EasyChair Preprint

№ 4813

A Transitive Rule of a Life Cycle of Game Actions and Effects.

Frank Appiah

December 28, 2020

A TRANSITIVE RULE OF A LIFE CYCLE OF GAME ACTIONS AND EFFECTS.

FRANK APPIAH.

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

frank.appiah@kcl.ac.uk

appiahnsiahfrank@gmail.com.

Extended Abstract[+]. This research focuses on transitive rules of a transition function and machine processing by implication[6] symbol. A transition function on a game is described here from [1].

Keywords. computational, game, lifecycle, remote, command, transitive, rules.

Year of Study: 2016 **Year of Publication: 2020**

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

1 INTRODUCTION

Illustration 1 is a state transition diagram[2] of the Game Automaton(G***A***)[2,3,4] called GM_1. It has five states labeled *GUI, CP, PTA , PDE and PPA* and five conditions labeled *PV, VE, PU, TR and EA*. The start state of WFA is GUI and it is normally indicated by a pointing arrow to the state GUI[5]. The accept state is the *GUI* state and it is normally indicated by double circle/round-shape around the state.

The arrows moving from one state to another is called transitions. When the automaton receives an input string {*GUI, CP, PTA , PDE, PPA*}, it processes that string and produces an output.

Life Cycle of Remote Command

The output is either accept or reject. The processing begins in **GM_1** start state. The automaton receives the symbols from the input string one by one from left to right. After playing the symbols, GM_1 moves from one state to another along the transition that has symbol as its label. When it plays the last symbol now it is in the accept state. The processing of GM_1 as follows:

1. start in state GUI;
2. play PU, follow transition from GUI to CP;
3. play TR, follow transition from GUI to PTA;
4. play TR, follow transition from CP to GUI;
5. accept because GM_1 is in accept state GUI;

6. play TR, follow transition from PTA to PPA;
7. play VE, follow transition from PTA to GUI;
8. accept because GM_1 is in accept state GUI;
9. play EA, follow transition from PTA to PDE;
10. play VE, follow transition from PDE to PTA;
11. play EA, follow transition from PPA to PTA.

The machine processing(MP) of GM will now be represented by transitive rules. A transitive Rule(TR) has form:

TR: ($A \rightarrow B$)*,

where A, B and C are actions and effects in a game.

1. $start \rightarrow GUI.$, after MP 1.
2. $play \rightarrow PU$.
3. $PU \rightarrow GUI \rightarrow CP$,after MP 2.
4. $play \rightarrow TR$
   $TR \rightarrow GUI \rightarrow PTA$,after MP 3.
5. $play \rightarrow TR$
   $TR \rightarrow CP \rightarrow GUI$,after MP 4.
6. $accept \rightarrow GUI$,after MP 5.
7. $play \rightarrow TR$
   $TR \rightarrow PTA \rightarrow PPA$,after MP 6
8. $play \rightarrow VE$
   $VE \rightarrow PTA \rightarrow GUI$,after MP 7.
9. $play \rightarrow EA$
   $EA \rightarrow PTA \rightarrow PDE$,after MP 9.
10. $play \rightarrow VE$
    $VE \rightarrow PDE \rightarrow PTA$ after MP 10
11. $play \rightarrow EA$
    $EA \rightarrow PPA \rightarrow PTA$

2 TRANSITION FUNCTION AND RULES

The transition function[2, 3] is used to define the rules of moving. The notation of the transition function is δ(state, input)=state. The transition functions for remote commanding are as follows:

1. δ(GUI, PU) = CP
2. δ(GUI, TR) = PTA
3. δ(CP, TR) = GUI
4. δ(GUI, PU) = CP
5. δ(PTA, TR) = PPA
6. δ(PTA, TR) = CP
7. δ(PTA, VE) = GUI
8. δ(PTA, EA) = PDE
9. δ(PDE, VE) = PTA
10. δ(PPA, EA) = PDA.

The transitive form will be altered to make space for inputs:

TR: $((A, B)\rightarrow(C, D)^{0/1}\rightarrow D)$.

1. $(GUI, PU)\rightarrow(PTA, TR)\rightarrow CP$
2. $(GUI, TR)\rightarrow PTA$
3. $(CP, TR)\rightarrow(PTA, VE)\rightarrow GUI$
4. $(GUI, PU)\rightarrow CP$
5. $(PTA, TR)\rightarrow PPA$
6. $(PTA, TR)\rightarrow CP$
7. $(PTA, VE)\rightarrow GUI$

8. $(PTA\,,\,EA) \rightarrow PDE$
9. $(PDE\,,\,VE) \rightarrow PTA$
10. $(PPA\,,\,EA) \rightarrow PDA$.

3 CONCLUSION

This research is about transitive modeling of game automaton developed from earlier research in [1]. Here, there is an enumeration of transitive rule form in two different of models. The first is based on the machine processing statements and second is based on transition functions that gives the rules of movement in an automaton[3]. There is about 21 transitive rules only in this short paper.

Compliance with Ethical Standards

(In case of funding) Funding: This is research is funded by King's Alumni Group, Association of Engineering 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. (2020). A Computational Model of Life Cycle of Game Actions and Effects. KCL School of Engineering.
2. Michael Sipser(1997). Introduction to Theory of Computations. PWS Publishing Company.
3. HopCroft, J. E. and Ulmann, J. D(1979). Introduction to Automata theory, Languages and Computation. Addison Wesley.
4. Roche, E. and Schabes, Y(1997). Finite-State Language Processing. MIT Press.
5. GUI (2020). Java AWT, Oracle Corporation USA.
6. Enderton H. B (1972). A mathematical Introduction to Logic. Academic Press.