The sample of quantum circuits used for the validation experiment are described in the following table. They are described in QCSR, which is a domain-specific language for describing quantum circuits as a matrix. QCSR specification is found below the sample of quantum circuits.

IDENTIFIER	NAME	QCSR
1	Random Number Generator	[["H","MEASURE"]]
2	Test Circuit 1	[["H",{"CONTROL":3},"H","_","_",{"CONTROL":1},"_",{"CONTROL":2},"_","_","MEASURE"],["H","_","_",{"CONTROL":3},"_","S","H","_",{"CONTROL":2},"_","MEASURE"],["H","_","_","_",{"CONTROL":3},"_","_","T","S","H","MEASURE"],["_",{"ORACLE":2},"_",{"ORACLE":2},{"ORACLE":2}],["_","ORACLE2","_","ORACLE2","ORACLE2"]]
3	Test Circuit 2	[[{"CONTROL":1},{"CONTROL":3},"_","_",{"CONTROL":4},{"CONTROL":1}],["X","_","_",{"CONTROL":3},"_","X",{"CONTROL":3}],["_","_",{"CONTROL":3}],["_","X","X",{"CONTROL":4},"_","_","X"],["_","_","_","X","X"]]
4	Basic Entanglement	[["H",{"CONTROL":1}],["_","X"]]
5	Deutsch-Jozsa Algorithm	[["_","H",{"ORACLE":4},"H","MEASURE"],["_","H","ORACLE2","H","MEASURE"],["_","H","ORACLE2","H","MEASURE"],["X","H","ORACLE2"]]
6	Grover Search	[["H",{"ORACLE":3},"H",{"ORACLE":3},"H","MEASURE"],["H","ORACLE2","H","ORACLE2","H","MEASURE"],["H","ORACLE2","H","ORACLE2","H","MEASURE"]]
7	Quantum Teleportation	[["_","_",{"CONTROL":1},"H","MEASURE","_",{"CONTROL":2}],["H",{"CONTROL":2},"X","_","MEASURE",{"CONTROL":2}],["_","X","_","_","_","X","Z"]]

8	Deutsch Algorithm	[["H", {"ORACLE": 1}, "H", "MEASURE"]]
9	False coin detector	[["_","H",{"ORACLE":8},"H","MEASURE"],["_","H","ORACLE2","H","MEASURE"],["_","H","ORACLE2","H","MEASURE"],["_","H","ORACLE2","H","MEASURE"],["_","H","ORACLE2","H","MEASURE"],["_","H","ORACLE2","H","MEASURE"],["_","H","ORACLE2","H","MEASURE"],["X","H","ORACLE2","H","MEASURE"]]
10	Half Adder	[[{"CONTROL":2},"_",{"CONTROL":1}],["_",{"CONTROL":2},{"CONTROL":3}],["X","X","_","MEASURE"],["_","_","X","MEASURE"]]
11	Decoder 2x4	[["X",{"CONTROL":1},"_",{"CONTROL":1},"X",{"CONTROL":1},"_",{"CONTROL":1}],["X",{"CONTROL":2},"X",{"CONTROL":3},"X",{"CONTROL":4},"X",{"CONTROL":5}],["_","X"],["_","_","_","X"],["_","_","_","_","_","X"],["_","_","_","_","_","_","_","X"]]
12	Swap Gate Circuit	[[{"CONTROL":1},"X",{"CONTROL":1}],["X",{"CONTROL":0},"X"]]
13	1-bit Quantum String Comparator	[["_",{"CONTROL":1},"X",{"CONTROL":1},"X"],["X",{"CONTROL":2},"X",{"CONTROL":3}],["_","X"],["_","_","_","X"]]
14	Shor Algorithm	[["H","_",{"CONTROL":4},"_","_","_",{"ORACLE":4}],["H","_","_",{"CONTROL":4},"_","_","ORACLE2"],["H","_","_","_",{"CONTROL":4},"_","ORACLE2"],["H","_","_","_","_",{"CONTROL":4},"ORACLE2"],["_","_",{"ORACLE":4},{"ORACLE":4},{"ORACLE":4},{"ORACLE":4}],["_","_","ORACLE2","ORACLE2","ORACLE2","ORACLE2"],["_","_","ORACLE2","ORACLE2","ORACL

