Finite Automata for Coffee Shop Order System

Introduction

We will be designing a formal language that simulates a structured version of how a coffee shop order system works. This automaton will model the sequence of actions a customer takes when customizing their drink. It ensures that each part of the order follows a consistent format, reducing errors and improving processing efficiency.

The structure can be used for UI/UX improvements, backend logic validation, or physical order kiosk design. It also serves as a good foundation for understanding how state machines apply in real-world processes.

Alphabet

Order Type: {Latte, Espresso, Cappuccino, ColdBrew, Chai, Matcha}
Milk: {Whole, Oat, Skim, Almond}
Flavors: {Vanilla, Caramel, Mocha, Hazelnut, CookieButter, Honey, SfVanilla, Raspberry, Blueberry}
Size: {Small, Medium, Large}
Temp: {Iced, Hot}
Toppings: {Whip, Coldfoam, Cinnamon, Salt, CookieButterChunks, BlueberryChunks}
Add-Ons: {ExtraShot, LightIce, Decaf, Stevia, ExtraMatcha, ExtraSweetener, RistrettoShot}
Name: {Name[x]}
Pick-Up: {ForHere, ToGo}
Checkout: {Checkout}

Semantics

1. Start with an **Order Type** (e.g., Latte, Espresso).
2. Specify a **Size** (Small, Medium, or Large).
3. Choose the **Temperature** (Iced or Hot).
4. Optionally include a **Milk** option (Whole, Oat, etc.).

5. Optionally include **Flavors** (Vanilla, Raspberry, etc.).
6. Optionally include **Toppings** (Whip, Salt, etc.).
7. Optionally include **Add-Ons** (Decaf, Stevia, etc.).
8. Optionally specify **Name** (Name[x] for any customer name).
9. Choose a **Pick-Up** method (ForHere or ToGo).
10. End the order with **Checkout**.

Example Orders

1. **Order 1: Latte Medium Iced Whole Mocha Salt ExtraShot NameIbrahim ToGo Checkout**
    - Latte, Medium, Iced, Whole, Mocha pumps, Salt, ExtraShot, NameIbrahim, ToGo

2. **Order 2: Matcha Large Iced Almond Blueberry Blueberry Chunks ExtraMatcha NameZack ForHere Checkout**
    - Matcha, Large, Iced, Almond milk, Blueberry syrup, Blueberry chunks, extra matcha, nameZack, For here

3. **Order 3: Chai Small Hot Oat Vanilla Cinnamon λ NameDulli ToGo Checkout**
    - Chai, Small, Hot, Oat Milk, Vanilla syrup, Cinnamon, No Addons, NameDulli, To go

Grammar

S → O
O → OrderType T
OrderType → Latte | Espresso | Cappuccino | ColdBrew | Chai | Matcha

T → Temp M
Temp → Iced | Hot

M → Milk F

Milk → Whole | Oat | Skim | Almond | λ

F → Flavor G
Flavor → Vanilla | Caramel | Mocha | Hazelnut | CookieButter | Honey | SfVanilla | Raspberry | Blueberry | λ

G → Topping A
Topping → Whip | Coldfoam | Cinnamon | Salt | CookieButterChunks | BlueberryChunks | λ

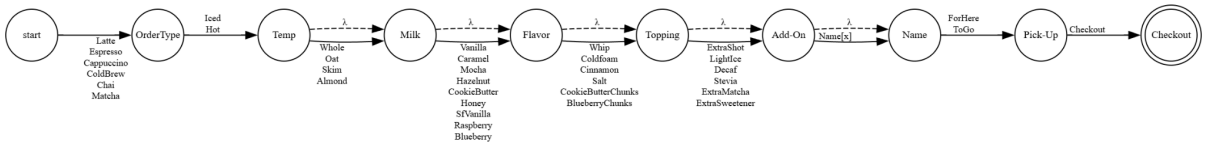
A → AddOn N
AddOn → ExtraShot | LightIce | Decaf | Stevia | ExtraMatcha | ExtraSweetener | λ

