Eli Fux
Jacob Sobolev

Assignment 2 - Additional Tasks

(1) Updated Grammer (Eliminating Common Prefixes & Left Recursion)

PROGRAM → TASK_DEFINITIONS; **parbegin** TASK_LIST **parend**

TASK_DEFINITIONS → TASK_DEFINITION TASK_DEFINITIONS'
TASK_DEFINITIONS' → ε | ; TASK_DEFINITION TASK_DEFINITIONS'

TASK_DEFINITION → **task** id **begin** DECLARATIONS **{** COMMANDS **}**
end

DECLARATIONS → DECLARATION | DECLARATION; DECLARATIONS
DECLARATION → **integer** id | **real** id

TASK_LIST → task_id TASK_LIST'
TASK_LIST' → ε | || task_id TASK_LIST'

COMMANDS → COMMAND COMMANDS'
COMMANDS' → ε | ; COMMAND COMMANDS'

COMMAND → id = EXPRESSION |
do COMMANDS **until** CONDITION **od** |
send task_id . signal_id (PARAM_LIST) |
accept signal_id (DECLARATIONS) |
begin DECLARATIONS **{** COMMANDS **} end**

PARAM_LIST → EXPRESSION | EXPRESSION, PARAM_LIST
EXPRESSION → int_num | real_num | id | id binary_ar_op EXPRESSION
CONDITION → (id rel_op id)

Eli Fux
Jacob Sobolev

(2) Summing Table (First / Follow / Nullable)

Expression	***First***	***Follow***	***Nullable***
PROGRAM	First(TASK_DEFINITIONS)	{ }	
TASK_DEFINITIONS	First(TASK_DEFINITION) ∪ First(TASK_DEFINITIONS')	{ ; , **parbegin }**	
TASK_DEFINITIONS'	{ε} ∪First(TASK_DEFINITION)	Follow(TASK_DEFINITIONS)	+
TASK_DEFINITION	**{ task** id **begin }**	First(TASK_DEFINITIONS')	
DECLARATIONS	First(DECLARATION)	{ {,) }	
DECLARATION	{ **integer** id , **real** id }	{ ; } ∪ First(DECLARATIONS)	
TASK_LIST	{ task_id } ∪ First(TASK_LIST')	{ parend }	
TASK_LIST'	{ε , || task_id }	Follow(TASK_LIST)	+
COMMANDS	First(COMMANDS')	{ until, } }	
COMMANDS'	{ε , ; , } ∪ First(COMMAND)	Follow(COMMANDS)	+
COMMAND	{ id =, **do** , **send** task_id . signal_id, **accept** signal_id , **begin** }	Follow(COMMANDS) ∪ { ;}	
PARAM_LIST	First(EXPRESSION)∪{ , }	{) }	
EXPRESSION	{ int_num , real_num , id ,id binary_ar_op }	{ comma,), } }	
CONDITION	{ (id rel_op id) }	{ od }	