Mutations as Graphs

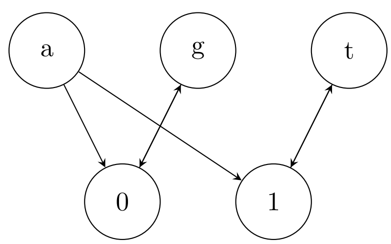
Group 3; Juan, Patrick and John

October 28, 2019

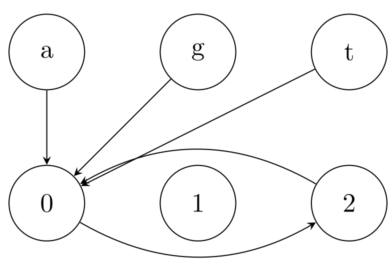
1 Rules as Graphs

Let the nodes labeled with numbers represent the sequence across which the mutation happens. The arrows going into the sequence specify the pattern that must be matched for the mutation to happen. The arrows leaving the sequence determine what the sequence will become after the mutation.

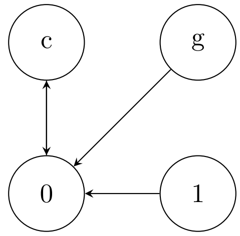
1.1 Rule: ag; at → gt



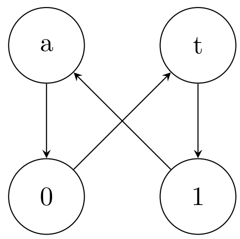
1.2 Rule: agt → 210



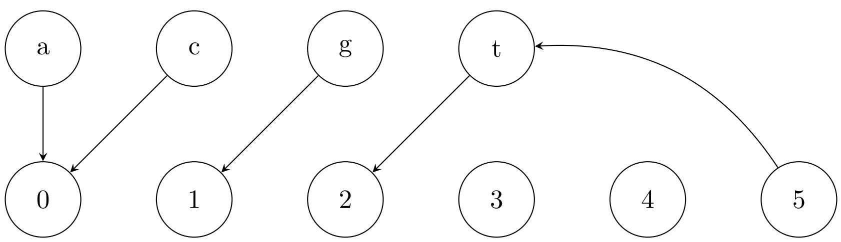
1.3 Rule: cg; → c0



1.4 Rule: a;t → ta

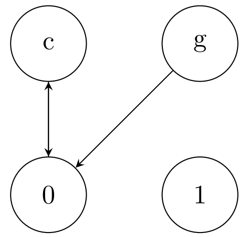


1.5 Rule: ac; g; t → 01234t



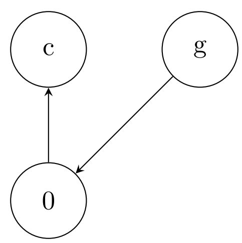
2 Simplifying Rules

2.1 Rule: cg;agtc → c1

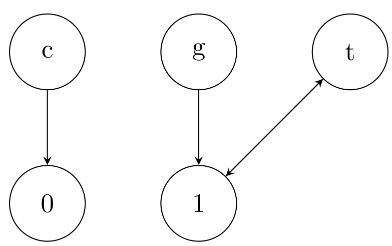


First, we may like to write this rule as cg; → c1, where nothing after the semicolon denotes a match with anything. When a node can match anything, we can say that the mutation has no matching requirement there.

We could simplify this rule to "g → c". Notice that in doing so we would be removing a cycle from the graph. Generally, if we can remove edges from a graph we will say it's simpler. Removing cycles often does not grant an equivalent mutation, but it's worth considering when we can remove cycles. Also notice that we remove the "1" node because the mutation is not required to match there and that node will not mutate. In other words, it has no incoming or outgoing edges. If the node was not on one of the ends of the sequence we would leave it as a placeholder. So now the resulting graph becomes:



2.2 Rule: c;gt → 0t



We can eliminate node 0 because it's dangling all the way to one side with no outgoing edges. Once we're left with a single sequence node, we can then eliminate the cycle. We then get the equivalent rule: g → t