$\mathbf{P \neq NP}$

Vinay Deolalikar
HP Research Labs, Palo Alto
vinay.deolalikar@hp.com

August 6, 2010

Abstract

We demonstrate the separation of the complexity class **NP** from its subclass **P**. Throughout our proof, we observe that the ability to compute a property on structures in polynomial time is intimately related to the statistical notions of conditional independence and sufficient statistics. The presence of conditional independencies manifests in the form of economical parametrizations of the joint distribution of covariates. In order to apply this analysis to the space of solutions of random constraint satisfaction problems, we utilize and expand upon ideas from several fields spanning logic, statistics, graphical models, random ensembles, and statistical physics.

We begin by introducing the requisite framework of graphical models for a set of interacting variables. We focus on the correspondence between Markov and Gibbs properties for directed and undirected models as reflected in the factorization of their joint distribution, and the number of independent parameters required to specify the distribution.

Next, we build the central contribution of this work. We show that there are fundamental conceptual relationships between polynomial time computation, which is completely captured by the logic $\mathsf{FO}(\mathsf{LFP})$ on some classes of structures, and certain directed Markov properties stated in terms of conditional independence and sufficient statistics. In order to demonstrate these relationships, we view a LFP computation as "factoring through" several stages of first order computations, and then utilize the limitations of first order logic. Specifically, we exploit the limitation that first order logic can only express properties in terms of a bounded number of local neighborhoods of the underlying structure.

Next we introduce ideas from the 1RSB replica symmetry breaking ansatz of statistical physics. We recollect the description of the d1RSB clustered phase for random k-SAT that arises when the clause density is sufficiently high. In this phase, an arbitrarily large fraction of all variables in cores freeze within

exponentially many clusters in the thermodynamic limit, as the clause density is increased towards the SAT-unSAT threshold for large enough k. The Hamming distance between a solution that lies in one cluster and that in another is $O(n)$.

Next, we encode k-SAT formulae as structures on which $\mathsf{FO(LFP)}$ captures polynomial time. By asking $\mathsf{FO(LFP)}$ to extend partial assignments on ensembles of random k-SAT, we build distributions of solutions. We then construct a dynamic graphical model on a product space that captures all the information flows through the various stages of a LFP computation on ensembles of k-SAT structures. Distributions computed by LFP must satisfy this model. This model is directed, which allows us to compute factorizations locally and parameterize using Gibbs potentials on cliques. We then use results from ensembles of factor graphs of random k-SAT to bound the various information flows in this directed graphical model. We parametrize the resulting distributions in a manner that demonstrates that irreducible interactions between covariates — namely, those that may not be factored any further through conditional independencies — cannot grow faster than $\mathrm{poly}(\log n)$ in the LFP computed distributions. This characterization allows us to analyze the behavior of the entire class of polynomial time algorithms on ensembles simultaneously.

Using the aforementioned limitations of LFP, we demonstrate that a purported polynomial time solution to k-SAT would result in solution space that is a mixture of distributions each having an exponentially smaller parametrization than is consistent with the highly constrained d1RSB phases of k-SAT. We show that this would contradict the behavior exhibited by the solution space in the d1RSB phase. This corresponds to the intuitive picture provided by physics about the emergence of extensive (meaning $O(n)$) long-range correlations between variables in this phase and also explains the empirical observation that all known polynomial time algorithms break down in this phase.

Our work shows that every polynomial time algorithm must fail to produce solutions to large enough problem instances of k-SAT in the d1RSB phase. This shows that polynomial time algorithms are not capable of solving $\mathbf{NP}$-complete problems in their hard phases, and demonstrates the separation of $\mathbf{P}$ from $\mathbf{NP}$.

Contents

1. Introduction

The $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ question is generally considered one of the most important and far reaching questions in contemporary mathematics and computer science.

The origin of the question seems to date back to a letter from Gödel to Von Neumann in 1956 [Sip92]. Formal definitions of the class **NP** awaited work by Edmonds [Edm65], Cook [Coo71], and Levin [Lev73]. The Cook-Levin theorem showed the existence of complete problems for this class, and demonstrated that SAT– the problem of determining whether a set of clauses of Boolean literals has a satisfying assignment – was one such problem. Later, Karp [Kar72] showed that twenty-one well known combinatorial problems, which include TRAVELLING SALESMAN, CLIQUE, and HAMILTONIAN CIRCUIT, were also **NP**-complete. In subsequent years, many problems central to diverse areas of application were shown to be **NP**-complete (see [GJ79] for a list). If $\mathbf{P} \neq \mathbf{NP}$, we could never solve these problems efficiently. If, on the other hand $\mathbf{P} = \mathbf{NP}$, the consequences would be even more stunning, since *every* one of these problems would have a polynomial time solution. The implications of this on applications such as cryptography, and on the general philosophical question of whether human creativity can be automated, would be profound.

