Computable logic through connections without measurements

John Lilley

ABSTRACT

A computable logic that arises from how connections are made and/or broken over time without measurements. This is a complete description of everything about this logic.

INTRODUCTION

This is a unique and unknown form of logic. I believe it is the most abstract logic possible. The problem is that it is not Turing Complete, it requires an operator.

MATERIALS AND METHODS

The models I created using the Quakeworld engine and its derivatives like Counter-Strike use one system scripting command to perform logic, "alias" that only allows you to create a command or rewrite a created command to execute a string of commands. In my models that string only calls other alias created commands, basically looping around itself in essence only using connections.

The other two commands I use are for input and output. The two other system commands are: "bind" which simply connects a key press to a command, and "echo" which outputs text to the console.

 I have built if-thens, sophisticated do-whiles, a randomizer, a relational database and math emulators.

RESULTS

Here is a simple example of emulating mathematics in an abacus-like calculator that only adds and subtracts between -20 to 20:

Text for calculator_simple.cfg

//Copyright All Rights Reserved 2025
//this script was built in Counter-Strike GO but will work in any QuakeWorld engine derivative game, including the original Quake.
//this script binds the KeyPad keys 0 through 9 along with - + and Enter
//This is limited to sums of -20 to 20, each number needs to be mapped and limits the system. This is also limited to single digit input.
//Starts in add mode. You can enter numbers in add or subtract mode without repeatedly hitting - or +
//list of system commands used (already built into the QuakeWorld engine):
//bind - input (for linking keys to commands - all of which in this example are user made commands)
//echo - output: for text output to the console (make sure you have the console turned on in settings, and use ~ in game to bring it down)
//alias - the connection creator: for linking a string of commands, both user and system commands, together into one new callable user command
//all of the rest of the commands in this particular script are user created, here, with alias

//input binds
//bind physical KeyPad numbers to adding/subtracting function
bind KP_END calc1
bind KP_DOWNARROW calc2
bind KP_PGDN calc3
bind KP_LEFTARROW calc4
bind KP_5 calc5
bind KP_RIGHTARROW calc6
bind KP_HOME calc7

```
bind KP_UPARROW calc8
bind KP_PGUP calc9
bind KP_INS calc0
bind KP_MINUS minus //bind physical KeyPad key - to the minus mode function
bind KP_PLUS plus //bind physical KeyPad key + to the plus mode function
bind KP_ENTER sum //bind physical KeyPad key Enter to output sum function

//performing of the logic - echo is the output

//increment or decrement based upon mode with input from a number key
alias calc0 "echo 0"
alias calc1 "rotate; echo 1"
alias calc2 "rotate; rotate; echo 2"
alias calc3 "rotate; rotate; rotate; echo 3"
alias calc4 "rotate; rotate; rotate; rotate; echo 4"
alias calc5 "rotate; rotate; rotate; rotate; rotate; echo 5"
alias calc6 "rotate; rotate; rotate; rotate; rotate; rotate; echo 6"
alias calc7 "rotate; rotate; rotate; rotate; rotate; rotate; rotate; echo 7"
alias calc8 "rotate; rotate; rotate; rotate; rotate; rotate; rotate; rotate; echo 8"
alias calc9 "rotate; rotate; rotate; rotate; rotate; rotate; rotate; rotate; rotate; echo 9"

//set mode
alias plus "alias rotate sumrotateup; echo PLUS"
alias minus "alias rotate sumrotatedown; echo MINUS"

//map of rotation
alias sumnegative20 "alias sum sumn20; alias sumrotatedown sumnegative20; alias sumrotateup sumnegative19"
//sumnegative20 bounces back to itself in sumrotatedown rather than going to an unmapped -21
alias sumnegative19 "alias sum sumn19; alias sumrotatedown sumnegative20; alias sumrotateup sumnegative18"
alias sumnegative18 "alias sum sumn18; alias sumrotatedown sumnegative19; alias sumrotateup sumnegative17"
alias sumnegative17 "alias sum sumn17; alias sumrotatedown sumnegative18; alias sumrotateup sumnegative16"
alias sumnegative16 "alias sum sumn16; alias sumrotatedown sumnegative17; alias sumrotateup sumnegative15"
alias sumnegative15 "alias sum sumn15; alias sumrotatedown sumnegative16; alias sumrotateup sumnegative14"
alias sumnegative14 "alias sum sumn14; alias sumrotatedown sumnegative15; alias sumrotateup sumnegative13"
alias sumnegative13 "alias sum sumn13; alias sumrotatedown sumnegative14; alias sumrotateup sumnegative12"
alias sumnegative12 "alias sum sumn12; alias sumrotatedown sumnegative13; alias sumrotateup sumnegative11"
alias sumnegative11 "alias sum sumn11; alias sumrotatedown sumnegative12; alias sumrotateup sumnegative10"
alias sumnegative10 "alias sum sumn10; alias sumrotatedown sumnegative11; alias sumrotateup sumnegative9"
alias sumnegative9 "alias sum sumn9; alias sumrotatedown sumnegative10; alias sumrotateup sumnegative8"
alias sumnegative8 "alias sum sumn8; alias sumrotatedown sumnegative9; alias sumrotateup sumnegative7"
alias sumnegative7 "alias sum sumn7; alias sumrotatedown sumnegative8; alias sumrotateup sumnegative6"
alias sumnegative6 "alias sum sumn6; alias sumrotatedown sumnegative7; alias sumrotateup sumnegative5"
alias sumnegative5 "alias sum sumn5; alias sumrotatedown sumnegative6; alias sumrotateup sumnegative4"
alias sumnegative4 "alias sum sumn4; alias sumrotatedown sumnegative5; alias sumrotateup sumnegative3"
alias sumnegative3 "alias sum sumn3; alias sumrotatedown sumnegative4; alias sumrotateup sumnegative2"
alias sumnegative2 "alias sum sumn2; alias sumrotatedown sumnegative3; alias sumrotateup sumnegative1"
alias sumnegative1 "alias sum sumn1; alias sumrotatedown sumnegative2; alias sumrotateup sumzero"

alias sumzero "alias sum sum0; alias sumrotatedown sumnegative1; alias sumrotateup sumpositive1"

alias sumpositive1 "alias sum sump1; alias sumrotatedown sumzero; alias sumrotateup sumpositive2"
alias sumpositive2 "alias sum sump2; alias sumrotatedown sumpositive1; alias sumrotateup sumpositive3"
alias sumpositive3 "alias sum sump3; alias sumrotatedown sumpositive2; alias sumrotateup sumpositive4"
alias sumpositive4 "alias sum sump4; alias sumrotatedown sumpositive3; alias sumrotateup sumpositive5"
alias sumpositive5 "alias sum sump5; alias sumrotatedown sumpositive4; alias sumrotateup sumpositive6"
alias sumpositive6 "alias sum sump6; alias sumrotatedown sumpositive5; alias sumrotateup sumpositive7"
alias sumpositive7 "alias sum sump7; alias sumrotatedown sumpositive6; alias sumrotateup sumpositive8"
alias sumpositive8 "alias sum sump8; alias sumrotatedown sumpositive7; alias sumrotateup sumpositive9"
```

