CS 880: Complexity of Counting Problems February 23, 2012

Lecture 10: Reduction and Dyer-Richerby

Instructor: Jin-Yi Cai Scribe: Aaron Gorenstein

1 Simulating an Existential Quantifier

We begin with a reduction. Let Γ be a finite set of relations on the finite set A. Say that there exists

$$P(x_0, \ldots, x_k) \in \Gamma, Q(x_1, \ldots, x_k) = \exists x_0 P(x_0, x_1, \ldots, x_k). \tag{1}$$

Note the k is constant here. We will prove the following lemma:

Lemma 1

$$\#CSP(\Gamma \cup \{Q\}) \leq_T \#CSP(\Gamma). \tag{2}$$

This has the corollary:

Corollary 1

$$\#CSP(\langle\Gamma\rangle) \equiv_T \#CSP(\Gamma). \tag{3}$$

Recall that $\langle\Gamma\rangle$ is technically infinite, so we really mean that every finite subset is equivalent to the RHS. One direction of the equivalence is easy, and the other direction follows from our reduction.

Simulating the Existential. Given an instance of $\#CSP(\Gamma \cup \{Q\})$, say that Q is used n times. This n, unlike k, *is* part of the input. Let's first try to replace Q with P. Understand that Q, being a relation, can be seen as the set of tuples:

$$Q = \{\overrightarrow{a_1}, \overrightarrow{a_2}, \ldots, \overrightarrow{a_t}\} \tag{4}$$

where t is finite (a combination of finite domain and fixed arity). So consider the tuple $\overrightarrow{a_1}$: what does that mean for P? It really means that $\overrightarrow{a_1}$ may have up to some ℓ_1 *extensions*, and $\overrightarrow{a_2}$ has ℓ_2 extensions, and so on.

From our original (Q-containing) instance I we can make I' without any Q, and we can Turing-reduce: if we know the number of solutions to I' (or multiple variations), we can compute the number of solutions to I, achieving our goal. We can start by considering that for every P, there's some $\overrightarrow{a_i}$ that satisfies it, plus ℓ_i extensions, but that is exponentially many cases to consider!

The trick we use is to define $N(s_1, s_2, \ldots, s_t)$ where $s_1, s_2, \ldots, s_t \geq 0$, and $\sum s_i = n$. This refers to the cardinality of solutions to I where, of the n occurrences of Q, exactly s_1 are satisfied by a_1, s_2 by a_2, and so on. So we avoid caring about the order, and you see this value is quite useful. We have seen this trick before! The number of solutions to I, our goal, can be defined in terms of N:

$$I = \sum_{\substack{\sum_{i=1}^{t} s_i = n, \\ s_1, s_2, \ldots, s_t \geq 0}} N(s_1, s_2, \ldots, s_t) \tag{5}$$

This does *not* have exponentially many terms! There are $\binom{n+t-1}{t-1}$ terms, and recall that t is constant, so that is polynomial.

For every solution, P has a superset of the satisfying assignments for Q. This is for *every* Q that we have replaced with P. So for *each* of these P satisfied by $\overrightarrow{a_i}$, there are ℓ_i many extensions. Thus we know the number of solutions for such an I' is

$$\sum_{s_1, s_2, \ldots, s_t} N(s_1, s_2, \ldots, s_t) \ell_1^{s_1} \ell_2^{s_2} \ldots \ell_t^{s_t}. \tag{6}$$

This comes about by replacing every instance of Q by P. We know the number of extensions, and the value we really want to know about is $N(\cdot)$. Note that if every tuple has exactly one extension, we're done! But of course we can't assume that, and we currently do not have enough information to infer $N(\cdot)$.

What we do now is construct duplicate copies of I'. In particular, we make m duplicates, where I^m means each Q constraint is replaced by P with m different free variables. For example, if we had $Q(x_3, x_1, x_1, x_7)$, we make m duplicates of P, in the following form: $P(y_i, x_3, x_1, x_1, x_7)$. Note that each y_i is used exactly once: so there are m times the *number* of Q clauses we're replacing! The number of satisfying solutions to I^m is:

$$\sum N(s_1, s_2, \ldots, s_t)(\ell_1^{s_1}\ell_2^{s_2}\ldots\ell_t^{s_t})^m \tag{7}$$

Note how the core value, the $N(\cdot)$ value, is *independent* of m.