The $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ question is also singular in the number of approaches that researchers have brought to bear upon it over the years. From the initial question in logic, the focus moved to complexity theory where early work used diagonalization and relativization techniques. However, [BGS75] showed that these methods were perhaps inadequate to resolve $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ by demonstrating relativized worlds in which $\mathbf{P} = \mathbf{NP}$ and others in which $\mathbf{P} \neq \mathbf{NP}$ (both relations for the appropriately relativized classes). This shifted the focus to methods us-

ing circuit complexity and for a while this approach was deemed the one most likely to resolve the question. Once again, a negative result in [RR97] showed that a class of techniques known as "Natural Proofs" that subsumed the above could not separate the classes **NP** and **P**, provided one-way functions exist.

Owing to the difficulty of resolving the question, and also to the negative results mentioned above, there has been speculation that resolving the $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ question might be outside the domain of mathematical techniques. More precisely, the question might be independent of standard axioms of set theory. The first such results in [HH76] show that some relativized versions of the $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ question are independent of reasonable formalizations of set theory.

The influence of the $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ question is felt in other areas of mathematics. We mention one of these, since it is central to our work. This is the area of descriptive complexity theory — the branch of finite model theory that studies the expressive power of various logics viewed through the lens of complexity theory. This field began with the result [Fag74] that showed that **NP** corresponds to queries that are expressible in second order existential logic over finite structures. Later, characterizations of the classes **P** [Imm86], [Var82] and **PSPACE** over ordered structures were also obtained.

There are several introductions to the $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ question and the enormous amount of research that it has produced. The reader is referred to [Coo06] for an introduction which also serves as the official problem description for the Clay Millenium Prize. An older excellent review is [Sip92]. See [Wig07] for a more recent introduction. Most books on theoretical computer science in general, and complexity theory in particular, also contain accounts of the problem and attempts made to resolve it. See the books [Sip96] and [BDG95] for standard references.

Preliminaries and Notation

Treatments of standard notions from complexity theory, such as definitions of the complexity classes **P**, **NP**, **PSPACE**, and notions of reductions and completeness for complexity classes, etc. may be found in [Sip96, BDG95].

Our work will span various developments in three broad areas. While we have endeavored to be relatively complete in our treatment, we feel it would be helpful to provide standard textual references for these areas, in the order in which they appear in the work. Additional references to results will be provided within the chapters.

Standard references for graphical models include [Lau96] and the more recent [KF09]. For an engaging introduction, please see [Bis06, Ch. 8]. For an early treatment in statistical mechanics of Markov random fields and Gibbs distributions, see [KS80].

Preliminaries from logic, such as notions of structure, vocabulary, first order language, models, etc., may be obtained from any standard text on logic such as [Hod93]. In particular, we refer to [EF06, Lib04] for excellent treatments of finite model theory and [Imm99] for descriptive complexity.

For a treatment of the statistical physics approach to random CSPs, we recommend [MM09]. An earlier text is [MPV87].

1.1 Synopsis of Proof

This proof requires a convergence of ideas and an interplay of principles that span several areas within mathematics and physics. This represents the majority of the effort that went into constructing the proof. Given this, we felt that it would be beneficial to explain the various stages of the proof, and highlight their interplay. The technical details of each stage are described in subsequent chapters.

Consider a system of n interacting variables such as is ubiquitous in mathematical sciences. For example, these may be the variables in a k-SAT instance that interact with each other through the clauses present in the k-SAT formula, or n Ising spins that interact with each other in a ferromagnet. Through their interaction, variables exert an influence on each other, and affect the values each other may take. The proof centers on the study of logical and algorithmic constructs where such complex interactions factor into "simpler" ones.

The factorization of interactions can be represented by a corresponding factorization of the joint distribution of the variables over the space of configurations of the n variables subject to the constraints of the problem. It has been realized in the statistics and physics communities for long that certain multivariate distributions decompose into the product of a few types of factors, with each factor itself having only a few variables. Such a factorization of joint distributions into simpler factors can often be represented by graphical models whose vertices index the variables. A factorization of the joint distribution according to the graph implies that the interactions between variables can be factored into a sequence of "local interactions" between vertices that lie within neighborhoods of each other.

