Amanda Ollis - December 5, 2025
CSCI 4890 – Dr. Tanya Lueder

Smoothie Machine Ordering System

Project Overview

The problem I will implement is a Smoothie Machine Ordering System. In this system, a customer makes a smoothie by choosing a size, a base, one or more add-ins, an ice level, and then pressing the blend button . The machine only accepts orders that include these choices in the correct order. My machine also allows the customer to start their order over (reset transitions) or cancel their order (dead state).

System Alphabet Σ = {s, b, a, i, reset, m} .

Those alphabets represent the following:

- s = choose a size
- b = choose a base (water, juice, or milk)
- a = add an add-in (fruit or protein powder)
- i = choose ice level (light, regular, extra)
- reset = machine returns to start (allows the customer to start over)
- m = press the blend button
- c = cancel order

System States Q = { q_0 , q_1, q_2, q_3, q_4, q_5, $(Dead\ States - q_{d0}, q_{d1}, q_{d2}, q_{d3}, q_{d4})$}

q_0 – Start State

- No selections have been made yet.
- The machine is waiting for the customer to begin their order.
- Next valid moves:
  - choose **s** (size of their smoothie) to move to $\boldsymbol{q_1}$ state.

q_1 – Size Selected

- The customer has selected their smoothie size (small, medium, or large).
- Next valid moves:
  - choose **b** (a base for your smoothie) to move to $\boldsymbol{q_2}$ state.
  - choose **reset** (start over) to move back to $\boldsymbol{q_0}$ start state.
  - choose **c** (cancel order) to move to dead state.

Amanda Ollis - December 5, 2025
CSCI 4890 – Dr. Tanya Lueder

Smoothie Machine Ordering System

q_2 – Base Selected

- The customer has selected their smoothie base (water, milk, or juice).
- Next valid moves:
  - choose **a** (add-ins for your smoothie) to move to q_3 state.
  - choose **reset** (start over) to move back to q_0 start state.
  - choose **c** (cancel order) to move to dead state.

q_3 – Add-Ins Selected

- This state accepts more than one add-in (fruits or protein powder)
- The smoothie machine will remain in this state as long as the customer keeps adding add-ins.
- Next valid moves:
  - choose **a** (continue adding more add-ins) causing the machine to loop.
  - chose **I** (ice level for smoothie) to move to q_4 state.
  - choose **reset** (start over) to move back to q_0 start state.
  - choose **c** (cancel order) to move to dead state.

q_4 – Ice Level Selected

- The customer has chosen their ice level (light, regular, or extra).
- Next valid moves:
  - choose **m** (to press blend button) to move to q_5 accept state.
  - choose **reset** (start over) to move back to q_0 start state.
  - choose **c** (cancel order) to move to dead state.

q_5 – Accept State (Blend)

- The machine smoothie order is complete, and the machine starts to blend (accept) the order.

