EasyChair Preprint

№ 4695

# Formal Theory of Regular Expressions of Maiar DTD Game World.

Frank Appiah

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

December 3, 2020

# Formal Theory of Regular Expressions of Maiar DTD Game World.

FRANK APPIAH.

KING'S COLLEGE LONDON, SCHOOL OF ENGINEERING, LONDON, ENGLAND, UNITED KINGDOM.

appiahnsiahfrank@gmail.com. | frank.appiah@kcl.ac.uk

**Extended Abstract**[+]. This thesis is based on formal theory of grammar focusing on regular expressions of Maiar[3] game world that seems to be in DTD. It describes how regular expressions are used to define the document types and element types.

**Keywords**. document types, formal theory, grammar, regular expression .

**Year of Study: 2017** **Year of Publication: 2020**

**1** ***AFFILIATE. UNIVERSITY OF LONDON, KING'S COLLEGE LONDON, DEPARTMENT OF INFORMATICS, LONDON, UK.**

## 1 INTRODUCTION

In grammar analytic, here is about regular expressions[2] of a game world[1] defined in document type definitions[2]. A schema defines a document type or class of documents by imposing a set of constraints on document instances. This thesis postulates that regular expressions in document of type, *map* introduces the whole content of

the game world. The unified formalism for both document type definition and element type definition is sensed and provided.

It is achieved by new substitution operation into regular expressions. This operation replaces the powerset operator and makes regular expressions equivalent to regular grammar. The basic composites of a regular expression are as follows:

1. *Variable*: It denotes an arbitrary expression. Y is a variable.
2. *Tagged Expression*: It is also denoted as m[Y].
3. *Empty Sequence*: It is denoted as ().

A composition of the three composite elements need three simple operators:

- *Concatenation*: X, Y denotes the concatenation of the expressions X and Y.
- *Union*: X | Y denotes that a pattern can consist of either X or Y. and
- *Substitution*: is an expression enclosed by curly brackets and decorated label (l).

## 2 WORLD COMPOSITION

The regular expression notations of a game world as represented as a Document Type Definition schema[2] is as shown below:

#### 2.1 Regular Map Expression

The three basic notations of regular map expression are:

- Variables : `exit, global, rooms, entry`.
- **Tagged Expression**: It is denoted as a map tag:
  `map[exit, global, rooms, entry]`

#### 2.2 Regular Rooms Expression

The basic notations of regular rooms expression are:

- Variable: `room`
- **Tagged Expression**: A rooms tag is denoted as:
  `rooms[room*]`

#### 2.3 Regular Room Expression

The basic notations of regular room expression are:

- Variables: `id, name, commands, directions, items, look, intro`
- **Tagged Expression**: A room tag is denoted as:
  `room[id, name, commands*, directions*, items*, look*, intro*]`

#### 2.4 Regular Intro Expression

The basic notations of regular intro expression are:

- Variable: `message`
- **Tagged Expression**: A intro tag is denoted as:
  `intro[message]`

#### 2.5 Regular Look Expression

The basic notations of regular Look expression are:

- Variable: `description`
- **Tagged Expression**: A Look tag is denoted as: `Look[description]`

#### 2.6 Regular Items Expression

The basic notations of regular items expression are:

- Variable: `item`
- **Tagged Expression**: A items tag is denoted as: `items[item*]`

#### 2.7 Regular Item Expression

The basic notations of regular item expression are:

- Variable: `name`
- **Tagged Expression**: A item tag is denoted as: `item[name]`

#### 2.8 Regular Directions Expression

The basic notations of regular directions expression are:

- Variable: `north, south, east, west`
- **Tagged Expression**: A directions tag is denoted as: `directions[north, south, east, west]`

#### 2.9 Regular North Expression

The basic notations of regular north expression are:

- Variable: `roomId`
- **Tagged Expression**: A north tag is denoted as: `north[roomId]`

#### 2.10 Regular South Expression

The basic notations of regular south expression are:

- Variable: `roomId`
- **Tagged Expression**: A south tag is denoted as: `south[roomId]`

#### 2.11 Regular East Expression

The basic notations of regular east expression are:

- Variable: `roomId`
- T**agged Expression**: A east tag is denoted as: `east[roomId]`

#### 2.12 Regular West Expression

The basic notations of regular west expression are:

- Variable: `roomId`
- **Tagged Expression**: A west tag is denoted as: `west[roomId]`

#### 2.13 Regular Commands Expression

The basic notations of regular commands expression are:

- Variable: `command`
- **Tagged Expression**: A commands tag is denoted as: `commands[command*]`

#### 2.14 Regular Command Expression

The basic notations of regular command expression are:

- Variable: `name, action`
- **Tagged Expression**: A command tag is denoted as: `command[name, action*]`

#### 2.15 Regular Action Expression

The basic notations of regular action expression are:

- Variable: `requirement, effect`
- **Tagged Expression**: A action tag is denoted as: `action[requirement*, effect*]`

#### 2.16 Regular Effect Expression

The basic notations of regular effect expression are:

- Variable: `operator, value, parameter`
- **Tagged Expression**: A effect tag is denoted as: `effect[operator, value, parameter]`

#### 2.17 Regular Requirement Expression

The basic notations of regular requirement expression are:

- Variables: `notSatisfied, satisfied, value, parameter`
- **Tagged Expression**: A requirement tag is denoted as: `requirement[value,parameter,notSatisfied, satisfied]`

#### 2.18 Regular Notsatisfied Expression

The basic notations of regular Notsatisfied expression are:

- Variable: `action`
- **Tagged Expression**: A Notsatisfied tag is denoted as: `Notsatisfied[action*]`

#### 2.18 Regular satisfied Expression

The basic notations of regular satisfied expression are:

- Variable: `action`
- **Tagged Expression**: A satisfied tag is denoted as: `satisfied[action*]`

#### 2.19 Regular Global Expression

The basic notations of regular global expression are:

- Variable: `command`
- **Tagged Expression**: A global tag is denoted as: `global[command*]`

#### 2.20 Regular Exit Expression

The basic notations of regular exit expression are:

- Variable: `id, room`
- **Tagged Expression**: A exit tag is denoted as:
  `exit[id, room]`

## 4 CONCLUSION

This section concludes work on grammar analysis on document type definition with regular expressions. In all, over 20 regular expressions were analyzed to generate tagged expressions for each element of map document content element. This forms part of one of the formal thesis on world grammar of Maiar game DTD based on regular expression notation.

**Compliance with Ethical Standards:**

(In case of funding) Funding: This is research is funded by King's Alumni Group, School of Engineering with postgraduate grant with ISAreference grant number: 204424 20821845.

**Conflict of Interest:**

Author, Dr. Frank Appiah declares that he has no conflict of interest .

## REFERENCES

1. Appiah F. (2012/13), Design and Implementation of a Graph-based Game, KCL School of Engineering, PhD Dissertation,
2. Berthold B.(2003). Modeling Business Objects With XML Schema. Morgan Kauffman, Elsevier Publishers.
3. Maiar(2020), Macmillan School Dictionary, Accessed January, 2020.