		E2","ORACLE2"],["_","_","ORACLE2","ORACLE2","ORACLE2","ORACLE2"]]
15	Swap Gate Circuit no CNOTS	[["H",{"CONTROL":1},"H","Z","H",{"CONTROL":1},"H"],["_","Z","H",{"CONTROL":0},"H","Z"]]
16	W State	[["RY",{"CONTROL":1},"_",{"CONTROL":1},"X"],["_","H",{"CONTROL":2},"X"],["_","_","X"]]
17	GHZ State	[["H",{"CONTROL":1},{"CONTROL":2},{"CONTROL":3},{"CONTROL":4},{"CONTROL":5}],["_","X"],["_","_","X"],["_","_","_","X"],["_","_","_","_","X"],["_","_","_","_","_","X"]]
18	Conmutativity	[["H","_",{"CONTROL":1},"_",{"CONTROL":2}],["H","X","X"],["H","_","_","X","X"]]
19	NOR Gate	[["X",{"CONTROL":1}],["X",{"CONTROL":2}],["_","X","MEASURE"]]
20	OR Gate	[[{"CONTROL":2},"_",{"CONTROL":1}],["_",{"CONTROL":2},{"CONTROL":2}],["X","X","X","MEASURE"]]
21	XOR Gate	[[{"CONTROL":2}],["_",{"CONTROL":2}],["X","X","MEASURE"]]
22	XNOR Gate	[[{"CONTROL":2}],["_",{"CONTROL":2}],["X","X","X","MEASURE"]]
23	AND Gate	[[{"CONTROL":1}],[{"CONTROL":2}],["X","MEASURE"]]
24	NAND Gate	[[{"CONTROL":1}],[{"CONTROL":2}],["X","X","MEASURE"]]
25	Bit Flip Correction	[[{"CONTROL":1},{"CONTROL":2},{"ORACLE":1},{"CONTROL":1},{"CONTROL":2},"MEASURE"],["X","_","_","X"],["_","X","_","_","X"]]

26	RNG in stairs	[["H","_","_","_","_","_","_","_","_","MEASURE"],["_","H","_","_","_","_","_","_","MEASURE"],["_","_","H","_","_","_","_","MEASURE"],["_","_","_","H","_","_","MEASURE"],["_","_","_","_","H","MEASURE"]]
27	2-bit Quantum String Comparator	[[{"ORACLE":4}],["ORACLE2"],["ORACLE2","X",{"CONTROL":3},"X","X"],["ORACLE2","X",{"CONTROL":4},"X","_","X"],["_","_","X","_",{"CONTROL":2},{"CONTROL":3}],[{"ORACLE":4}],["ORACLE2"],["ORACLE2","_","_","_",{"CONTROL":4}],["ORACLE2","_","_","_","_",{"CONTROL":4}]]
28	Anti–Control (2 bits)	[["X",{"CONTROL":1},"X"],["X",{"CONTROL":2},"X"],["_","X"]]
29	Quantum Phase Estimation (QPE)	[["X","T","T","T","T","T","T","T"],["H","_","_","_",{"CONTROL":0},{"CONTROL":0},{"CONTROL":0},{"CONTROL":0},{"ORACLE":3},"MEASURE"],["H","_",{"CONTROL":0},{"CONTROL":0},"_","_","_","_","ORACLE2","MEASURE"],["H",{"CONTROL":0},"_","_","_","_","_","_","ORACLE2","MEASURE"]]
30	Quantum Random Access Memory - Address = 2 bits (qRAM)	[["H","X",{"CONTROL":1},"X","X",{"CONTROL":1},"X","_",{"CONTROL":1},"_","X",{"CONTROL":1},"X",{"CONTROL":1},{"CONTROL":1}],["H","X",{"CONTROL":3},"X","X",{"CONTROL":4},"X","X",{"CONTROL":2},"X","_",{"CONTROL":3},"_",{"CONTROL":2},{"CONTROL":4}],["_","_","_","_","_","_","_","_","X","_","_","_","_","X"],["_","_","X","_","_","_","_","_","_","_","_","X"],["_","_","_","_","_","X","_","_","_","_","_","_","_","_","X"]]
31	Superdense Coding	[["_","X","_","_","_","X","_","

