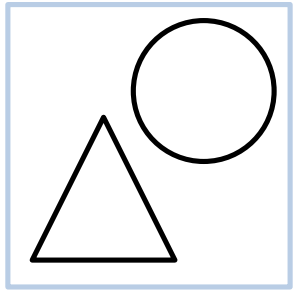
#



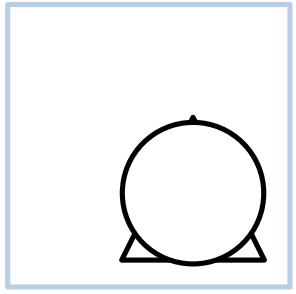

1



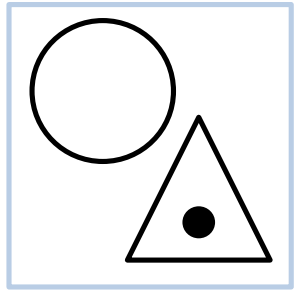
2



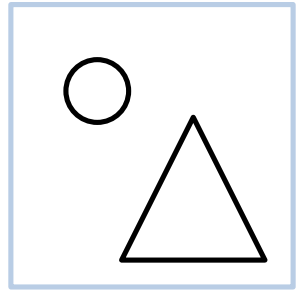
3



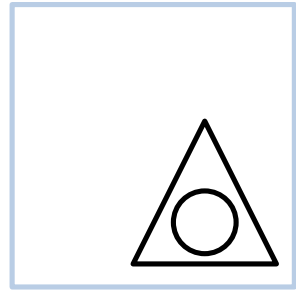
4



5



6



# Good Generators

- Complete
- Efficient
- Smart

# Assignment

How would you use the generate and test method to design an agent that could answer Raven's Progressive Matrices?

# To recap…

- Generate and test
- Smart testers
- Smart generators
- Generate and test in an unconstrained domain