Consider the case of an undirected graphical model. The factoring of interactions may be stated in terms of either a Markov property, or a Gibbs property with respect to the graph. Specifically, the *local Markov property* of such models states that the distribution of a variable is only dependent *directly* on that of its neighbors in an appropriate neighborhood system. Of course, two variables arbitrarily far apart can influence each other, *but only through a sequence of successive local interactions*. The *global Markov property* for such models states that when two sets of vertices are separated by a third, this induces a conditional independence on variables corresponding to these sets of vertices, given those corresponding to the third set. On the other hand, the *Gibbs property* of a distribution with respect to a graph asserts that the distribution factors into a product of potential functions over the maximal cliques of the graph. Each potential captures the interaction between the set of variables that form the clique. The Hammersley-Clifford theorem states that a *positive* distribution having the Markov property with respect to a graph must have the Gibbs property with respect to the same graph.

The condition of positivity is essential in the Hammersley-Clifford theorem for undirected graphs. However, it is not required when the distribution satisfies certain directed models. In that case, the Markov property with respect to the directed graph implies that the distribution factorizes into local conditional

probability distributions (CPDs). Furthermore, if the model is a directed acyclic graph (DAG), we can obtain the Gibbs property with respect to an undirected graph constructed from the DAG by a process known as *moralization*. We will return to the directed case shortly.

At this point we begin to see that factorization into conditionally independent pieces manifests in terms of economical parametrizations of the joint distribution. Thus, the *number of independent parameters* required to specify the joint distribution may be used as a measure of the complexity of interactions between the covariates. When the variates are independent, this measure takes its least value. Dependencies introduced at random (such as in random k-SAT) cause it to rise. Roughly speaking, this measure is $(O(c^k), c > 1)$ where k is the largest interaction between the variables that cannot be decomposed any further. Intuitively, we know that constraint satisfaction problems (CSPs) are hard when we cannot separate their joint constraints into smaller easily manageable pieces. This should be reflected then, in the growth of this measure on the *distribution of all solutions* to random CSPs as their constraint densities are increased. Informally, a CSP is hard (but satisfiable) when the distribution of all its solutions is complex to describe in terms of its number of independent parameters due to the extensive interactions between the variables in the CSP. Graphical models offer us a way to measure the size of these interactions.

Chapter 2 develops the principles underlying the framework of graphical models. We will not use any of these models in particular, but construct another directed model on a larger *product* space that utilizes these principles and tailors them to the case of least fixed point logic, which we turn to next.

At this point, we change to the setting of finite model theory. Finite model theory is a branch of mathematical logic that has provided machine independent characterizations of various important complexity classes including **P**, **NP**, and **PSPACE**. In particular, the class of polynomial time computable queries on *ordered structures* has a precise description — it is the class of queries expressible in the logic $\mathsf{FO}(\mathsf{LFP})$ which extends first order logic with the ability to compute least fixed points of positive first order formulae. Least fixed point

constructions iterate an underlying positive first order formula, thereby building up a relation in stages. We take a geometric picture of a LFP computation. Initially the relation to be built is empty. At the first stage, certain elements, whose types satisfy the first order formula, enter the relation. This changes the neighborhoods of these elements, and therefore in the next stage, other elements (whose neighborhoods have been thus changed in the previous stages) become eligible for entering the relation. The positivity of the formula implies that once an element is in the relation, it cannot be removed, and so the iterations reach a fixed point in a polynomial number of steps. Importantly from our point of view, the positivity and the stage-wise nature of LFP means that the computation has a *directed* representation on a graphical model that we will construct. Recall at this stage that distributions over directed models enjoy factorization even when they are not defined over the entire space of configurations.

We may interpret this as follows: LFP relies on the assumption that variables that are highly entangled with each other due to constraints can be disentangled in a way that they now interact with each other through conditional independencies induced by a certain directed graphical model construction. Of course, an element does influence others arbitrarily far away, *but only through a sequence of such successive local and bounded interactions*. The reason LFP computations terminate in polynomial time is analogous to the notions of conditional independence that underlie efficient algorithms on graphical models having sufficient factorization into local interactions.