	(Message = ‘10’)	MEASURE"],["H",{"CONTROL":0},"_","Z","_",{"CONTROL":0},"H","MEASURE"]]
32	Bernstein-Vazirani Algorithm	[["H","_","_",{"CONTROL":3},"_","_","H","MEASURE"],["H","_","_","_","I","_","H","MEASURE"],["H","_","_","_",{"CONTROL":3},"_","H","MEASURE"],["H","Z","_","X","X"]]
33	Full Adder	[["X",{"CONTROL":3},"_","_",{"CONTROL":1},{"CONTROL":2}],["_","_",{"CONTROL":3},"_",{"CONTROL":4},"_","_",{"CONTROL":2}],["_","_","_",{"CONTROL":3},"_",{"CONTROL":4},"_",{"CONTROL":4}],["_","X","X","X","_","_","MEASURE"],["_","_","_","_","X","X","_","X","MEASURE"]]
34	Quantum Single Value Transformation (QSVT)	[["_","_","RY"],["_",{"ORACLE":3},{"CONTROL":0},{"ORACLE":4}],["_","ORACLE2","_","ORACLE2"],[{"ORACLE":2},"ORACLE2","_","ORACLE2"],["ORACLE2","_","_","ORACLE2"]]
35	Quantum Principal Component Analysis (QPCA)	[[{"CONTROL":1},{"CONTROL":1},"_","MEASURE"],[{"ORACLE":5},{"ORACLE":5},"_","MEASURE"],["ORACLE2","ORACLE2"],["ORACLE2","ORACLE2"],["ORACLE2","ORACLE2","_","_",{"ORACLE":2}],["ORACLE2","ORACLE2","_","_","ORACLE2"]]
36	Quantum HHL	[["_","_","_","_","_","R1"],["H","_","_",{"CONTROL":4},{"ORACLE":3},{"CONTROL":2},{"ORACLE":3},{"CONTROL":4},"_","_","H"],["H","_",{"CONTROL":4},"_","ORACLE2",{"CONTROL":3},"ORACLE2","_",{"CONTROL":4},"_","H"],["H",{"CONTROL":4},"_","_","ORACLE2",{"CONTROL":0},"ORACLE2","_","_",{"CONTROL":4},"H"],["_",{"ORACLE":1},{"ORACLE":1},{"ORACLE":1},"_","_","_",{"ORACL

		E":1},{"ORACLE":1},{"ORACLE":1}],[]]
37	GHZ State v2	[["H",{"CONTROL":1},{"CONTROL":2}],["H","Z","_","H"],["H","_","Z","H"]]
38	Controlled Z identity	[["_",{"CONTROL":1}],["H","X","H"]]
39	Toffoli identity	[[{"CONTROL":2},"_",{"CONTROL":2}],["_",{"CONTROL":2}],["H","Z","H"]]
40	S identity	[["T","T"]]
41	Upside-down CNOT identity	[["H",{"CONTROL":1},"H"],["H","X","H"]]
42	Simon’s Algorithm	[["H",{"ORACLE":4},"_","H","MEASURE"],["H","ORACLE2","_","H","MEASURE"],["_","ORACLE2","MEASURE"],["_","ORACLE2","MEASURE"]]
43	Shift Operation (4D Hypercube)	[["_","_","_","_","_","X"],["_","_","_","_","_","_","_","X"],["_","_","_","_","_","_","_","_","_","X"],["_","_","_","_","_","_","_","_","_","_","_","X"],["H","Z",{"CONTROL":5},"H","X",{"CONTROL":0},"X",{"CONTROL":1},"X",{"CONTROL":2},"X",{"CONTROL":3}],["H","Z","Z","H","X",{"CONTROL":4},"_",{"CONTROL":4},"X",{"CONTROL":4},"_",{"CONTROL":4}]]
44	Quantum Image Processing (2x2 with Greyscale)	[["H",{"CONTROL":2},{"CONTROL":1},"_",{"CONTROL":1},"_","_",{"CONTROL":2},{"CONTROL":1},"_",{"CONTROL":1},"_","X",{"CONTROL":2},{"CONTROL":1},"_",{"CONTROL":1},"_","_",{"CONTROL":2},{"CONTROL":1},"_",{"CONTROL":1},"_","MEASURE"],["H","_","X",{"CONTROL":2},"X",{"CONTROL":2},"X","_","X",{"CONTROL":2},"X",{"CONTROL":2},"X","_","X",{"CONTROL":2},"X",{"CONTROL":2},"X","_","X",{"CONT

