On the Power of Classical and Quantum Branching Programs

Chris Pollett, SJSU

(joint work with F. Ablayev, A. Gainutdinova, M. Karpinski, and C. Moore.)

Outline

- What are branching programs?
- Uses of branching programs
- An algebraic definition
- Barrington's result
- Making things random or quantum
- Power of these models.

More about the model

- Allowed to query same variable more than once along path.
- For this talk can break graph into levels according to distance from source.
- Width of program is number of nodes at a level.
- Look at families of programs {Fn} such that

Fn computes a given function on n variable inputs.

Intuitions

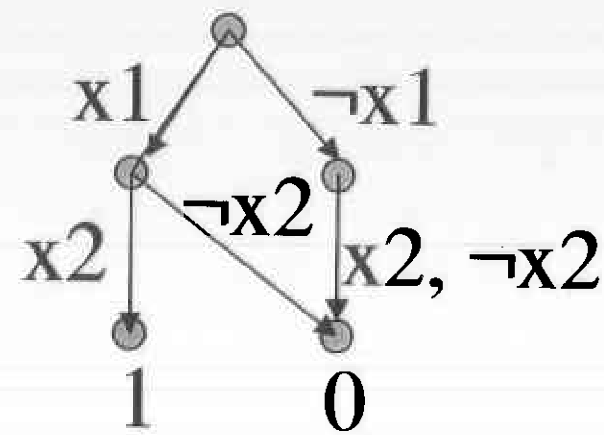
1. If width fixed as input size grows and query variables in order x1,…,xn then very much like a finite automata.
2. So can get good algorithms for synthesizing such BPs according to a given function.
3. Can do minimization.
4. Hence, can equivalence of two such restricted BPs efficiently.

Uses of branching programs

- Given a function f and a circuit R supposedly compute f, we can get corresponding BPs for each of the above restricted type and verify their equivalence.
- Can also use to verify sequential circuits.
- Other uses: test generation, network flow, counting problems, genetic programming.

What are branching programs?

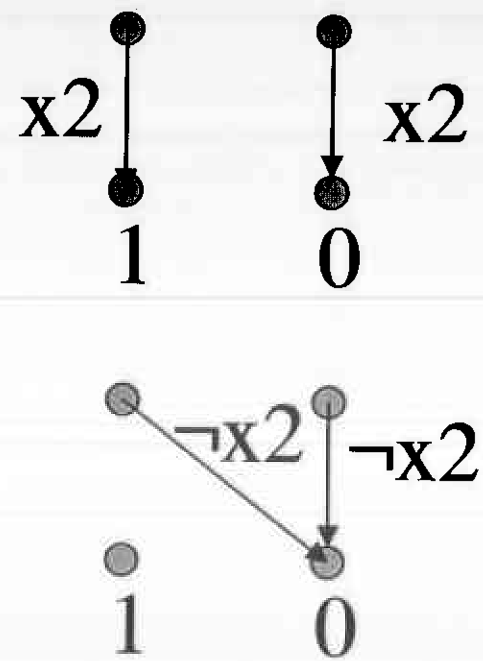
- Acyclic graphs
- Edges labelled with variables
- Follows paths from source to a sink according to variables
- Read off value of sink



AND Function

An algebraic definition

Can view the operation of going from one level to the next as multiplying one of two matrices depending on the value of a variable.



$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

More ideas

- Can also show a gap on read-once models between quantum and random programs for Mod gates.
- The NC^1 result uses the fact that 2 x 2 unitary matrices can be used to represent rotations in 3 dimensions. So can do rotations for symmetries of dodecahedron. The group of these symmetries has necessary properties to allow one to do Barrington's argument.

More on algebraic view

- To evaluate a branching program can thus be viewed as multiplying 0,1 valued matrices that correspond to value of the variables and seeing what if the final state.
- The width of program correspond to the number of rows or columns in the matrix.

Barrington's Result

- Barrington'86 used this algebraic idea, together with the fact you can come up with 5 x 5-matrices A,B such that $ABA^{-1}B^{-1}$ is not the identity matrix in a special way to show width-5 branching programs can simulate log-depth, polynomial size AND, OR, NOT – circuits. His result 5-BP = NC^1.

Making things random or quantum

- Algebraic point of view makes it easy to define randomized or quantum programs.
- In random case, we allow entries in matrices to be from interval [0, 1] and such that the rows sum to 1.
- In quantum case, we take matrices U over complex numbers such $U^{\dagger}U=I$. (Unitary).
- Width can be defined in terms of matrix size.
- In both case need to define what it means to accept in terms of probability see a 1 after performing the matrix multiplications on an input vector.

Power of these models

- No width-2 stochastic program (our randomized model above) can recognize majority with success >3/4.
- Width-2 quantum branching programs compute exactly NC^1.

Ideas behind these results

- The first result actually follows from a general trade-off result in both the quantum and random case we get on acceptance error versus program width.
- Result exploits the fact that neither the stochastic nor quantum matrices can ``increase'' distances and that we need to have a certain distance between accepting and rejecting states.