In order to apply this picture in full generality to all LFP computations, we use the simultaneous induction lemma to push all simultaneous inductions into nested ones, and then employ the transitivity theorem to encode nested fixed points as sections of a single relation of higher arity. Finally, we either do the extra bookkeeping to work with relations of higher arity, or work in a larger structure where the relation of higher arity is monadic (namely, structures of k-types of the original structure). Either of these cases presents only a polynomially larger overhead, and does not hamper our proof scheme. Building the machinery that can precisely map all these cases to the picture of factorization

into local interactions is the subject of Chapter 4.

The preceding insights now direct us to the setting necessary in order to separate **P** from **NP**. We need a regime of **NP**-complete problems where interactions between variables are so "dense" that they cannot be factored through the bottleneck of the local and bounded properties of first order logic that limit each stage of LFP computation. Intuitively, this should happen when each variable has to simultaneously satisfy constraints involving an extensive ($O(n)$) fraction of the variables in the problem.

In search of regimes where such situations arise, we turn to the study of ensemble random k-SAT where the properties of the ensemble are studied as a function of the clause density parameter. We will now add ideas from this field which lies on the intersection of statistical mechanics and computer science to the set of ideas in the proof.

In the past two decades, the phase changes in the solution geometry of random k-SAT ensembles as the clause density increases, have gathered much research attention. The 1RSB ansatz of statistical mechanics says that the space of solutions of random k-SAT shatters into exponentially many clusters of solutions when the clause density is sufficiently high. This phase is called 1dRSB (1-Step Dynamic Replica Symmetry Breaking) and was conjectured by physicists as part of the 1RSB ansatz. It has since been rigorously proved for high values of k. It demonstrates the properties of high correlation between large sets of variables that we will need. Specifically, the emergence of *cores* that are sets of C clauses all of whose variables lie in a set of size C (this actually forces C to be $O(n)$). As the clause density is increased, the variables in these cores "freeze." Namely, they take the same value throughout the cluster. Changing the value of a variable within a cluster necessitates changing $O(n)$ other variables in order to arrive at another satisfying solution, which would be in a different cluster. Furthermore, as the clause density is increased towards the SAT-unSAT threshold, each cluster collapses steadily towards a single solution, that is maximally far apart from every other cluster. Physicists think of this as an "energy gap" between the clusters. Such stages are precisely the ones that cannot be factored

through local and bounded first order stages of a LFP computation due to the tight coupling between $O(n)$ variables. Finally, as the clause density increases above the SAT-unSAT threshold, the solution space vanishes, and the underlying instance of SAT is no longer satisfiable. We reproduce the rigorously proved picture of the 1RSB ansatz that we will need in Chapter 5.

In Chapter 6, we make a brief excursion into the random graph theory of the factor graph ensembles underlying random k-SAT. From here, we obtain results that asymptotically almost surely upper bound the size of the largest cliques in the neighborhood systems on the Gaifman graphs that we study later. These provide us with bounds on the largest irreducible interactions between variables during the various stages of an LFP computation.

Finally in Chapter 7, we pull all the threads and machinery together. First, we encode k-SAT instances as queries on structures over a certain vocabulary in a way that LFP captures all polynomial time computable queries on them. We then set up the framework whereby we can generate distributions of solutions to each instance by asking a purported LFP algorithm for k-SAT to extend partial assignments on variables to full satisfying assignments.

Next, we embed the space of covariates into a larger *product space* which allows us to "disentangle" the flow of information during a LFP computation. This allows us to study the computations performed by the LFP with various initial values under a directed graphical model. This model is only polynomially larger than the structure itself. We call this the *Element-Neighborhood-Stage Product*, or ENSP model. The distribution of solutions generated by LFP then is a mixture of distributions each of whom factors according to an ENSP.

At this point, we wish to measure the growth of *independent parameters* of distributions of solutions whose embeddings into the larger product space factor over the ENSP. In order to do so, we utilize the following properties.

1. The directed nature of the model that comes from properties of LFP.

2. The properties of neighborhoods that are obtained by studies on random graph ensembles, specifically that neighborhoods that occur during the

LFP computation are of size $\mathrm{poly}(\log n)$ asymptotically almost surely in the $n \to \infty$ limit.

3. The locality and boundedness properties of FO that put constraints upon each individual stage of the LFP computation.

4. Simple properties of LFP, such as the closure ordinal being a polynomial in the structure size.

The crucial property that allows us to analyze mixtures of distributions that factor according to some ENSP is that we can parametrize the distribution using potentials on cliques of its moralized graph that are of size at most $\mathrm{poly}(\log n)$. This means that when the mixture is exponentially numerous, we will see features that reflect the $\mathrm{poly}(\log n)$ factor size of the conditionally independent parametrization.