alias sumpositive9 "alias sum sump9; alias sumrotatedown sumpositive8; alias sumrotateup sumpositive10"
alias sumpositive10 "alias sum sump10; alias sumrotatedown sumpositive9; alias sumrotateup sumpositive11"
alias sumpositive11 "alias sum sump11; alias sumrotatedown sumpositive10; alias sumrotateup sumpositive12"
alias sumpositive12 "alias sum sump12; alias sumrotatedown sumpositive11; alias sumrotateup sumpositive13"
alias sumpositive13 "alias sum sump13; alias sumrotatedown sumpositive12; alias sumrotateup sumpositive14"
alias sumpositive14 "alias sum sump14; alias sumrotatedown sumpositive13; alias sumrotateup sumpositive15"
alias sumpositive15 "alias sum sump15; alias sumrotatedown sumpositive14; alias sumrotateup sumpositive16"
alias sumpositive16 "alias sum sump16; alias sumrotatedown sumpositive15; alias sumrotateup sumpositive17"
alias sumpositive17 "alias sum sump17; alias sumrotatedown sumpositive16; alias sumrotateup sumpositive18"
alias sumpositive18 "alias sum sump18; alias sumrotatedown sumpositive17; alias sumrotateup sumpositive19"
alias sumpositive19 "alias sum sump19; alias sumrotatedown sumpositive18; alias sumrotateup sumpositive20"
alias sumpositive20 "alias sum sump20; alias sumrotatedown sumpositive19; alias sumrotateup sumpositive20"
//sumpositive20 bounces back to itself in sumrotateup rather than going to an unmapped 21

//end of logic

//initialize the system for first use
sumzero //initialize position of sum
plus //initialize for addition

//output echos
alias sumn20 "echo sum -20"
alias sumn19 "echo sum -19"
alias sumn18 "echo sum -18"
alias sumn17 "echo sum -17"
alias sumn16 "echo sum -16"
alias sumn15 "echo sum -15"
alias sumn14 "echo sum -14"
alias sumn13 "echo sum -13"
alias sumn12 "echo sum -12"
alias sumn11 "echo sum -11"
alias sumn10 "echo sum -10"
alias sumn9 "echo sum -9"
alias sumn8 "echo sum -8"
alias sumn7 "echo sum -7"
alias sumn6 "echo sum -6"
alias sumn5 "echo sum -5"
alias sumn4 "echo sum -4"
alias sumn3 "echo sum -3"
alias sumn2 "echo sum -2"
alias sumn1 "echo sum -1"
alias sum0 "echo sum 0"
alias sump1 "echo sum 1"
alias sump2 "echo sum 2"
alias sump3 "echo sum 3"
alias sump4 "echo sum 4"
alias sump5 "echo sum 5"
alias sump6 "echo sum 6"
alias sump7 "echo sum 7"
alias sump8 "echo sum 8"
alias sump9 "echo sum 9"
alias sump10 "echo sum 10"
alias sump11 "echo sum 11"
alias sump12 "echo sum 12"
alias sump13 "echo sum 13"
alias sump14 "echo sum 14"
alias sump15 "echo sum 15"
alias sump16 "echo sum 16"

```
alias sump17 "echo sum 17"
alias sump18 "echo sum 18"
alias sump19 "echo sum 19"
alias sump20 "echo sum 20"
```

End of calculator_simple.cfg

DISCUSSION

This example is undeniably performing computational logic. This form of logic is not in the human lexicon. Ask any LLM this question: "Can you name a form of logic that is formed by making and/or breaking connections over time without using measurements, with this being a complete description of the logic?" Remember LLMs "hallucinate", make sure you investigate the answer.

I call this Logic Geometry.

Is the computer the fabric of space-time, through entanglement alone?

ACKNOWLEDGEMENTS

If it was not for John Carmack's programming of Quake, I would not have stumbled on this.