The Complexity Class CC

Comparator circuits are composed of comparator gates:

Wires are initialized with constants, inputs or negated inputs. Output is read from a designated output wire. 

Wires cannot split!

Oracle Separations

Conjecture: CC and NC are incomparable. Implies NL ≠ P. Evidence: oracle separations.

Oracle models: (separations work for both)

(1) n inputs, 1 output

(2) n inputs, n outputs

Satisfies 1-flip property: If one input bit is flipped, one output bit is flipped.

Comparator Circuit Value Problem (CCVP):

Given a comparator circuit and an assignment to the inputs, determine the value of the output. [Mayr & Subramanian ’92]

Containment Relations and Complete Problems

Solving directed reachability using comparator circuits:

(our simplified proof)

• Repeat n times:
  ▶ Pebble source node
  ▶ For each edge $x \rightarrow y$:
    ▶ Move pebble from $x$ to $y$ if possible (★)
  ▶ Check if target is pebbled

Comparator circuit depends on input graph, but can be constructed from input graph in $AC^0$.

Operation ★ is implemented by a comparator.

Complete problems:

• Comparator circuit value problem
• Lexicographically-first maximal matching: [Mayr & Subramanian ’92]

Given a bipartite graph $(V,W,E)$, greedily match vertices in $V$ to first unmatched vertex in $W$.

Is a given vertex matched? Is a given edge part of the matching?

• Stable marriage problem: [Subramanian ’94]

Input: n men and n women, each with a preference order on people of the opposite sex.

A stable marriage is a perfect matching in which no two unmatched people prefer each other over their current partners.

Are two people matched in the woman-optimal solution?

Gale & Shapley: stable marriage always exists, algorithm.

Subramanian: fixed-point algorithm using comparator circuits.

The complexity class CC

Definitions as closure of CCVP:

1. All languages $AC^0$-Turing reducible to CCVP.
2. All languages $AC^0$-many-one reducible to CCVP.
3. All languages $NL$-many-one reducible to CCVP. [Mayr & Sub. ’92]

Definitions as uniform comparator circuits:

4. Predicates computed by uniform polynomial-size comparator circuits with input wires initialized by constants, inputs and negated inputs.
5. Predicates computed by uniform polynomial-size comparator circuits with negation gates and input wires initialized by constants and inputs.

A function in $CC^0 \setminus NC^0$:

Oracle encodes a function $f: \{0,1\}^n \rightarrow \{0,1\}^n$.

Problem: compute bit $0$ of $f^{\sqrt{n}}(0)$.

Requires depth $\sqrt{n}$, which is easy for $CC^0$.

A function in $NC^0 \setminus CC^0$:

Oracle encodes a sequence of functions $f_1, \ldots, f_m: \{0,1\}^{2n} \rightarrow \{0,1\}^n$, where $m = \log^2 n$.

Each function expects an input of the form $x_1 x_2 \cdots x_n x_1 x_2 \cdots x_n y_1 y_2 \cdots y_n$.

Problem: compute parity of $f_m \circ \cdots \circ f_2 \circ f_1$.

Requires fan-out, which is easy for $NC^0$.

Lower bound for $CC^0$ uses 1-flip property.