APPENDIX A

Addition Productions*

1	**Start-Problem push goal check declarative memory**	
	IF **AND**	- the goal is to do an addition problem - status is start and column is empty
	THEN	- set the current goal status to "after checking declarative memory" - push a new subgoal chunktype check_declarative to retrieve the complete answer to the addition from memory
2	**Check addition problem in memory**	
	IF **AND**	- the goal is to retrieve the answer from memory - the answer to the problem is found in memory
	THEN	- substitute the answer in a chunk and put it in the communication buffer - remove the subgoal - set the upper goal status to found
3	**Found and clean**	
	IF	- the goal has a status of found
	THEN	- remove the top goal *(end cycle solved problem)*
4	**After check declarative memory start problem**	
	IF	- the goal has status "after checking declarative memory
	THEN	- set goal status to start *and* column = 1
5	**Addition read goal**	
	IF	- status equals start and column == 1 Remember temporary variables col1number1 = 5 col1number2 = 7 nrColumns = 4
	THEN	- set goal status to wait for subgoal - transfer nrColumns=4 to retrievalbuffer - push subgoal chunktype add_column number1 5 number2 7 column 1 carry Zero

*Productions of the multi-column addition problem in English Language

Appendix A: Addition Productions

6	**Add numbers no carry**	
	IF	- the goal is to add two number with no carry number1 = 5 number2 = 7 carry = Zero
	AND	- if the summation fact (5 + 7=?) can be retrieved from memory
	THEN	- set the answer of add_column to the answer of the summation fact (12) - set the status(carry) on waiting
7	**Add numbers process carry**	
	IF	- the goal is to add two numbers with a carry number1 = 5 number2 = 7 carry = one
	AND	- if the summation fact (number1 + carry = newnum1 (5+1 = ?)) can be retrieved from memory
	AND	- if the summation fact (newnum1 + number2 = answer (6+7 = ?) can be retrieved from memory
	THEN	- set the answer of add_column to the answer of the summation fact(13) - set the status(carry) on waiting
8	**Add column answer greater than 9 (creating a carry)**	
	IF	- the goal is to find out if there is a carry after addition number1 = 5 number2 = 7 answer = 12 carry == waiting
	AND	- it is not the last column (column != nrColumns)
	AND	- the summation fact (10 + addend2 = answer (10 + 2 = 12) can be retrieved from memory
	THEN	- set the answer to addend2 (2) and carry to one - remove nrColumns from retrieval buffer - put current goal in retrieval buffer - pop the goal with the add_column
9	**Add column answer smaller than 10 (no carry)**	
	IF	- the goal is to find out if there is no carry after addition number1 = 2 number2 = 7 answer = 9 carry == waiting
	AND	- it is not the last column (column != nrColumns)
	AND	- the summation fact (0 + addend2 = answer (0 + 9 = 9) can be retrieved from memory
	THEN	- set the answer to addend2 (9) and carry to zero - remove nrColumns from retrieval buffer - put current goal in retrieval buffer - pop the goal with the add_column *(back at top goal == addition_problem)*
10	**Add column last column**	
	IF	the goal is to add the last column of the addition problem carry == waiting
	AND	- it is the last column (column equals nrColumns)
	THEN	- set carry to "finished" - remove nrColumns from retrieval buffer - put current goal in retrieval buffer - pop the current goal with the add_column

11	**Addition problem write column_abstract** ***(depending on column number)***	
	IF	the goal is to write the answer to the goal chunk column !=null nrColumns != column column is 1...4 (column number) col(column number)answer == null
	AND	- in the retrieval buffer is a chunk of add_column
	AND	- retrieve (1+ column number = new column number (e.g. 2 → 3) from memory
	AND	- read the new numbers of the new column
		- write the answer to the col(column number)answer slot of the goal
	THEN	- remove the chunk from retrieval buffer and put nrColumns in retrieval buffer
		- push the new subgoal with a new add_column of the new column
12	**Addition problem write last column**	
	IF	- the goal is to write the answer of the last column to the goal chunk nrColumns equals column number status equals not communicate
	AND	- if there is an add_column chunk in the retrieval buffer
		- write the answer to the last column
	THEN	- set status to communicate
		- remove the chunk from the retrieval buffer
13	**Addition problem communicate**	
	IF	- status is communicate
		- put goal chunk in communication buffer
	THEN	- put the chunk in declarative memory
		- pop the goal : problem solved