We can set this up as a linear system, writing it as $\sum_{\overrightarrow{s}} x_{\overrightarrow{s}} \cdot (\alpha_s)^m$ and noting this is a value we know due to our Turing reduction. The unknowns in this system are the x values, by which we mean the $N(\cdot)$ value. But this then becomes a Vandermonde system! There is the question if every row is distinct, but even if some are the same we can combine the associated xs and continue: our goal is to compute the sum of the x values, so adding a few "earlier" does not harm our outcome. □

2 Proof by Dyer-Richerby

This proof relates how the Malt'sev polymorphism relates to rectangularity and its connection to congruency. In the big picture: This will sew up our clone theorem, with the idea being that if you *don't* have a Malt'sev polymorphism, you get something $\#P$-hard, like INDSET to ANTICHAIN. In the future we will discuss this in the context of graph homomorphisms: there the Malt'sev is iff, not just if, in terms of tractability. Then we will return to $\#CSP$.

The theorem, that there is a Malt'sev polymorphism if and only if the relational clone is congruence-permutable, is already known to be true in Universal Algebra, but it is usually stated in an infinite setting. It involves constructions of the (arbitrary, possibly infinite) direct product, and sub-algebra, homomorphism constructions. It can be carefully checked that the corresponding finite version also holds, by the standard (but long chain of) proofs from Universal Algebra. But we will present a direct proof by Dyer-Richerby for the finite case.

We begin by defining rectangularity.

Definition 1 (Rectangularity) A relation R is expressed as $R \subseteq A^{k+\ell}$, for k, ℓ expressing an arbitrary partition of R's arity. The relation R is rectangular if:

$$\begin{aligned}
&(\overrightarrow{a}, \overrightarrow{b}) \in R \\
&(\overrightarrow{b}, \overrightarrow{c}) \in R \\
&(\overrightarrow{b}, \overrightarrow{d}) \in R \implies (\overrightarrow{a}, \overrightarrow{d})
\end{aligned}$$

Consider this visualization:

$$\begin{array}{cc} a & c \\ b & d \end{array}$$

Essentially the "rows" are the tuples, and they are partitioned into $(\overrightarrow{a}, \overrightarrow{c})$ and $(\overrightarrow{b}, \overrightarrow{d})$. Rectangularity says that if we draw one diagonal line and still have a valid tuple, then the other diagonal must also be a valid tuple. We will call Γ rectangular if every $R \in \langle\Gamma\rangle$ is rectangular.

Now we are ready to state our theorem:

Theorem 1 If Γ (equiv. $\langle\Gamma\rangle$) is rectangular, then there exists a Malt'sev polymorphism of Γ ($\langle\Gamma\rangle$).

Proof. We will start by listing all tuples for all relations H in Γ. We list: $H_1, H_2, \ldots, H_{|\Gamma|}$. Each H_i is a segment, indexed by *triples* of elements in H. Do not get confused by the fact the elements in H are themselves also tuples! So each H_i has length $|H_i|^3$.

With this listing, we define the relation R as all possible tuples of that form: where the i^{th} segment (being the slots indexed by triples from H_i) contains only elements from H_i. So the *arity*, not the size, of this R is

$$\sum_{i=1}^{|\Gamma|} |H_i|^3 \cdot arity(H_i). \tag{8}$$

We can notate $|H_i|^3$ as ℓ_{H_i} and the arity as r_{H_i}. In other words, $R = H_1 \times H_1 \dots H_1 \times H_2 \dots H_2 \dots$, where each i is present the right amount of times.

We highlight 3 entries in R in particular.

$$\begin{gathered}\overrightarrow{u_1} \text{ in index } j_1, j_2, j_3 \in [\ell_H] \text{ has tuple } \overrightarrow{x_{j_1}} \\ \overrightarrow{u_2} \text{ in index } j_1, j_2, j_3 \in [\ell_H] \text{ has tuple } \overrightarrow{x_{j_2}} \\ \overrightarrow{u_3} \text{ in index } j_1, j_2, j_3 \in [\ell_H] \text{ has tuple } \overrightarrow{x_{j_3}}\end{gathered}$$

In other words, $\overrightarrow{u_i}$ has, in each slot, the i^{th} tuple that makes up the index for that slot. We can construct these tuples, because $R \in \langle\Gamma\rangle$, and it is conjunction and such.

