SYNTHESE ET RAPPORT FINAL

**Introduction**

Le système de contrôle d’accès développé dans ce projet vise à permettre l’entrée dans un bâtiment sécurisé à l’aide d’un **système basé sur des cartes et des codes personnels**. Le projet se divise en plusieurs étapes qui utilisent des concepts de **logique formelle**, de **modélisation par automates**, de **conception logique** et de **validation formelle**. Ce rapport présente une synthèse de ces étapes.

**Objectifs du Projet**

Le projet a pour objectifs :

* Spécifier les règles logiques permettant d'autoriser ou refuser l'accès en fonction des cartes et codes.
* Modéliser un automate pour gérer les états du système.
* Implémenter un système logique pour valider l’accès à un bâtiment en fonction des cartes et des codes.
* Valider formellement le système en s'assurant de sa robustesse et de sa conformité aux spécifications.

Les améliorations apportées incluent la gestion des cartes et descodes, permettant à l’utilisateur d'enregistrer et tester différentes cartes et codes via une interface simple.

### ****Développement du Système****

### ****Spécification avec Logique Formelle****

* Les règles de sécurité du système ont été spécifiées en utilisant des expressions logiques, par exemple :
  + Si la carte est valide ET le code est correct, l’accès est accordé.
  + Si la carte est invalide, l’accès est refusé et une alarme est déclenchée.
  + Si la carte est valide mais le code est incorrect, l’accès est refusé.

La classe **LogiqueFormelle** gère la vérification de la validité des cartes et des codes.

#### **Modélisation avec Automate**

* Le système utilise un **automate fini déterministe (AFD)** pour représenter les états du système, permettant de suivre les transitions entre les différents états (ex. : initial, validation de la carte, validation du code, accès accordé, alarme, etc.).
* La classe **AutomatesLangages** permet de gérer les transitions d’états et d'afficher l’état actuel du système.

**Conception Logique**

* La classe **AccessSysteme** intègre les règles logiques et l’automate pour vérifier les accès. Elle permet d'enregistrer les cartes et codes, puis de valider un accès en fonction des informations fournies.
* Les cartes et les codes sont stockés dans une **Map** (dictionnaire) et peuvent être vérifiés à chaque tentative d'accès.

**Validation et Vérification Formelle**

* La classe **ValidationSysteme** permet de vérifier la conformité des états de l'automate avec les règles logiques, en simulant des scénarios d'attaque pour tester la robustesse du système.

Tests de Robustesse et Vérification Formelle :

**Enregistrement des Cartes et Codes :**

* + Le système permet d'enregistrer des cartes et leurs codes associés. Ces informations sont stockées dans une **Map** et peuvent être utilisées pour valider les accès futurs.

**Validation des Accès :**

* + Lorsqu'une personne tente d’accéder au bâtiment, elle doit entrer l’ID de sa carte et son code.
  + Le système vérifie si la carte est valide et si le code associé est correct. En fonction du résultat, l’accès est soit accordé, soit refusé.

**Gestion des États du Système :**

* + L'automate gère les états du système et effectue les transitions nécessaires en fonction des événements (validation de la carte, validation du code).

**Tests de Robustesse et Vérification Formelle :**

* + Le système inclut des tests pour vérifier sa robustesse face à des attaques simulées et une validation formelle pour s’assurer qu’il respecte bien les règles spécifiées.

### ****Résultats et Vérification****

#### **Scénarios de Test**

Les tests réalisés ont couvert différents scénarios :

* **Test 1 :** Carte valide et code correct → Accès accordé.
* **Test 2 :** Carte invalide → Accès refusé et alarme déclenchée.
* **Test 3 :** Carte valide mais code incorrect → Accès refusé.

Ces tests ont validé le bon fonctionnement du système, en assurant qu'il répond bien aux règles logiques spécifiées.

**Validation du Fonctionnement**

Le système a été validé avec succès grâce aux vérifications formelles et aux tests manuels des cartes et codes. Le processus de validation formelle a confirmé que les transitions d’états respectent les spécifications du système.

**Conclusion**

Ce projet a permis de développer un **système de contrôle d'accès** complet, en utilisant des principes de logique formelle et de modélisation par automate. Le système répond aux besoins de base d'un contrôle d’accès sécurisé, tout en étant éligible pour des améliorations futures.

Les tests effectués ont confirmé la robustesse et la conformité du système, et les fonctionnalités d'enregistrement de cartes et de codes permettent de gérer facilement l’accès des utilisateurs. Le programme est prêt à être étendu avec de nouvelles fonctionnalités.