$q_{d0}, q_{d1}, q_{d2}, q_{d3}, q_{d4}$– Reject States

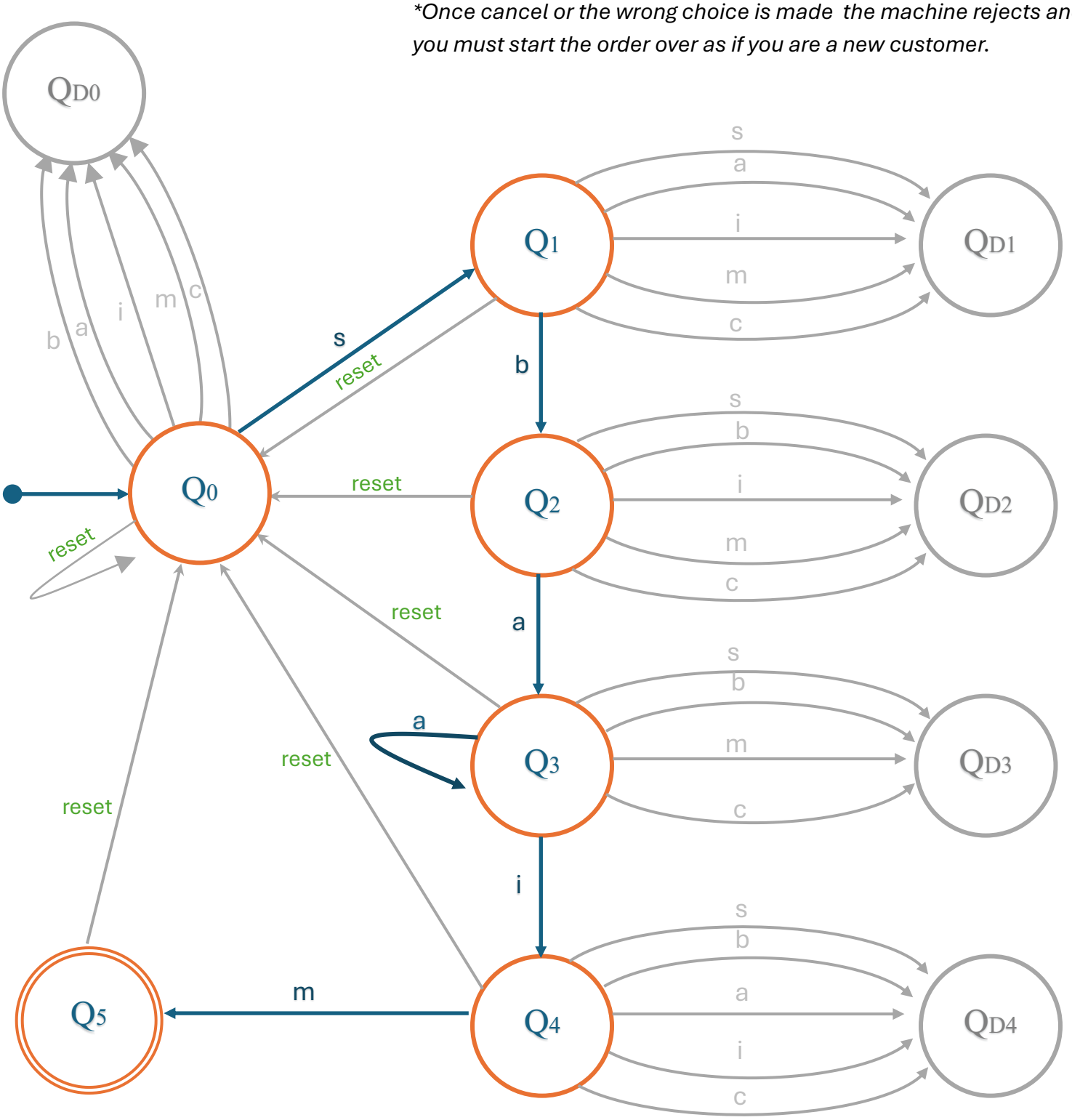
- The customer chose **c** to cancel their order.
- Valid moves:
  - **reset** (the order is complete and machine resets for next customer) moves to q_0 start state.

