**RAPPORT**

**INTRODUCTION**

Ce projet consiste à concevoir, modéliser et vérifier un système basé sur des automates. L’objectif est de garantir la conformité du système aux spécifications en utilisant des méthodes formelles. Le système étudié est un modèle simplifié de gestion de commandes dans un restaurant, avec des états bien définis (q0 à q5), illustrant les étapes de traitement d’une commande.

**1. Spécification du Système avec la Logique Formelle (Étape 1)**

Le système est spécifié en termes d’états et de transitions, décrivant les actions possibles pour chaque étape d’une commande. Les états principaux sont définis comme suit :

|  |  |
| --- | --- |
| ETATS | DESCRIPTION |
| Q0 | En attente de commande |
| Q1 | Commande en préparation |
| Q2 | Commande en prête |
| Q3 | Commande livrée |
| Q4 | Commande annulée |
| Q5 | Commande rejetée |

• q0 : En attente de la carte d’accès

Le système est dans cet état au démarrage, en attente qu’une carte soit insérée.

• q1 : En attente du code d’accès

Si la carte insérée est valide, le système passe à cet état pour que l’utilisateur saisisse son code d’accès.

• q2 : Code d’accès incorrect, deuxième tentative

Si la première tentative de saisie du code est incorrecte, le système passe dans cet état en attendant la deuxième tentative.

• q3 : Accès accordé

Si le code saisi est correct, le système accorde l’accès et passe à cet état.

• q4 : Accès refusé, alarme déclenchée

Si le code saisi est incorrect lors de la deuxième tentative ou si la carte est invalide, l’alarme est déclenchée, et l’accès est refusé.

**Transitions entre les états :**

Les transitions entre ces états sont définies par des actions :

• De l’état q0 :

• Carte valide → Passage à l’état q1.

• Carte invalide → Passage à l’état q4 (alarme déclenchée).

• De l’état q1 :

• Code correct → Passage à l’état q3 (accès accordé).

• Code incorrect → Passage à l’état q2 (attente de la deuxième tentative).

• De l’état q2 :

• Code correct → Passage à l’état q3 (accès accordé).

• Code incorrect → Passage à l’état q4 (alarme déclenchée).

Règles logiques :

Les règles logiques qui régissent ces transitions sont les suivantes :

• Si la carte est valide, le système passe à l’état q1.

• Si la carte est invalide, l’alarme est déclenchée, et le système passe à l’état q4.

• Si le code est correct, l’accès est accordé, et le système passe à l’état q3.

• Si le code est incorrect, le système attend une deuxième tentative, et passe à l’état q2.

• Si la deuxième tentative est correcte, l’accès est accordé (passage à l’état q3).

• Si la deuxième tentative échoue, l’alarme est déclenchée (passage à l’état q4).

Expression logique des transitions :

1. q0 ^ CarteValide → q1

2. q0 ^ CarteInvalide → q4

3. q1 ^ CodeCorrect → q3

4. q1 ^ CodeIncorrect → q2

5. q2 ^ CodeCorrect → q3

6. q2 ^ CodeIncorrect → q4

**2. Modélisation du Système avec des Automates et Langages Formels (Étape 2)**

Automate Fini Déterministe (AFD) :

L’automate fini déterministe (AFD) que nous avons créé modélise les états et transitions du système de contrôle d’accès. Il est défini par un ensemble d’états, un alphabet d’entrées (cartes et codes), et un ensemble de transitions qui spécifient le changement d’état en fonction des actions effectuées.

États :

• q0 : En attente de la carte d’accès.

• q1 : En attente du code d’accès.

• q2 : Première tentative échouée, en attente de la 2e tentative.

• q3 : Accès accordé.

• q4 : Accès refusé, alarme déclenchée.

Alphabet des entrées :

• CarteValide

• CarteInvalide

• CodeCorrect

• CodeIncorrect

Transitions :

Les transitions sont gérées par un tableau de transitions ou une map de type Map<String, Map<String, String>> :

• De q0 :

• Carte valide → Passage à q1.

• Carte invalide → Passage à q4.

• De q1 :

• Code correct → Passage à q3.

• Code incorrect → Passage à q2.

• De q2 :

• Code correct → Passage à q3.

• Code incorrect → Passage à q4.

Le code java de cet automate se situe dans les classes ci-dessus.

**Conception de Circuits Logiques avec l’Algèbre de Boole (Étape 3)**

Règles logiques :

Nous utilisons les règles de logique booléenne pour simuler les transitions entre les états. L’algèbre de Boole permet de simplifier les expressions des transitions et de concevoir des circuits logiques optimisés.

Les transitions peuvent être exprimées comme des expressions logiques grâce à l’utilisation de cartes de Karnaugh pour simplifier

:• Accès accordé (q3) :

q3 = (q1^CodeCorrect) v(q2^CodeCorrect})

• Alarme déclenchée (q4) :

q4 = (q0^CarteInvalide)v (q2 ^(CodeIncorrect)