		ROL":2},"X",{"CONTROL":2},"MEASURE"],["_","RY","_","RY","_","RY","_","RY","_","RY","_","RY","_","RY","_","RY","_","RY","_","RY","_","RY","MEASURE"]]
45	Grover Diffusor’s Operator (3-qubits)	[["H","X",{"CONTROL":1},"X","H"],["H","X",{"CONTROL":2},"X","H"],["H","X","Z","X","H"]]
46	Phase Kickback Application (2-qubits)	[["H",{"CONTROL":1},"H"],["_",{"ORACLE":1}]]
47	QAOA State Preparation Example	[[{"ORACLE":1},"R1",{"CONTROL":1},"_",{"CONTROL":1},{"CONTROL":3},"_",{"CONTROL":3}],[{"ORACLE":1},"R1","X","RZ","X","_","_","_",{"CONTROL":2},"_",{"CONTROL":2}],[{"ORACLE":1},"R1","_","_","_","_","_","_","X","RZ","X",{"CONTROL":3},"_",{"CONTROL":3}],[{"ORACLE":1},"R1","_","_","_","X","RZ","X","_","_","_","X","RZ","X"]]
48	Quantum Classification Example (2 datapoints & 2 features)	[["H","_","_","_","_","_",{"CONTROL":1},{"CONTROL":2},"_",{"CONTROL":2},"_",{"CONTROL":1},{"CONTROL":2},"_",{"CONTROL":2},"_","X",{"CONTROL":1},{"CONTROL":2},"_",{"CONTROL":2},"_",{"CONTROL":1},{"CONTROL":2},"_",{"CONTROL":2},"_",{"CONTROL":3}],["H",{"CONTROL":2},"_",{"CONTROL":2},"_","X",{"CONTROL":2},"_","_","_","_",{"CONTROL":2},"_","_","_","_","_",{"CONTROL":2},"_","_","_","_",{"CONTROL":2},"_","_","_","_","_","H","MEASURE"],["_","X","RY","X","RY","_","X","X","RY","X","RY","X","X","RY","X","RY","_","X","X","RY","X","RY","X","X","RY","X","RY"],["_","X","_","MEASURE"]]

49	Error correction	[[{"CONTROL":1},{"CONTROL":2},"_","_","_",{"CONTROL":3},"_",{"CONTROL":4},"_","_","X","_","_","MEASURE"],["X","_","_","X","_","_",{"CONTROL":3},"_","_","_","_","X","_","MEASURE"],["_","X","_","_","_","_","_","_",{"CONTROL":4},"_","_","_","X","MEASURE"],["_","_","_","_","_","X","X","_","_","MEASURE"],["_","_","_","_","_","_","_","X","X","MEASURE"]]
50	Separated Control identity	[[{"SWAP":2},"_",{"SWAP":2}],[],["SWAP2","X","SWAP2"],["_",{"CONTROL":2}]]

Bibliographical sources used:

- quantum studio
- qiskit
- microsoft azure quantum
- quantumcomputing.uk repository
- quantumalgorithms.org book
- quantum-inspire

QCSR Specification

1. QCSR Lexicon

The lexical aspects of a formal language are compounded by the lexical components, also known as tokens, which are basic elements of the language that have meaning. In this case, the lexicon, among other aspects, describes the low-level constructions of the language, such as the quantum gates.