These three entries help us build this important table. This table is like what we've seen in previous lecture, but with rows instead of columns. There are only going to be three rows: those very tuples we just defined.

$$N = \begin{bmatrix} \overrightarrow{u_1} \\ \overrightarrow{u_2} \\ \overrightarrow{u_3} \end{bmatrix} \tag{9}$$

Note that if there are any two identical columns (here the columns are 3 "tall"), then we can just define $R' = \exists x_j (R \wedge (x_i = x_j))$. Clearly $R' \in \langle\Gamma\rangle$. The table cannot become empty from this. We can define the corresponding N' table.

Now we can assume that N' has no two identical columns. We permute the columns of N' to make N'', such that for all columns of the form

$$\begin{pmatrix} a \\ a \\ c \end{pmatrix} \tag{10}$$

are in the front, followed by all columns of the form

$$\begin{pmatrix} a \\ c \\ c \end{pmatrix} \tag{11}$$

and the the rest of the columns. Note that this division is always possible—the table is nonempty.

We can ensure that there is at least one column in the table corresponding to the parts in (a, a, c) and (a, c, c). Say that there are two relations $H, H' \in \Gamma$—they can be equal. There are also some tuples, $\overrightarrow{x} \in H$, $\overrightarrow{y} \in H'$ and two locations k, ℓ—again, the tuples could be equal and the indices could also be equal. And lastly say that $x_k = a, y_\ell = b$ for some $a \neq b \in A$. Then somewhere in our table there must be a triple (column-wise) of the form (a, a, a), and by symmetry (b, b, b). Because these are distinct, they would not be combined in our N' step, and so we can say the first is in our first kind of tuple in N'', and the second our second kind.

If such a difference does not exist, then every single relation still in N' must be on the same element a, so that the relation has this point on which it is true. In that case it is trivially rectangular and can easily define the Malt'sev:

$$m = \begin{cases} (a, a, a) = a \\ (b, c, c) = b \\ (b, b, c) = c \end{cases}. \tag{12}$$

For the remaining cases we can define it arbitrarily.

Now we can define relation and corresponding table N''' to take away the "other stuff" off of R''. We're just using projection, so we are still in the clone. The only thing in N''' are the triples of the form either (10) or (11).

Now we use our rectangularity hypothesis: R''' is rectangular, so if we have the situation

$$\begin{array}{ll} \overrightarrow{a} \to & \overrightarrow{c} \\ \overrightarrow{b} \to & \overrightarrow{d} \\ \overrightarrow{a} \to & \overrightarrow{d} \end{array}$$

means that $(\overrightarrow{b} \to \overrightarrow{c})$ must also exist. As $(\overrightarrow{b}, \overrightarrow{c}) \in R'''$, so $\exists(\overrightarrow{b}, \overrightarrow{c}, \star, \ldots, \star) \in R''$. Observe that we transitioned from R''' to R''. We will ultimately get all the way back to R.

Now we are ready to define our Malt'sev polymorphism, which will be $f : A^3 \to A$. We'll define in every column of N'' the image in the corresponding entry as $\overrightarrow{v}$. So $\overrightarrow{v} = (\overrightarrow{b}, \overrightarrow{c}, \star, \ldots, \star) \in R''$. Note that N'' is really the same as N'. Is this function well defined? The triples making our input are all distinct, so there's no inconsistency. While not every triple has to occur in N', we know that so far we can have a well-defined Malt'sev operation.

We define the operation:

$$\forall(a, b, c) \in N^3 f(a, b, c) = \begin{cases} c \text{ if } b = a \\ \text{a otherwise} \end{cases} . \tag{13}$$

So for all (a, a, b) or (a, b, b) that are *in* the table we are well-defined, and now for those not in the table we still work.

This reassures us that it is a Malt'sev operation, but it remains to be shown that it is a polymorphism. Fortunately, that just follows from our construction. Consider a segment of N referring to relation H. A particular column: $\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$, well, the $f(\cdot)$ application on that will of course return something in R. The $(\overrightarrow{b}, \overrightarrow{c}) \in R''$ because of rectangularity, and we can reverse-construct that back into R. If f is a polymorphism, it must be that $f(i, j, k)$ looks at an index and produces something in R, so it must be in H. □