CS38600-1 Complexity Theory A **Spring 2003**

Lecture 5 : Littlewood-Richardson Rule

Lecturer: Ketan Mulmuley *Scribe: Murali Krishnan Ganapathy*

Abstract

In this lecture, we discuss the Littlewood Richardson rule which allows us to determine the multiplicity of each irreducible representation V_λ in $V_\alpha \otimes V_\beta$, where V_θ is the irreducible representation of SL_n corresponding to the partition θ with at most n parts.

5.1 Recall

All the finite dimensional representations of SL_n are the Weyl modules V_λ. Here λ can be any partition with at most n parts. Unlike in the finite group case, the number of irreducible representations are infinite.

5.2 The Littlewood Richardson coefficients

For partitions α, β, we know that $s_\alpha s_\beta = \sum_\lambda N_{\alpha\beta\lambda} s_\lambda$, for suitable coefficients $N_{\alpha\beta\lambda}$. Here λ varies over all partitions of $|\alpha| + |\beta|$. This follows, since the Schur polynomials form a basis of the vector space of all symmetric homogenous polynomials of appropriate degree.

The same coefficients $N_{\alpha\beta\lambda}$ also appear in the decomposition of $V_\alpha \otimes V_\beta$, through the formula

$$V_\alpha \otimes V_\beta = \sum_\lambda N_{\alpha\beta\lambda} V_\lambda$$

In particular this shows that even though SL_n has infinitely many irreducible representations, only a small number of them actually occur in the representation $V_\alpha \otimes V_\beta$, since $N_{\alpha\beta\lambda} = 0$ if $|\alpha| + |\beta| \neq |\lambda|$.

5.3 Combinatorial evaluation of $N_{\alpha\beta\lambda}$

Fix any Young Diagram α. A β_1 expansion of α is obtained by appending α to get a new Young diagram using β_1 1's so that no two 1's are in the same column. Similarly a β_2 expansion is obtained by appending β_2 2's so that no two are in the same column.

Definition 5.1 *Let $\alpha, \beta = (\beta_1 \geq \cdots \geq \beta_k)$ be two partitions. A β expansion of α is defined to be a β_k expansion of a β_{k-1} expansion of a $\ldots \beta_2$ expansion of a β_1 expansion of α. Such an expansion is called **strict**, if when reading right to left, top down, we have that $\forall p \in \{1, \ldots, k-1\}$ and all positions t, the number of p's up to position t is $\geq$ the number of $(p+1)$'s up to position t.*

Theorem 5.2 *$N_{\alpha\beta\lambda}$ equals the number of strict β-expansions of α of shape λ.*

Hence we have that

$$V_\alpha \otimes V_\beta = \bigoplus_E V_{sh(E)}$$

where the direct sum is taken over over all strict β expansions of α, and $sh(E)$ refers to the shape of E, i.e. the Young Diagram underlying E.

This immediately gives an NP algorithm to check if $N_{\alpha\beta\lambda} > 0$. On input α, β, λ, we check if $|\lambda| = |\alpha| + |\beta|$ and guess a tableaux E of size $|\lambda|$. Then we verify that E is a strict β expansion of α, and if so we accept. It is easy to see that this algorithm is in NP if the input were specified in unary, i.e. input size $= O(|\lambda|)$. In fact, even when λ is compressed, i.e. has input size $O(\sum \log \lambda_i)$, if $\lambda = (\lambda_1 \geq \lambda_2 \geq \dots)$, it can be shown that the above algorithm is still in NP. In particular, we have shows that the computation of $N_{\alpha\beta\lambda}$ lies in $\#P$ and has a $P^{\#P}$ decomposition formula without alternating signs.

Theorem 5.3 *Deciding if V_λ occurs in the decomposition of $V_\alpha \otimes V_\beta$ is polynomial time decidable, even if input is compressed.*

Proof: If there exists a β-expansion of α of shape λ, then there is a greedy strict β-expansion of α of shape λ. The greedy algorithm first carves out α from the shape λ (if unable to do, then return false).

Assume we have filled up the remaining squares with $\beta_1 1$'s, $\beta_2 2$'s, $\dots \beta_p p$'s. Processing the squares top to bottom and right to left. Find the first empty square whose column does not contain a p. Fill that box with a p and continue filling the remaining $\beta_p - 1 p$'s.

If at any stage we are unable to find a empty square satisfying the strictness condition (i.e. column does not already have a p), then it is easy to see that there is no strict β-expansion of α of shape λ.

