Gate Elimination

George Voutsadakis[1]

[1]Mathematics and Computer Science
Lake Superior State University

Seminar Presentation
Lake Superior State University

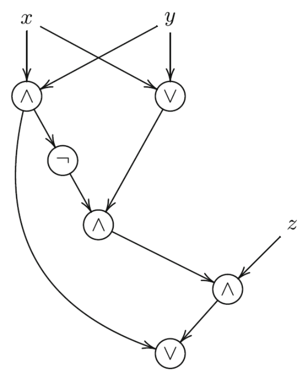
Circuits

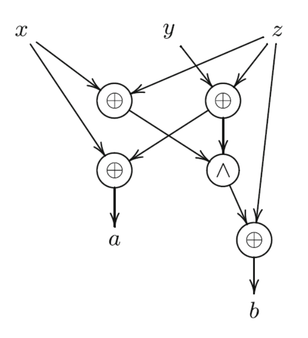
- Let Φ be a set of some boolean functions.
- A **circuit** (or a **straight line program**) of n variables over the basis Φ is just a sequence $g_1, \ldots, g_t$ of $t \geq n$ boolean functions such that:
  - the first n functions are input variables $g_1 = x_1$, ..., $g_n = x_n$;
  - each subsequent g_i is an application $g_i = \varphi(g_{i_1}, \ldots, g_{i_d})$ of some basis function $\varphi \in \Phi$ (called the **gate** of g_i) to some previous functions.

  I.e., the value $g_i(a)$ of the i-th gate g_i on a given input $a \in \{0,1\}^n$ is the value of the boolean function $\varphi \in \Phi$ applied to the values $g_{i_1}(a)$, ..., $g_{i_d}(a)$ computed at the previous gates.
- A circuit **computes** a boolean function (or a set of boolean functions) if it (or they) are among the g_i.

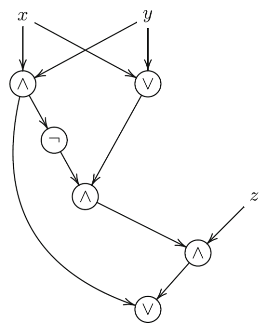
Visualizing Circuits

- Each circuit can be viewed as a directed acyclic graph whose:
  - fanin-0 nodes (those of zero in-degree) correspond to variables;
  - each other node v corresponds to a function $\varphi \in \Phi$;
  - one (or more) nodes are distinguished as outputs.



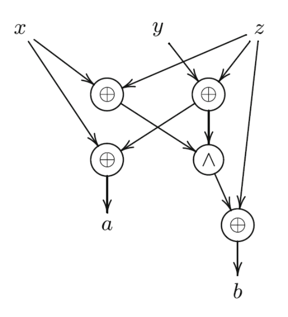


Majority Functions



- This circuit has six gates over the basis $\{\wedge, \vee, \neg\}$, is of depth 5 and computes the **majority** $\text{Maj}_3(x, y, z) = 1$ iff $x + y + z \geq 2$.
- In fact, the output is $(x \wedge y) \vee ((x \vee y) \wedge \neg(x \wedge y) \wedge z)$, which says:
  - $x = y = 1$ or
  - exactly one of x and y is 1 and $z = 1$.

Binary Sum



- This circuit has five gates over $\{\oplus, \wedge\}$ and computes the binary representation (a, b) of the (real) sum $x + y + z$ of three bits.
  - $a = x \oplus (y \oplus z)$ is 1 exactly when one or three of x, y and z are 1.
  - $b = ((x \oplus z) \wedge (y \oplus z)) \oplus z$ is 1 if
    - $x = y = 1$ and $z = 0$; or
    - at least one of x, y is 1 and $z = 1$.

Size of a Circuit

- The **size** of the circuit is the total number $t-n$ of its gates (that is, we do not count the input variables);
- Its **depth** is the length of a longest path from an input to an output gate:
  - Input variables have depth 0;
  - If $g_i = \varphi(g_{i_1}, \ldots, g_{i_d})$, then the depth of the gate g_i is 1 plus the maximum depth of the gates g_{i_1}, ..., g_{i_d}.
- We assume that every circuit can use constants 0 and 1 as inputs for free.

Outline of the Gate Elimination Technique

- The **gate-elimination argument** does the following:
  - Starts with a given circuit for the function in question.
  - Argues that some variable (or set of variables) must fan out to several gates.
  - Sets this variable to a constant to eliminate several gates.
  - By repeatedly applying this process, concludes that the original circuit must have had many gates.

