TECHNICAL UNIVERSITY OF DENMARK

Written examination, May 16th 2025

Course title: Quantum Algorithms and Machine Learning

Course number: 02195

Aids allowed: Written material only

Exam duration: 4 hours

Weighting: All problems contribute towards the final grade based on their respective number of points.

You are not allowed to use devices such as laptops, PDAs, mobile phones, MP3 players or pocket calculators.

All questions contribute toward the final grade and are not sorted in order of difficulty. The maximum number of points awarded by each correctly answered question is listed next to the question.

Partial answers are considered and may results in points!

If you make additional assumptions, remember to briefly explain them. For all problems, when answering the different questions, remember to show/explain your calculations and motivate your answers. Failure to do so may cause substantial reduction of the number of points given.

1 Compilers

1.1 Sketch the main stages (for example with a figure) for a classical compiler and a quantum compiler. (2p)

1.2 Consider the following simple grammar for Boolean expressions:

$\langle var \rangle$ ::= [a-zA-Z][a-zA-Z0-9]*

$\langle bool \rangle$ ::= `0` | `1` | $\langle var \rangle$ | $\langle bool \rangle$ `and` $\langle bool \rangle$ | $\langle bool \rangle$ `or` $\langle bool \rangle$ | $\langle bool \rangle$ `xor` $\langle bool \rangle$ | `not` $\langle bool \rangle$ | `(` $\langle bool \rangle$ `)`

Given the following precedence order from tightest to weakest binding:

$$\texttt{not} > \texttt{and} > \texttt{xor} > \texttt{or}$$

and left-associativity for all binary operators, draw the syntax tree for the following expression: `(a or b) xor b and c`. (3p)

1.3 Consider the following partial evaluation rules for Boolean expressions:

$$\begin{aligned}
\mathcal{E}(B, \mathbf{u}, \sigma) &= (B, \top) && \text{if } B \in \{0, 1\} && (1)\\
\mathcal{E}(V, \mathbf{u}, \sigma) &= (\mathbf{u}(V), \top) && \text{if } \sigma(V) = \top && (2)\\
\mathcal{E}(V, \mathbf{u}, \sigma) &= (V, \bot) && \text{if } \sigma(V) = \bot && (3)\\
\mathcal{E}(B_1 \oplus B_2, \mathbf{u}, \sigma) &= \mathcal{O}(\oplus, (x_1, s_1), (x_2, s_2)) && \text{where and } (x_i, s_i) = \mathcal{E}(B_i, \mathbf{u}, \sigma), i = 1, 2 && (4)\\
& && \text{and } \oplus \in \{\texttt{and}, \texttt{or}, \texttt{xor}\}
\end{aligned}$$

where B is any `bool` and V is any `var`. As input to the partial evaluation, we also give an environment $\mathbf{u}$ with values for the known variables, $\sigma(V) = \top$ if V is a statically known variable, and $\sigma(V) = \bot$ if V is dynamic.

The rules for operator evaluation of `or` are:

$$\begin{aligned}
\mathcal{O}(\texttt{or}, (1, \top), (x, s)) = \mathcal{O}(\texttt{or}, (x, s), (1, \top)) &= (1, \top) && (5)\\
\mathcal{O}(\texttt{or}, (0, \top), (x, s)) = \mathcal{O}(\texttt{or}, (x, s), (0, \top)) &= (x, s) && (6)\\
\mathcal{O}(\texttt{or}, (x_1, \bot), (x_2, \bot)) &= (x_1 \texttt{ or } x_2, \bot) && (7)
\end{aligned}$$

i) Complete the $\mathcal{O}$-evaluation rules for `and` and `xor`. (1p) ii) Given the environment $\mathbf{u} = [b \mapsto 0]$, and $\sigma = [a \mapsto \bot, b \mapsto \top, c \mapsto \bot]$, calculate the partial evaluation of the expression `(a or b) xor b and c`. Annotate the equalities between intermediate results with the corresponding rule number. (3p)

Hint: For example, we get the following evaluation:

$$\mathcal{E}(\texttt{a or b}, \mathbf{u}, \sigma) \overset{2,3,4}{=} \mathcal{O}(\texttt{or}, (\texttt{a}, \bot), (0, \top)) \overset{6}{=} (\texttt{a}, \bot)$$

2 Quantum Transformations

2.1 Consider a quantum computing architecture which implements the basic quantum gates

$$SZ = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{8}$$

$$SX = \frac{1}{\sqrt{-2i}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} \tag{9}$$

$$\tag{10}$$

but does not natively support the Hadamard gate H. The gates SZ and SX are the square roots of the Pauli Z and X gates:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = SZ^2 \tag{11}$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = SX^2 \tag{12}$$

Show that the Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{13}$$

can be implemented using the gates SX and SZ as:

$$H = \varphi SZSXSZ \tag{14}$$

for some phase factor φ. What is the value of φ? (3p)

3 Quantum technologies

3.1 You are playing with two quantum computer technologies. Technology A, with a long coherence time but with also long gate time. Technology B has ten time shorter coherence time than technology A and five times shorter gate time. Technology B has twice the connectivity than technology A. Measurement fidelity is roughly the same for both technologies. For what type of circuits would technology A be preferable over B? For what circuits would B be preferable over A? (3p)

4 NISQ

When using superconducting qubits, qubits need to be adjacent for multi-qubit gates to be applied. To allow for arbitrary circuits to be be used, it is possible to apply SWAP gates to swap the states of two adjacent qubits, thus allowing the state of qubits to move. SWAP gates are composed from multiple CNOT gates.

4.1 Why is this a problem in the NISQ era? (1p)

4.2 Propose a compiler strategy which would reduce the number of SWAP gates. (2p)