Clearly this gives a P-time computable algorithm assuming the input is in unary, i.e. not compressed. The same algorithm can be modified to handle the case when the input is compressed. In this case, we fill up the empty squares in blocks, i.e. many at a time. ■

Conjecture 5.4 *(The above problem for S_n) The problem of deciding if S_λ occurs in the representation $S_\alpha \otimes S_\beta$ is P-computable.*

Fact 5.5 *The above problem for SO_n, SP_{2n}, and other reductive groups are P-computable.*

5.4 An effective version of $V_\alpha \otimes V_\beta = \sum_\lambda N_{\alpha\beta\lambda} V_\lambda$

Since $V_\alpha \otimes V_\beta = \sum_\lambda N_{\alpha\beta\lambda} V_\lambda$ as SL_n-modules, the same holds in particular as complex vector spaces. A natural vector space basis of V_α is given by $I_\alpha = \{T \colon T$ is a semi-standard tableaux of shape $\alpha\}$. So from the above result, we know that the sets

$$I_\alpha \times I_\beta \text{ and } \dot{\bigcup_\lambda} I_\lambda^{N_{\alpha\beta\lambda}}$$

are in bijective correspondence with each other. We now show a natural P-computable correspondence between the two sets.

Theorem 5.6 ***(Polynomial time decomposition rule)*** *There exists a P-computable correspondence* $\varphi\colon I_\alpha \times I_\beta \Longleftrightarrow \cup_\lambda I_\lambda^{N_{\alpha\beta\lambda}}$. *Moreover* φ^{-1} *is also P-computable.*

5.4.1 RSK Correspondence

Before we get into the proof, we need to define the RSK Correspondence:

Given a sequence S of symbols $\{1,\ldots,n\}$ of length $\ell = \ell(S)$, we define $RSK(S) = (P,Q)$, where P is a semi-standard tableaux over $\{1,\ldots,n\}$ of shape λ, where $|\lambda| = \ell$ and Q is a standard tableaux over $[1,\ldots,\ell]$ also of shape λ. The construction of P and Q proceeds as follows:

Let $S = Rk$, where R is a sequence of length one less than S and k is the last symbol occurring in S. By induction hypothesis, assume that we already know $RSK(R) = (P,Q)$. The Algorithm starts with P and Q and inserts k into P and let it propagate down modifying P along the way. Q records the position at which k settles down finally. The propagation of k occurs by executing Bump(first row,k).

Bump(x'th row,v): If all the entries of the x'th row of P is $\leq v$, append v to the x'th row. Otherwise, let i be the left most entry of the x'th row which is larger than v. Replace i with v and bump out i. Now execute Bump(x+1'st row,i). Since the number of rows in P in bounded, eventually this procedure will encounter an empty row, and become the first entry in that row.

For Q, create a box where the new box in P was created and fill it with $\ell = \ell(S)$, thus noting the time at which this box was created in P.

It is easy to see that this mapping can be reversed as follows. Starting with Q, we know the position of ℓ is the box where the last symbol was added to P. If that is not the first row of P, we know this symbol was bumped out of the previous row. Applying a reverse bump procedure, we can find out which symbol replaced in the previous row. Repeat the above reverse bump with the new symbol we have identified. Eventually we will reverse bump a symbol out of the first row. This is the last symbol of S.

Repeating the above process, we can recover all the symbols of S. It is also easy to see that this can be implemented in polynomial time, even if the input sequence is encoded.

5.4.2 Proof of Theorem 5.6

Let $T_\alpha \in I_\alpha, T_\beta \in I_\beta$ be given. Denote by C_α the canonical standard tableaux of shape α, i.e. numbers $1\ldots n$ increasing left to right top to bottom. Similarly for C_β.

Let $S_\alpha = RSK^{-1}(T_\alpha, C_\alpha)$ and $S_\beta = RSK^{-1}(T_\beta, C_\beta)$, and define $\varphi(T_\alpha, T_\beta) = RSK(S_\alpha \circ S_\beta)$, where $\circ$ is the concatenation operator. So $\varphi(T_\alpha, T_\beta) = (P_\lambda, Q_\lambda)$, where $P_\lambda \in I_\lambda$ and Q_λ is thought of as the index in a set of size $N_{\alpha\beta\lambda}$.

We then complete the proof by showing that if $(X, Q_\lambda) \in \Im(\varphi)$, then so is (Y, Q_λ) for any $Y \in I_\lambda$. This is left as an exercise to the reader. This completes the proof.

As a consequence we have

$$V_\alpha \otimes V_\beta = \sum_{(T_\alpha, T_\beta)} V_{\lambda(\varphi(T_\alpha, T_\beta))}$$

where the sum is taken over semi-standard tableaux's such that $\varphi(T_\alpha, T_\beta) = (C_\lambda, \cdot)$, where C_λ is the canonical standard tableaux of shape λ. Note that the original result was for the vector space basis, and now we have it for the Weyl modules themselves. That is why we had to restrict the first component of $\varphi(T_\alpha, T_\beta)$ to be some fixed standard tableaux.

The above is a $P^{\#P}$ formula without alternating signs. Similar formulae are known for the Specht modules as well as for the Plethysm problem.