Quantum Software

Modelling Irreversible Quantum Computation

The University of Nottingham
School of Computer Science & IT
and School of Mathematical Sciences

Quantum Computing

- Can quantum effects be utilised to speed-up computation?
- **Quantum Parallelism** offers a significant speed-up in the computation of some algorithms:
  Shor's Factorisation Algorithm (exponential speed-up);
  Grover's Quantum Database Search (quadratic speed-up).

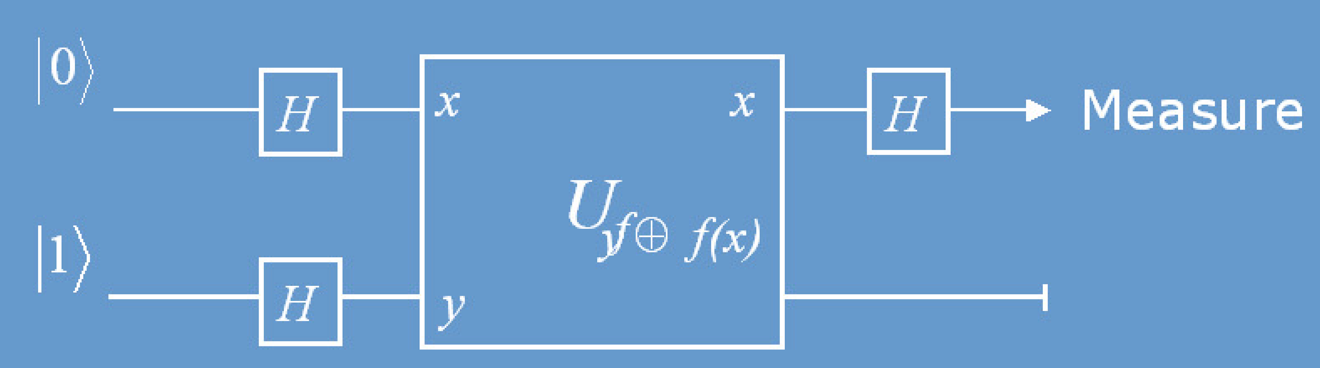
Are there *parallel universes*, computationally?

The Quantum Software Crisis

- How do we develop new quantum algorithms that are better than classical algorithms?
- Current state-of-the-art: use the **Quantum Circuit** metaphor.
- Problems:
  The circuit model is **low-level** and circuits are difficult to design;
  Comparable to programming classical computers using the **Billiard Ball** model of **reversible** computation.

There is also a Quantum Hardware Crisis - only 5 qubits

Deutsch's Algorithm



Where: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

$$U_f = \begin{pmatrix} 1-f(0) & f(0) & 0 & 0 \\ f(0) & 1-f(0) & 0 & 0 \\ 0 & 0 & 1-f(1) & f(1) \\ 0 & 0 & f(1) & 1-f(1) \end{pmatrix}$$

```
H    :: Qubit ⊸ Qubit
H x = if x then { 1 | -0 }
             else { 1 |  0 }

Deutsch      :: (Qubit → Qubit) ⊸ Qubit
Deutsch  f  = Let (x,  y) = (H 0, H 1)
                  (x',!y') = (x, f x ? y)
              in
                  !(H x')
```

- The diagram on the left shows **Deutsch's Algorithm** implemented as a quantum circuit, while the code above shows a QML program realising the same algorithm.
- Deutsch's Algorithm is the prototypical example of quantum computing, making use of both **quantum parallelism** and **interference**. The algorithm takes a function $f \in \{0,1\} \rightarrow \{0,1\}$ and can tell us with certainty, after only one run, whether *f* is a **constant** function. This is twice as fast as is classically possible.

Our Proposal

- We plan to develop **high-level** programming constructs for quantum computers, including:
  Recursion;
  Tree-like Data Structures;
  Higher Order Functions.
- All based on a model of **irreversible** quantum computation.

CS&IT:
J Grattage
T Altenkirch

Mathematics:
V Belavkin

QML: Quantum Meta Language

- QML is an impure **functional** language with **monadic** effects.
- The prototype is implemented in the functional language Haskell.
- Goals:
  To produce a compiler for QML, which outputs Quantum Circuits;
  Denotational semantics to support reasoning about QML programs.

Not enough money from EPSRC to get a real quantum computer ☺

Sponsored by EPSRC MathFIT grant GR/S30818/01

References:
M Nielsen & I Chuang, Quantum Computation & Quantum Information, 2000
P Selinger, Towards A Quantum Programming Language, 2003