Gate Elimination for Threshold Functions

- We apply the gate elimination argument to threshold functions

$$\mathrm{Th}_k^n(x_1, \ldots, x_n) = 1 \quad \text{iff} \quad x_1 + x_2 + \cdots + x_n \geq k.$$

Theorem

Even if all boolean functions in at most two variables are allowed as gates, the function Th_2^n requires at least $2n - 4$ gates.

- By induction on n.
  - For $n = 2$ and $n = 3$ the bound is trivial.
  - For the induction step, take an optimal circuit for Th_2^n.
    Suppose that the top-most gate g acts on variables x_i and x_j, $i \neq j$.
    This gate has the form $g = \varphi(x_i, x_j)$, for some $\varphi : \{0,1\}^2 \to \{0,1\}$.
    Notice that under the four possible settings of these two variables, the function Th_2^n has three different subfunctions
    - Th_0^{n-2}, if $x_i = x_j = 1$;
    - Th_1^{n-2}, if exactly one of x_i, x_j is 1;
    - Th_2^{n-2}, if $x_i = x_j = 0$.

Gate Elimination for Threshold Functions

- It follows that either x_i or x_j fans out to another gate h.

  Otherwise our circuit would have only two inequivalent sub-circuits under the settings of x_i and x_j, since the gate $g = \varphi(x_i, x_j)$ can only take two values, 0 and 1.

  Now suppose that it is x_j that fans out to h.

  Setting x_j to 0 eliminates the need of both gates g and h.

  The resulting circuit computes Th_2^{n-1}.

  By induction, it has at least $2(n-1) - 4$ gates.

  Adding the two eliminated gates to this bound shows that the original circuit has at least $2n - 4$ gates.

The Parity Function

- For circuits over the basis $\{\wedge, \vee, \neg\}$ one can prove a slightly stronger lower bound.
- We consider the parity function

$$\oplus_n(x_1, \ldots, x_n) = x_1 \oplus x_2 \oplus \cdots \oplus x_n.$$

Schorr's Theorem

The minimal number of $\wedge$ and $\vee$ gates in a circuit over $\{\wedge, \vee, \neg\}$ computing $\oplus_n$ is $3(n-1)$.

- The upper bound follows since $x \oplus y$ is equal to $(x \wedge \neg y) \vee (\neg x \wedge y)$.

  For the lower bound we prove the existence of some x_i whose replacement by a suitable constant eliminates 3 gates.

  This implies the assertion for $n = 1$ directly and for $n \geq 3$ by induction.

The Parity Function

- Let g be the first gate of an optimal circuit for $\oplus_n(x)$.

  Its inputs are different variables x_i and x_j.

  If x_i had fanout 1, that is, if g were the only gate for which x_i is acting as input, then we could replace x_j by a constant so that gate g be a constant ($x_j = 0$ if $g =$ "$\wedge$" and $x_j = 1$ if $g =$ "$\vee$").

  This would imply that the output became independent of the i-th variable x_i in contradiction to the definition of parity.

  Hence, x_i must have fanout at least 2.

  Let g' be the other gate to which x_i is an input.

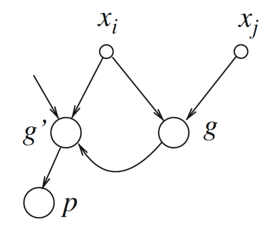
  We now replace x_i by such a constant that g becomes replaced by a constant ($x_i = 0$ if $g =$ "$\wedge$" and $x_i = 1$ if $g =$ "$\vee$").

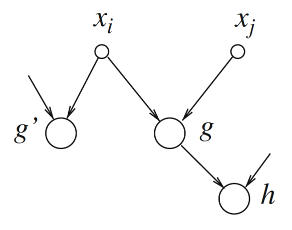
  Since under this setting of x_i the parity is not replaced by a constant, the gate g cannot be an output gate.

  Let h be a successor of g.

Configurations

- We only have two possibilities: either h coincides with g' (that is, g has no other successors besides g') or not.





$g' = h$: In this case g has fanout 1.
We can set x_i to a constant so that g' be set to a constant.
This will eliminate the need for all three gates g, g' and p.

$g' \neq h$: Then we can set x_i to a constant so that g be set to a constant.
This will eliminate the need for all three gates g, g' and h.

In either case we eliminate at least 3 gates.

A Remark Concerning the Proof

- The same argument works if we allow as gates any boolean functions $\varphi(x, y)$ with the following property:

  There exist constants $a, b \in \{0, 1\}$ such that both $\varphi(a, y)$ and $\varphi(x, b)$ are constants.

- The only two-variable functions that do not have this property are the parity function $x \oplus y$ and its negation $x \oplus y \oplus 1$.

Thank you!

- In closing...

Thank you for your Attention!!