自主課題研究：Timed-Automataによる リアルタイムシステムのモデル検証

The Model Checking of Realtimed System with Timed-Automata

情報システム工学科 3 年 13 番　　姜　帆

1　まえがき

リアルタイムシステムは組み込み機器などに多くつかわれており、航空機や自動車などに代表される人命に関わるシステムでは特に時間制約が厳しい。また、こうしたシステムでは外部環境に影響を受け、その動作の状態は複雑であり、システムの整合性の検証は難しい分野である。今回は Timed-Automata に基づく、時相論理式を用いた検証手法についての調査、そして共同で製作したレゴロボットのシステムを理論的に検証しようとする試みである。

2　Timed-Automata について

Timed-Automata の定義 Timed-Automata（以下時間オートマトン）はω-オートマトンに、クロックによる時間制約という概念を付加したものである。クロック制約とはシステム内部にあるクロックを条件とし、経過時間を測りながら、状態の遷移可能性について記述したものである。
時間オートマトンは G=〈 S,s, Σ,X,inv,E 〉として 6 つの組で定義される：

- ノードの有限集合 S
- 初期ノード s
- イベントの有限集合Σ（外部動作、内部動作）
- クロックの有限集合 X
- 不変式の関数 inv
- エッジの有限集合 E

3　検証ソフト Uppaal

UPPAAL を選択した理由
UPPAAL は主に Aalborg University、Uppsala University で作られ、変数やクロック値、チャンネル通信などを扱うことができ、GUI によりグラフィカルに操作できる点で他のソフトより優れていると判断した。

4　時相論理式

時相論理とは命題論理と時相オペレータによる表現であり、システムの検証においては重要な項目である。以下は UPPAAL で実際に使用できる時相オペレータである。

NAME	Property	Equivalent to
Possibly	E <> p	
Invariantly	A [] p	not E <> not p
Potentially alway	E[]	
Eventually	A <> p	not E[] not p
Leads to	p －－> q	A[] (p imply A <> q)

これらは、シミュレーションを行うだけでは見出すことのできないエラーを見つけ出すのに役立つ。

5　レゴロボットの検証

今回、グループで共同制作したレゴロボットを時間オートマトンで記述し、UPPAAL でモデル検証を行う。（レゴロボットプログラムの仕様については、森本および仲居による報告書を参照）

6　最後に

今回の自主課題研究では序盤で時間オートマトンについて論文調査を行い、後半でシステムの検証を行った。しかし、レゴロボットをモデル化する前に、すでにレゴロボットプログラムが完成していたため、検証が難しく、仕様自体のモデル検査とプログラムのモデル検査とが乖離してしまった。この反省を活かし、共同で何かを作るときには、モデル検査はできるだけ早い段階から仕様を分解し、分解した仕様をそれぞれタスクとして実装し、それに対してモデル検査をするべきだということがよくわかった。また、UPPAAL の時相論理式では Untill や Next が実装されていなかったため、通信が成功すれば必ず所定の動作を終了させることができる、といった命題を検証することはできなかった。また、終ってみれば、外部環境に起因する状態の数が膨大になり、時間オートマトンを構築できなかった部分もあり、それがとても残念だ。