QuantumScheme 1.0

A quantum programming language

# The implementation is based on JScheme and thus contains the basic feature set of the Scheme programming language. Although there are few predefined functions the extensible nature of Lisp allows the user to easily construct new functionality from the provided operations.

# 

# 1 Quantum gates (constants)

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| hadamard | Unary gate | Entangles the qubit states. |
| paulix | Unary gate | Rotates qubit around hypercomplex x. |
| pauliy | Unary gate | Rotates qubit around hypercomplex y. |
| pauliz | Unary gate | Rotates qubit around hypercomplex z. |
| not | Unary gate | Equivalent to paulix. |
| sqrtnot | Unary gate | A “half-not” in a sense that when squared, it yields a NOT. |
| cnot | Binary gate. | Inverts the second qubit based on the first. Generally used to entangle qubits. |
| sqrtswap | Binary gate. | “Half-swaps” the qubits. |
| swap | Binary gate. | Swaps the qubits. |
| ccnot | Ternary gate. | Only inverts the third qubit if the first two qubits are active. |
| cswap | Ternary gate. | Only swaps the second two qubits if the first is active. |

# 2 Functions

|  |  |  |
| --- | --- | --- |
| Syntax | Description | Example |
| (print s) | Prints the string s to the console. | (print "Test") |
| (println s) | Prints the line s to the console. | (println "Test") |
| (qsupos l) | Constructs a new superposition with qubits from the given list l of bits. | (qsupos (list 1 0 1 1 1)) |
| (qgate g i s) | Applies the quantum gate g to the qubit number i in the quantum superposition s and yields the result. | (qgate hadamard 0 (qsupos (list 1 0 1))) |
| (collapse s) | Collapses the quantum superposition s to a state of zeroes and ones (which it returns as a list). This function is inherently not pure as it determines its result randomly based off a weighted probability distribution. | (collapse (qgate sqrtswap 0 (qsupos (list 1 0 0 1)))) |