N → Name P
Name → Name[x] | λ

P → PickUp C
PickUp → ForHere | ToGo

C → Checkout

Automata



This photo is also in the GitHub as **graphviz(2).png**

Here is the dot code that we put into GraphvizOnline to make graph above

```
digraph CoffeeOrder {
  graph [layout=circo, overlap=false, splines=curved];
  node  [shape=circle, fixedsize=true, width=0.7, height=0.7, fontsize=10];
  edge  [fontsize=9];

  /* states */
```

```
  q0 [label="start"];
  q1 [label="OrderType"];
  q2 [label="Temp"];
  q3 [label="Milk"];
  q4 [label="Flavor"];
  q5 [label="Topping"];
  q6 [label="Add-On"];
  q7 [label="Name"];
  q8 [label="Pick-Up"];
  q9 [shape=doublecircle, label="Checkout"];   // accepting state

  /* transitions with line-wrapped labels */
  q0 -> q1 [label="Latte\nEspresso\nCappuccino\nColdBrew\nChai\nMatcha"];

  q1 -> q2 [label="Iced\nHot"];

  q2 -> q3 [label="λ", style=dashed];
  q2 -> q3 [label="Whole\nOat\nSkim\nAlmond"];


  q3 -> q4 [label="λ", style=dashed];
  q3 -> q4
[label="Vanilla\nCaramel\nMocha\nHazelnut\nCookieButter\nHoney\nSfVanilla\nRaspbe
rry\nBlueberry"];


  q4 -> q5 [label="λ", style=dashed];
  q4 -> q5
[label="Whip\nColdfoam\nCinnamon\nSalt\nCookieButterChunks\nBlueberryChunks"];

  q5 -> q6 [label="λ", style=dashed];
  q5 -> q6 [label="ExtraShot\nLightIce\nDecaf\nStevia\nExtraMatcha\nExtraSweetener"];


  q6 -> q7 [label="λ", style=dashed];
  q6 -> q7 [label="Name[x]"];


  q7 -> q8 [label="ForHere\nToGo"];
  q8 -> q9 [label="Checkout"];
}
```

Description of Schema

The automaton schema models a coffee-shop ordering process, enforcing a fixed sequence while allowing customers to skip certain categories. It contains 10 states (q0 through q9) that correspond to each step of the order: order-type, temperature, milk, flavor, topping, add-on, name, pick-up, and checkout.

1. Required Components:
Every order starts in state q0 and must move through three mandatory stages in sequence.

- Order Type (q1) – The customer selects a drink style such as Latte, Espresso, Cappuccino, Cold Brew, Chai, or Matcha.

- Temperature (q2) – They specify whether the drink is Iced or Hot.

- Pick-Up Method (q8) – Finally, they indicate For Here or To Go.
  There are no λ-transitions out of these stages, so the automaton cannot reach checkout until all three choices have been made.

2. Optional Components:
After temperature, the customer may visit up to four optional categories, each of which can be skipped via a λ-transition or repeated where self-loops are available.

- Milk (q3) – Whole, Oat, Skim, Almond; may be skipped entirely.

- Flavor (q4)
  – Vanilla, Caramel, Mocha, Hazelnut, Cookie Butter, Honey, Sugar-Free Vanilla, Ra spberry, Blueberry; also skippable.

- Toppings (q5)
  – Whip, Cold Foam, Cinnamon, Salt, Cookie Butter Chunks, Blueberry Chunks; a self-loop on q5 allows any number of toppings, including none.

- Add-Ons (q6)
  – Extra Shot, Light Ice, Decaf, Stevia, Extra Matcha, Extra Sweetener; q6 likewise has a self-loop for unlimited selections.

- Name on Cup (q7) – An explicit name token (Name[x]) may be given or skipped with λ.

3. Checkout:
The special input Checkout moves the automaton from state q8 to the accepting state q9, finalising the order. Because Checkout is only recognised after the Pick-Up decision, every accepted sequence is guaranteed to include an Order Type, a Temperature, and a Pick-Up method, while all other categories remain optional at the customer's discretion.

Instructions for Using the Coffee Shop Order Machine

How to Use:

1. Run the file using the command:
   python testing.py or throw it into an online Python compiler
2. You will be prompted to enter your order as a single string.

3. Follow the logical order for menu items:

   - **Order Type** (Latte, Espresso, etc.)

   - **Size** (Small, Medium, Large)

   - **Temperature** (Hot, Iced)

   - **Milk** (Whole, Oat, Skim, Almond) — Optional

   - **Flavor** (Vanilla, Caramel, Mocha, etc.) — Optional

   - **Topping** (Whip, Coldfoam, Cinnamon, etc.) — Optional

   - **Add-On** (ExtraShot, Decaf, etc.) — Optional

   - **Name** (Name followed by any text, e.g., "NameZack")

- **Pick-Up Method** (ForHere, ToGo)
- **Checkout** (Required to complete the order)

Understanding Lambdas (Optional Steps):

- If you skip any optional step, it is treated as **λ (lambda)** — which means "no selection".
- For example:
  - "Latte Small Hot Checkout" means you skipped milk, flavor, topping, add-ons, and name.
  - "Latte Small Iced Whole Checkout" means you skipped flavor, topping, add-ons, and name.

Example strings:

Valid Stings

- Latte Medium Iced Oat Mocha Whip ExtraShot NameZack ForHere Checkout
- Chai Large Hot Skim Honey Cinnamon ExtraSweetener NameAva ForHere Checkout
- ColdBrew Medium Iced Almond Caramel Coldfoam Decaf NameAlex ToGo Checkout
- Matcha Large Hot Whole SfVanilla BlueberryChunks LightIce NameJennifer ForHere Checkout

Invalid Stings

- Latte Medium Iced Whole Vanilla LightIce ForHere Checkout (missing name)
- Espresso Small Hot Vanilla Whip ExtraShot NameZack ForHere Checkout (Missing "Milk" type after "Hot")
- Chai Large Hot Skim NameAva ForHere Checkout (Missing "Flavor" and "Topping")
- ColdBrew Medium Iced Almond Caramel CookieButterChunks NamePaul ForHere (Missing "Checkout")