Amanda Ollis - December 5, 2025
CSCI 4890 – Dr. Tanya Lueder

Smoothie Machine Ordering System

DFA Implementation

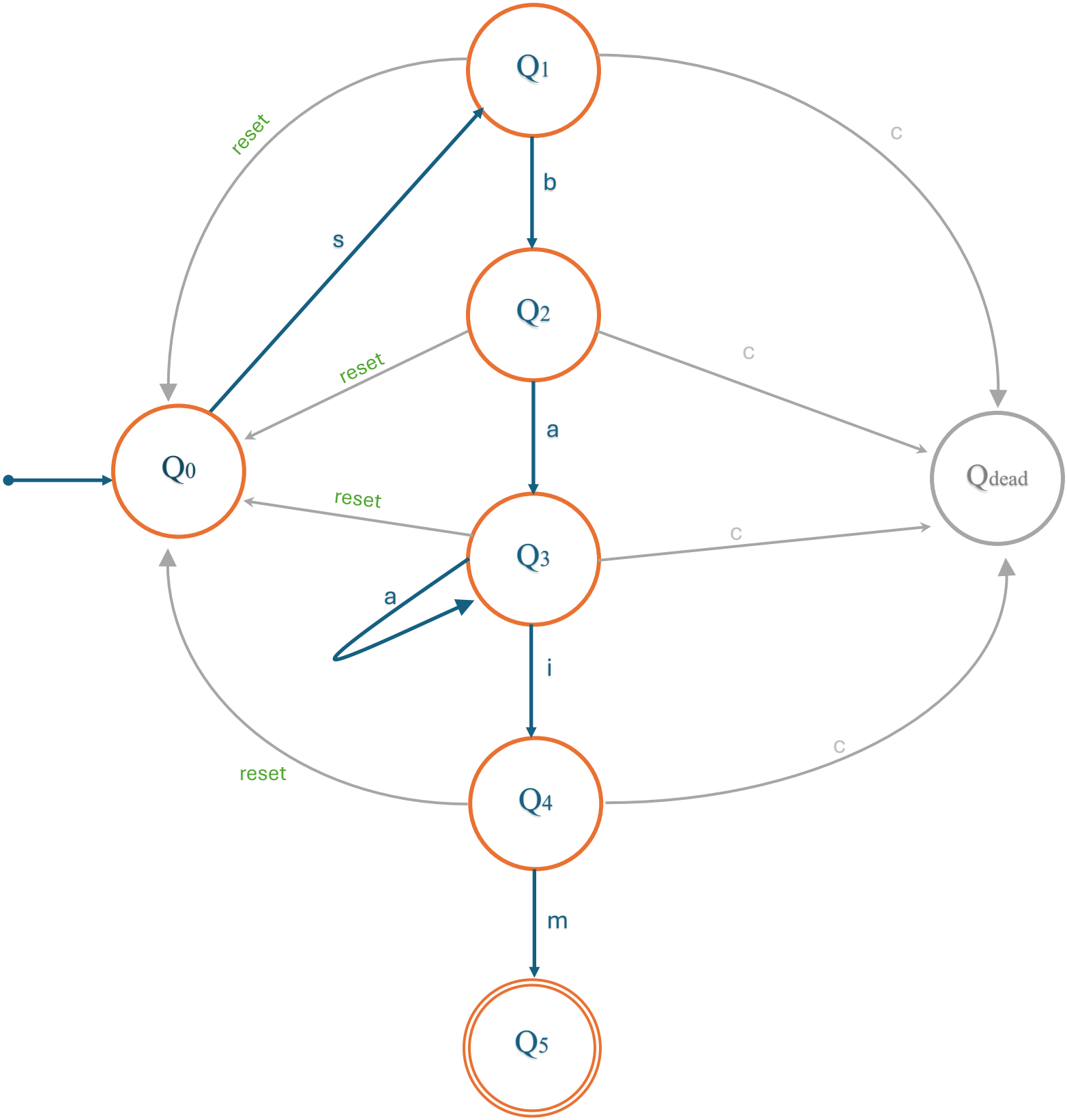
**Once cancel or the wrong choice is made the machine rejects and you must start the order over as if you are a new customer.*



Amanda Ollis - December 5, 2025
CSCI 4890 – Dr. Tanya Lueder

Smoothie Machine Ordering System

NFA Implementation



Amanda Ollis - December 5, 2025
CSCI 4890 – Dr. Tanya Lueder

Smoothie Machine Ordering System

Context Free Grammar for Accepted Orders Implementation

I used my state labels as nonterminals and alphabets as terminals to make my CFG understandable in the context of my project. This is reflective of my DFA.

Q0 ----> s Q1 |

Q1 ----> b Q2 | r Q0

Q2 ----> a Q3 | r Q0

Q3 ----> a Q3 | i Q4 | r Q0

Q4 ----> m Q5 | r Q0

Q5 ----> ε | r Q0

** r = reset

Regular Language for Accepted Orders Implementation

This regular language expression shows that my smoothie machine ordering system accepts the following choices for a valid order to complete: size, base, one or more add-ins, ice level, and blend with the option to reset as needed.

$$L = (s\ b\ a+\ i\ m)^*$$

Accepted Cases

1. Size (**s**), Base (**b**), Add-On (**a**), Add-on (**a**), Ice Level (**i**), Blend (**m**)
2. Size (**s**), Base (**b**), Reset (**reset**), Add-In (**a**), Ice Level (**i**), Blend (**m**)
3. Size (**s**), Base (**b**), Add-In (**a**), Reset (**reset**), Size (**s**), Base (**b**), Add-In (**a**), Add-In (**a**), Add-In (**a**), Ice Level (**i**), Blend (**m**)

Rejected Cases

1. Size (**s**), Base (**b**), Ice-Level (**i**), Blend (**m**)
2. Size (**s**), Base (**b**), Add-On (**a**), Add-on (**a**), Cancel (**c**)