However, before describing the lexicon it is necessary to define the alphabet that the formal language uses, as other authors have already done when applying language processors. The alphabet includes all the alphanumeric characters and some other special characters such as commas, brackets and curly braces. The complete description of the used alphabet, expressed in EBNF, which is a notation for describing formal languages defined in the International Organization for Standardization (ISO) Standard 14977:1996, can be found in :

```
letter ::= A | B | C | D | E | F | G | H | I | J | K | L | M
         | N | O |P | Q | R | S | T | U | V | W | X | Y | Z

digit ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

alphanumeric_characters ::= letter | digit

special_characters ::= '{' | '}' | '[' | ']' | ',' | '"'
                     | ':' | '_'

alphabet ::= (alphanumeric_characters | special_characters)
```

Additionally, the lexical description of the proposed formal language is given next:

Token	Lexeme/s	Description
VALID_NUM	0, 1, 103...	The symbol VALID_NUM represents a non-negative integer number.
GATEX	"X"	The symbol GATEX represents the Pauli-X gate (X).
GATEY	"Y"	The symbol GATEY represents the Pauli-Y gate (Y).
GATEZ	"Y"	The symbol GATEZ represents the Pauli-Z gate (Z).
GATEID	"Y"	The symbol GATEID represents the identity gate (I).
GATER1	"R1"	The symbol GATER1 represents the rotation-1 gate (R1).
GATERX	"RX"	The symbol GATERX represents the axis x rotation gate (RX).
GATERY	"RY"	The symbol GATERY represents the axis y rotation gate (RY).
GATERZ	"RZ"	The symbol GATERZ represents the axis z rotation gate (RZ).
GATEH	"H"	The symbol GATEH represents the Hadamard gate (H).
GATES	"S"	The symbol GATES represents the S gate (S).
GATESR	"SR"	The symbol GATESR represents the inverse-S gate (S^{-1}).
GATET	"T"	The symbol GATET represents the T gate (T).
GATETR	"TR"	The symbol GATETR represents the inverse-T gate (T^{-1}).
GATEMEASURE	"TR"	The symbol GATEMEASURE represents the measure gate.
GATESWAP	{"SWAP":4}	The symbol GATESWAP represents the swap gate.
GATECONTROL	{"CONTROL":1}	The symbol GATECONTROL represents a controlled gate.
GATEORACLE	{"ORACLE":2}	The symbol GATEORACLE represents a quantum oracle.
GATENONE	"_"	The symbol GATENONE represents a position in a quantum circuit diagram that has no quantum gates (to allow a matrix representation of the quantum circuit).
GATESWAP2	"SWAP2"	The symbol GATESWAP2 represents the effect of the swap gate on its second target (to allow a matrix representation of the quantum circuit).
GATEORACLE2	"ORACLE2"	The symbol GATEORACLE2 represents the effect of the oracle gate on its other targets whenever its size is higher than 1 (to allow a matrix representation of the quantum circuit).

2. QCSR Syntax

The syntax of QCSR, described in EBNF notation, is found next:

```
circuit ::= '[' [listQubits] ']'
listQubits ::= {qubit ','} qubit
qubit ::= '[' [listGates] ']'
listGates ::= {gate ','} gate
gate ::= GATEX | GATEY | GATEZ | GATEID | GATER1
       | GATERX | GATERY | GATERZ | GATEH | GATES
       | GATESR | GATET | GATETR | GATEMEASURE
       | '{' GATESWAP ':' VALID_NUM '}'
       | '{' GATECONTROL ':' VALID_NUM '}'
       | '{' GATEORACLE ':' VALID_NUM '}'
       | GATENONE | GATESWAP2 | GATEORACLE2
```

3. QCSR Semantics

The semantics must also be specified to conclude the design of the formal language. Each of the productions of the grammar described previously has its own meaning:

A. **Declaration of quantum circuit:** the non-terminal symbol circuit describes a quantum circuit, which can be empty or non-empty depending on the presence of the non-terminal listQubits. The declared quantum circuit can then be understood as a matrix.
B. **Declaration of a list of qubits:** the non-terminal symbol listQubits describes a list of qubits with at least one qubit. The whole list of qubits represents the rows compounding the quantum circuit.
C. **Declaration of a qubit:** the non-terminal symbol qubit describes a qubit. In this language each qubit is understood as each of the rows of the matrix that compose the quantum circuit.
D. **Declaration of a quantum gate:** the non-terminal symbol gate describes a quantum gate. All the quantum gates declared as the aforementioned tokens are valid quantum gates to be included within a qubit.

