1 Håstad's Switching Lemma

FSS81, Ajtai83 proved the first switching lemmas. Håstad proved the strongest version. Here we give a variant of Razborov's "constructive" proof of the switching lemma. The presentation is borrowed from Paul Beame's survey.

A boolean function f on n inputs is a *k-DNF* if it can be written as an OR of ANDs, where each AND consists of at most k literals. Each of the ANDs is called a *term* of f. The switching lemma studies how f simplifies when we apply a random *restriction*. We randomly pick t inputs and randomly (and independently) set each to 0/1. Let this restriction be denoted ρ. Note that after applying ρ, we have converted f into a function on $n-t$ variables, called $f|_\rho$.

Formally, the above restriction method is equivalent to choosing a restriction randomly from R_n^t, the set of all restrictions that fix t inputs. Note that $|R_n^t| = \binom{n}{t}2^t$.

The switching lemma says that a random restriction chosen this way is very likely to simplify the function considerably. Let $D(f)$ denote the decision-tree complexity of f.

Switching Lemma: *Let f be a k-DNF on n inputs. If ρ is randomly chosen from R_n^t then*

$$\Pr_\rho[D(f|_\rho) > s] < (\frac{7(n-t)k}{n})^s.$$

Main Idea: Let us call a restriction $\rho \in R_n^t$ as *bad* if $D(f|_\rho) > s$. The main idea is to show that any bad ρ can by *uniquely* described by an ordered pair (ρ', y, z), where $\rho' \in R_n^{t+s}$, y is a string in a set $stars(k,s)$ (to be described later) of size at most $(k/\ln 2)^s$, and z is a string of s bits. Note that this allows us to upperbound the number of bad restrictions by $|R_n^{t+s}| \cdot |stars(k,s)| \cdot 2^s$, and thus the probability of picking a bad restriction by

$$\begin{aligned}\frac{|R_n^{t+s}| \cdot 2^s\,|stars(k,s)|}{|R_n^t|} &= 2^{2s} \cdot |stars(k,s)| \cdot \frac{\binom{n}{n-t-s}}{\binom{n}{n-t}} \\ &\leq (\frac{7(n-t)k}{n})^s.\end{aligned}$$

(In the last line we used the approximation $\binom{m}{q} \approx (me/q)^q$.)

Detailed proof: Let $stars(k,s)$ be the set of sequences of the form $y_1, y_2, \ldots,$, where each $y_i \in \{*,\#\}^k \setminus \#^k$ (i.e., each y_i is a sequence of k symbols from $\{*,\#\}$ and contains at least one $*$) and the total number of $*$'s in the sequence is at most s. The following fact is easily proved by induction on s.

Claim: *The number of sequences in $stars(k,s)$ is at most $(k/\ln 2)^s$.*

Now we return to our function f. We arbitrarily order the variables and terms of f. The *canonical decision tree* T for f is the one that first queries all input variables in the first term, then all variables in the second term, and so on until either it determines that a term is true (thus implying f is true) or it verifies that all terms are false. Within each term the tree queries the input variables in order.

Note that applying restriction ρ on f simplifies the decision tree and gives a decision tree, $T|_\rho$, for $f|_\rho$. Let us say ρ is *evil* if $T|_\rho$ has a path of length more than s. Note that a bad restriction is also evil. Now we state the main lemma, which, as already noted, implies the switching lemma.

Main Lemma: *There is a* 1-1 *map that maps the set of evil restrictions in* R_n^t *to* $R_n^{t+s} \times stars(k,s) \times \{0,1\}^s$.

Let ρ be an evil restriction and let π be a path in $T|_\rho$ of length more than s. We truncate π so that it has length exactly s. Consider the first term of f, say C_1, that is not set to 0 by ρ (i.e., ρ does not set any literal of C_1 to 0 and does not set all literals of C_1 to 1.) Let σ_1 be the unique assignment to input variables in C_1 that sets C_1 to 1. Let π_1 be the portion of π that assigns values to variables in C_1. There are two cases.

Case 1: π_1 *is a proper subset of* π. Then π_1 must assign some variable differently than σ_1, since otherwise f would be set to 1 by π_1 and this would contradict the assumption that $T|_\rho$ had a path of length $> s$. Hence we conclude that π_1 sets C_1 to 0.

Case 2: π_1 *is exactly* π*:* Then we can again conclude that π_1 cannot set C_1 to 1.

Now consider the $\rho\pi_1$, which sets all variables according to ρ and π_1. If $\pi_1 = \pi$ we stop otherwise we can reason as above to find a clause C_2 that is not set to 0, and therefore corresponding partial assignments σ_2 and π_2. Continuing this way we get a sequence of clauses $C_1, C_2, \ldots$. Each C_i contains some variable that was not in C_j for $j < i$, so we must stop after we have identified at most s clauses. Say we identified m clauses.

Let $\beta_i \in \{*, \#\}^k \setminus \#^k$ be a string that contains a $*$ in the jth position iff the jth variable of term C_i is set by σ_i . Let $z \in \{0,1\}^s$ be a vector that denotes, for each variable set by π (which are the same variables set by $\sigma_1\sigma_2\cdots\sigma_m$), whether π and $\sigma_1\sigma_2\cdots\sigma_m$ agree on the that variable.

Now we define the mapping mentioned in the Main Lemma. We map restriction ρ to $(\rho\sigma_1\sigma_2\cdots\sigma_m, \beta_1\beta_2\cdots\beta_m, z)$.

Finally, we need to argue that the mapping is 1-1. We show how to construct ρ uniquely, given $(\rho\sigma_1\sigma_2\cdots\sigma_m, \beta_1\beta_2\cdots\beta_m, z)$.

First note that it is easy to reconstruct σ_1. Identify the first clause that is not set to 0 in $\rho\sigma_1\sigma_2\cdots\sigma_m$: this must be C_1. Now use β_1 to determine the variables of C_1 that are set by σ_1, and use z to identify how π_1 would set these variables. Thus we have both σ_1 and π_1. Now we can identify C_2: it is the first clause that is not set to 0 by $\rho\pi_1\sigma_2\cdots\sigma_m$. Then use β_2 to identify the variables set by σ_2, and use z to identify how these variables would be set by π. Continuing this way, we can reconstruct $\sigma_1\sigma_2\cdots\sigma_m$, and thus ρ.