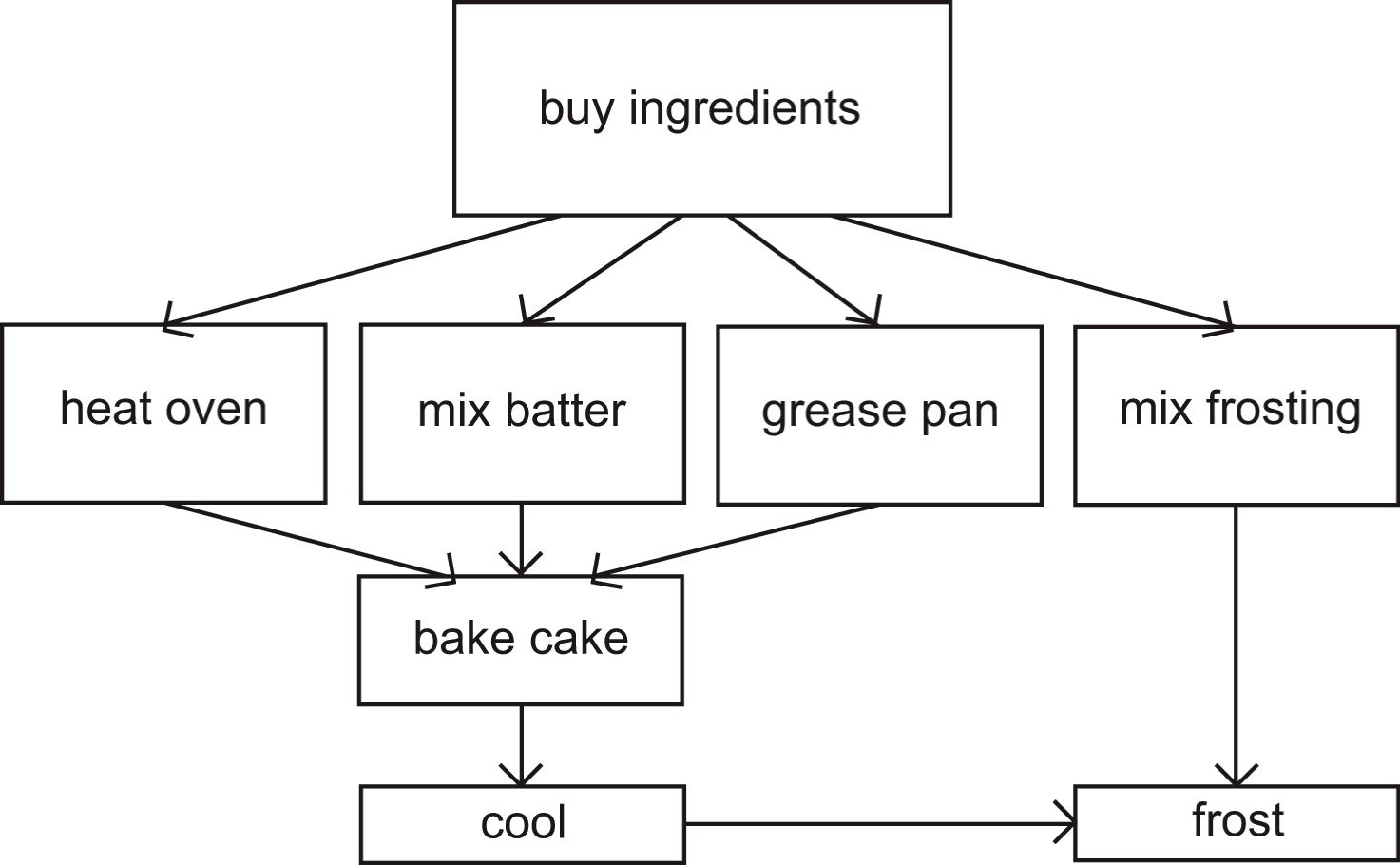
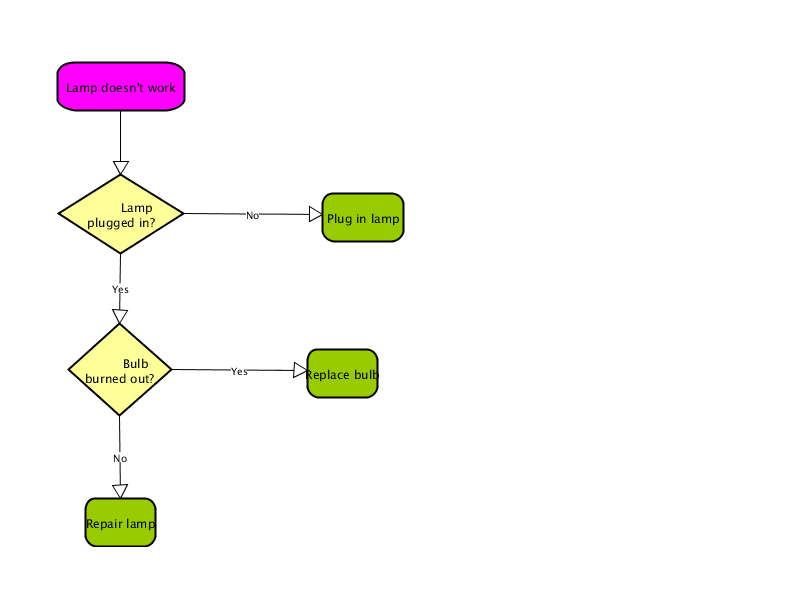
I want to discuss the purpose of the Bohm-Jacopini Theorem. In my research I found that the purpose of the Bohm-Jacopini Theorem is to “solve all problems using a combination of three logical constructs. These logical constructs are Sequence, Selection, and Iteration.” (Winegar, 2019). The Bohm-Jacopini Theorem is also called the structured program theorem. It is a “fundamental principle of computer science.” (Winegar, 2019).

