*1. Program → Start-Symbols ClassDeclaration End-Symbols*

*2. Start-Symbols → @ | ^*

*3. End-Symbols → $ | #*

*4. ClassDeclaration → Type ID ClassDeclaration’*

*4.1. ClassDeclaration’ → { Class\_Implementation } | Infer { Class\_Implementation }*

*5. Class\_Implementation→ Type ID Temp Class\_Implementation | Comment Class\_Implementation | require\_command Class\_Implementation| Func \_Call Class\_Implementation |em*

*5.1. Temp → Method\_Decl | ID\_List Variable\_Decl’*

*6. Method\_Decl → Func Decl Method\_Decl’*

*6.1. Method\_Decl’ → ; | { Variable\_Decl Statements }*

*7. Func Decl → (ParameterList)*

*8. Type →* Ipok |Sipok |Craf |Sequence |Ipokf |Sipokf |Valueless |Rational

*9. ParameterList → em| None | Non-Empty List*

*10. Non-Empty List→ Type ID Non-Empty List’*

*10.1. Non-Empty List’ → , Type ID Non-Empty List’ | em*

*11. Variable\_Decl → em | ID\_List Variable\_Decl’*

*11.1. Variable\_Decl’ → ; Variable\_Decl | [ID] ; Variable\_Decl*

*12. ID\_List → ID\_List’*

*12.1. ID\_List’ → , ID ID\_List’ | em*

*13. Statements → em | Statement Statements*

*14. Statement → Assignment | If \_Statement | However \_Statement | when\_Statement | Respondwith \_ Statement | Endthis \_Statement*

*15. Assignment → Type ID Variable\_Decl = Expression;*

*16. Func\_Call → ID ( Argument\_List ) ;*

*17. Argument\_List → em | NonEmpty\_Argument\_List*

*18. NonEmpty\_Argument\_List → Expression NonEmpty\_Argument\_List’*

*18.1. NonEmpty\_Argument\_List’ → , Expression NonEmpty\_Argument\_List’ | em*

*19. Block Statements→{ Statements }*

*20. If\_Statement→ if (Condition \_Expression) Block Statements If\_Statement’*

*20.1. If\_Statement’ → else Block Statements | em*

*21. Condition\_Expression→ Condition Condition\_Expression’*

*21.1. Condition\_Expression’ → Condition\_Op Condition | em*

*22. Condition\_Op → && | ||*

*23. Condition → Expression Comparison\_Op Expression*

*24. Comparison\_Op → == | != | Comparison\_Op’*

*24.1. Comparison\_Op’ → > Comparison\_Op’’ | < Comparison\_Op’’*

*24.2. Comparison\_Op’’ → = | em*

*25. However\_Statement → However (Condition\_Expression) Block Statements*

*26. when\_Statement → when* ***(*** *expression* ***;*** *expression* ***;*** *expression* ***)*** *Block Statements*

*27. Respondwith\_Statement→ Respondwith Expression ; | return ID ;*

*28. Endthis\_Statement→ Endthis;*

*29. Expression → Term Expression’*

*29.1. Expression’ → Add\_Op Term Expression’ | em*

*30. Add\_Op → + | -*

*31. Term → Factor Term’*

*31.1. Term’ → Multi\_Op Factor Term’ | em*

*32. Mul\_Op→\* | /*

*33. Factor → ID| Number*

*34. Comment → </ ID /> | \*\*\*ID*

*35. Require\_command →Require ( F\_name . ID );*

*36. F\_name → ID*