Moreover, more complex gates, such as the swap gates, controlled gates and oracles have a deeper semantic meaning and restrictions, due to how they are declared in this language:

A. **Swap gates**: the non-negative integer value defined within the swap gate refers to the identifier number of the qubit that is the second target of the swap gate, whereas the first target is the qubit where the swap gate is defined.
B. **Controlled gates**: the non-negative integer value defined within the controlled gate refers to the identifier number of the target qubit of the controlled gate, whereas the controlled qubit is the qubit where the controlled gate is defined.
C. **Oracles**: the non-negative integer value defined within the oracle gate refers to the size of the oracle. The oracle has at least one target qubit, which is the qubit where the oracle is defined. However, it can have more target qubits if the size of the oracle is greater than 1.

However, it is also important to remark that multiple semantic restrictions must be considered when using QCSR to declare quantum circuits. Although the matrix representation of a quantum circuit simplifies its understanding from the user point of view, it also allows the possibility of creating legal syntactic constructions that are semantically illegal, e.g. the presence of swap2 gates without their corresponding swap gate. Thus, the restrictions required to guarantee a well-built quantum circuit are as follows:

A. **Swap gates semantic restrictions**: to guarantee that the usage of swap gates in QCSR is semantically correct, several aspects must be taken into account:
    a. **Swap gate second target**: from the syntactical point of view, it is completely correct to declare a swap gate with a second target of value n, as long as n is higher or equal to 0 (see the construction pattern of the token VALID_NUM). However, for the quantum circuit to be semantically correct using QCSR, the nth qubit of the quantum circuit must have a swap2 gate in the same column as the swap gate. Thus, if there was no gate at such position of the quantum circuit, or if the gate was not a swap2 gate, this semantic restriction would not have been met.
    b. **Unmatched swap2 gates**: although introducing a swap2 gate is syntactically correct, a swap2 gate can only be introduced if there is a swap gate related to it. Thus, the presence of swap2 gates with no swap gate related to them leads to a semantically incorrect quantum circuit in this representation.
    c. **Swap gate ordering**: the swap2 gate must always appear in a latter qubit than the swap gate related to it because it represents the second (not the first) target qubit of a swap gate.
    d. **Quantum gates within swap gates**: in QCSR, for understandability purposes, there cannot be gates in between the swap gate. Thus, in the column where a swap gate is placed, between the first and the second target qubits of the swap gate, there can only be empty space or none gates, which were also designed for this purpose.
B. **Oracles semantic restrictions**: the semantic restrictions that apply to the usage of oracles in QCSR are the following ones:
    a. **Oracle size**: from a syntactical point of view, the introduction of an oracle of a size n is correct, from the semantics point of view, it implies the introduction of an oracle2 gate in the n − 1 consecutive qubits of the quantum circuit and in the same column as the oracle gate was defined, to represent the introduction of an oracle of size n. An oracle defined with size 0 is also considered incorrect from the semantics point of view because an oracle must have at least one target qubit.
    b. **Unmatched oracle2 gates**: as well as the presence of unmatched swap2 gates is considered a semantic error within this language, the presence of unmatched oracle2 gates is also considered semantically incorrect. This is because the oracle2 gates' purpose is to represent an oracle affecting several qubits in the matrix representation of a quantum circuit. Thus, if there is no oracle gate that justifies their usage, they cannot be used.
C. **Controlled gates semantic restrictions**: multiple aspects must also be considered to guarantee the correct usage of controlled gates from the semantics point of view:
    a. **Target of a controlled gate**: the target qubit of the controlled gate must contain, in the same column as the declared controlled gate, the gate that is being controlled and whose control qubit is the qubit where the token GATECONTROL is defined. Moreover, the gate that is being controlled, i.e. the target of the defined controlled gate, must be a valid gate, thus it must not be a swap2 nor an oracle2 gate. Thus, to create a controlled oracle, the oracle gate must be controlled, whereas to create a controlled swap9, the swap gate must be controlled.

b. **Quantum gates within controlled gates**: as well as there cannot be quantum gates in between a swap gate, the same semantic restriction applies to controlled gates. However, there is a slight difference, because the controlled quantum gate could also be a controlled gate itself (which allows representing multi-controlled gates). For this reason, within a controlled gate there can only be empty space, none gates and controls that represent controlled qubits of such controlled gate.