What are Sequence, Selection, and Iteration? To make this easy to understand, sequence is “completing a task by taking it one step at a time.” (Winegar, 2019). A sequence structure can be easy to understand if you think in the terms of a flow chart. The author in the video states that a sequence can like “following a recipe to bake a cake.” (Winegar, 2019). As you move downwards through the flow chart, you are “falling through the logic.” (Winegar, 2019). I have included this flow chart below as a visual:



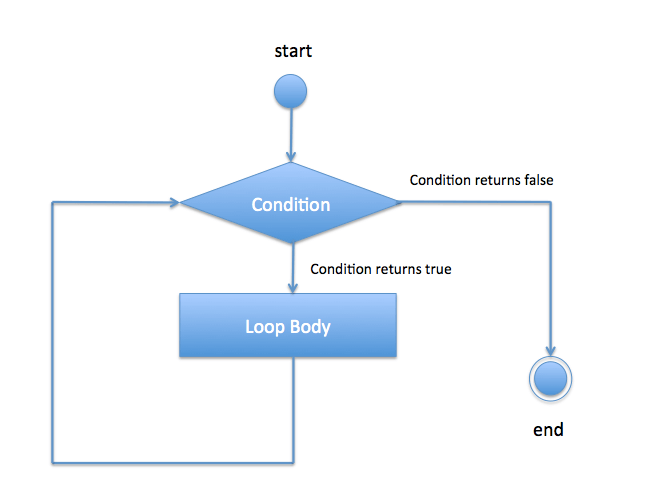
(*Cake Flow Chart*, 2022).

To best understand selection the narrator in the video states selection is “Alternate tasks are performed based upon the truth of a condition.” (Winegar, 2019). This example can be explained in the flow chart below. In this example the lamp does not work. As we fall through the conditions we come to a solution. For instance, we can see falling to “is the lamp plugged?” if no, then we move to plug in the lamp. If the lamp is plugged in then we fall through to is the bulb burned out, if yes then we replace the bulb, if no, then we fall through to replace the lamp.



(*Lamp Flow Chart*, n.d.)

Iteration or repetition is a “task or set of tasks repeated until a predetermined event occurs.” (Winegar, 2019). This is also called looping or a while loop. I have included the flow chart below to show a generic example of looping. As we fall through the conditions if the condition is true, we will keep iterating the “while loop”. It is only when the condition returns as false that we will move to the end of the loop or “outside of the loop”. For example, if the condition = 11, we will have moved through the through the loop body at least ten times until the condition meets 11 and then we will move outside of the loop.



(whileLoopFlowDiagram, n.d).

In conclusion, we can see from the visual illustrations above how the Bohm-Jacopini Theorem can be sufficient in programming by using a simple methodology to solve problems. By falling through the logic to a specific condition our programs can then be directed to the next step. They may operate an alternate task based on the condition to move to the next step in the process or they may operate in a while loop to keep repeating the condition until the condition is false to move outside of the while loop.

References:

Winegar, M. (2019, March 24). *JavaScript Episode #9: The Structured Theorem* [Video]. Vimeo. <https://vimeo.com/30499767>

*Cake Flow Chart*. (2022). [Illustration] <https://people.well.com/user/abs/Writing__/Nonfiction/Survival_/BW/cake.jpg/>

*Lamp Flow Chart*. (n.d.). [Illustration]. Diagramo.com. <https://diagramo.com/editor/raster.php?hash=n1qyX0&type=png/>

*whileLoopFlowDiagram*. (n.d.). [Illustration]. Java2blog.com. <https://www.java2blog.com/wp-content/uploads/2017/04/whileLoopFlowDiagram.png>