



Mini-mémoire

Model Checking CTL

[Cours : INFO-F308]

PAQUET Michael - 000410753
PROMOTEUR : GEERAERTS Gilles

08 Novembre 2017

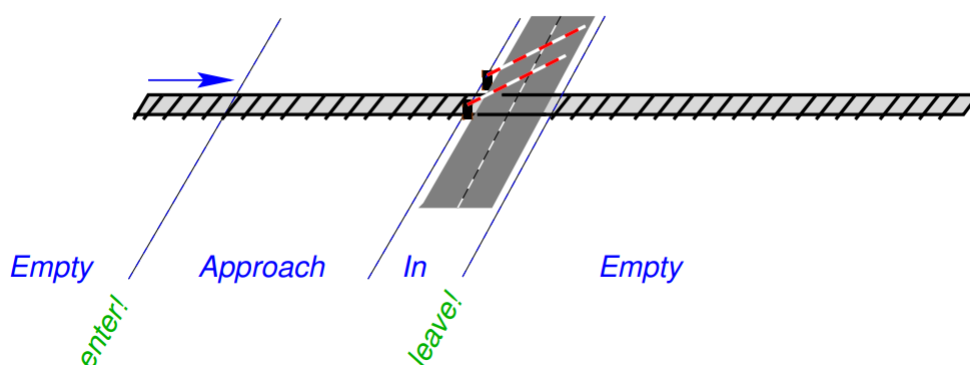
Table des matières

1	Introduction	2
2	Chapitre 1	4
2.1	Du graphe aux arbres de recherche	4

1 Introduction

Avec l'évolution technologique faisant acte de présence jours après jours, la plupart des services, informatique ou non, sont maintenant gérés via des systèmes informatiques vérifiant leur bon fonctionnement. Mais pour certains services, il est impératif que ces systèmes fonctionnent correctement.

Prenons l'exemple d'une voie ferrée :



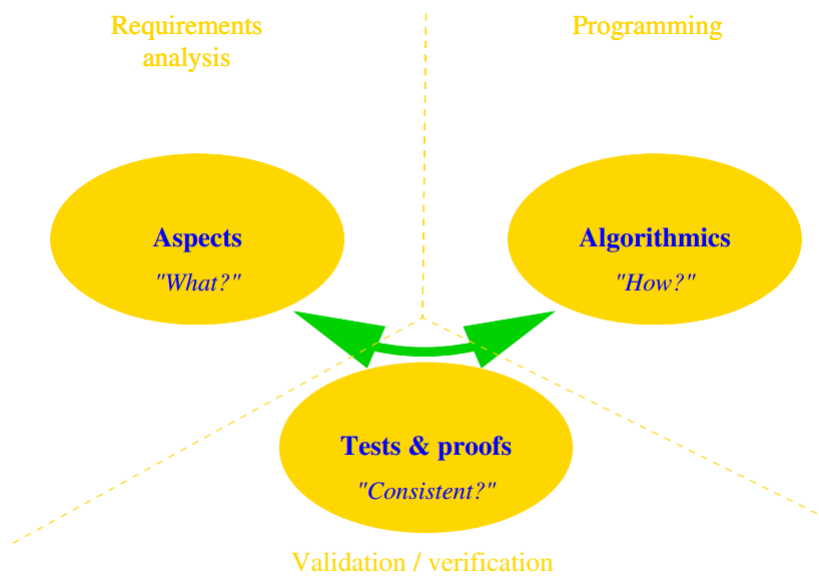
Dans ce cas de figure-ci, il est impératif que les portes soient fermées lorsque le train traverse le passage à niveau. Comme dit précédemment, un système informatique s'occupera de gérer cela.

Mais puisqu'on ne peut pas se permettre la moindre erreur, nous nous devons de vérifier que le système n'échouera jamais, et c'est dans ce cas de figure que le *Model Checking* fut créé et utilisé.

Au cours de cette dissertation, nous passerons donc en revue chacun des points de vue qu'on peut adapter lorsqu'on parle de *Model Checking CTL*. Le premier point que nous aborderons sera l'aspect analytique du modèle (Qu'est-ce qu'on doit vérifier?). Lors de ce chapitre, nous allons montrer la façon dont on découpe un système pour en faire des structures propices à l'analyse et à la vérification. Des structures comme les *Kripke models* ou encore les arbres seront bien évidemment abordées. Une fois ces objets observés, nous étudierons la façon dont on peut vérifier l'efficacité d'un système.

Après avoir déterminé la façon dont on peut vérifier l'exactitude d'un système, nous évoquerons la consistance de telles méthodes. Nous prouverons donc de manière rigoureuse l'efficacité du *Model Checking CTL*.

Enfin, nous aborderons la question algorithmique : Comment implémenter un tel modèle ?



Il est évident que notre sujet ne s'arrêtera pas à ces points-ci. A titre d'exemple, nous comparerons également notre *Model Checking CTL* (*Computational Tree Logic*) avec le *LTL* (*Linear temporal logic*).

2 Chapitre 1

2.1 Du graphe aux arbres de recherche