Now we close the loop and show that a distribution of solutions for SAT with these properties would contradict the known picture of k-SAT in the d1RSB phase for $k > 8$ — namely, the presence of extensive frozen variables in exponentially many clusters with Hamming distance between the clusters being $O(n)$. In particular, in exponentially numerous mixtures, we would have conditionally independent variation between blocks of $\mathrm{poly}(\log n)$ variables, causing the Hamming distance between solutions to be of this order as well. In other words, solutions for k-SAT that are constructed using LFP will display aggregate behavior that reflects that they are constructed out of "building blocks" of size $\mathrm{poly}(\log n)$. This behavior will manifest when exponentially many solutions are generated by the LFP construction.

This shows that LFP cannot express the satisfiability query in the d1RSB phase for high enough k, and separates $\mathbf{P}$ from $\mathbf{NP}$. This also explains the empirical observation that all known polynomial time algorithms fail in the d1RSB phase for high values of k, and also establishes on rigorous principles the physics intuition about the onset of extensive long range correlations in the d1RSB phase that causes all known polynomial time algorithms to fail.

2. Interaction Models and Conditional Independence

Systems involving a large number of variables interacting in complex ways are ubiquitous in the mathematical sciences. These interactions induce dependencies between the variables. Because of the presence of such dependencies in a complex system with interacting variables, it is not often that one encounters independence between variables. However, one frequently encounters *conditional independence* between sets of variables. Both independence and conditional independence among sets of variables have been standard objects of study in probability and statistics. Speaking in terms of algorithmic complexity, one often hopes that by exploiting the conditional independence between certain sets of variables, one may avoid the cost of enumeration of an exponential number of hypothesis in evaluating functions of the distribution that are of interest.

2.1 Conditional Independence

We first fix some notation. Random variables will be denoted by upper case letters such as X, Y, Z, etc. The values a random variable takes will be denoted by the corresponding lower case letters, such as x, y, z. Throughout this work, we assume our random variables to be discrete unless stated otherwise. We may also assume that they take values in a common finite state space, which we usually denote by Λ following physics convention. We denote the probability mass functions of discrete random variables X, Y, Z by $P_X(x), P_Y(y), P_Z(z)$ respectively. Similarly, $P_{X,Y}(x, y)$ will denote the joint mass of (X, Y), and so

on. We drop subscripts on the P when it causes no confusion. We freely use the term "distribution" for the probability mass function.

The notion of conditional independence is central to our proof. The intuitive definition of the conditional independence of X from Y given Z is that the conditional distribution of X given (Y, Z) is equal to the conditional distribution of X given Z alone. This means that once the value of Z is given, no further information about the value of X can be extracted from the value of Y. This is an asymmetric definition, and can be replaced by the following symmetric definition. Recall that X is independent of Y if

$$P(x, y) = P(x)P(y).$$

Definition 2.1. Let notation be as above. X is *conditionally independent* of Y given Z, written $X \perp\!\!\!\perp Y \mid Z$, if

$$P(x, y \mid z) = P(x \mid z)P(y \mid z),$$

The asymmetric version which says that the information contained in Y is superfluous to determining the value of X once the value of Z is known may be represented as

$$P(xcondy, z) = P(x \mid z).$$

The notion of conditional independence pervades statistical theory [Daw79, Daw80]. Several notions from statistics may be recast in this language.

EXAMPLE 2.2. The notion of sufficiency may be seen as the presence of a certain conditional independence [Daw79]. A *sufficient statistic* T in the problem of parameter estimation is that which renders the estimate of the parameter independent of any further information from the sample X. Thus, if Θ is the parameter to be estimated, then T is a sufficient statistic if

$$P(\theta \mid x) = P(\theta \mid t).$$

Thus, all there is to be gained from the sample in terms of information about Θ is already present in T alone. In particular, if Θ is a posterior that is being

computed by Bayesian inference, then the above relation says that the posterior depends on the data X through the value of T alone. Clearly, such a statement would lead to a reduction in the complexity of inference.

2.2 Conditional Independence in Undirected Graphical Models

Graphical models offer a convenient framework and methodology to describe and exploit conditional independence between sets of variables in a system. One may think of the graphical model as representing the family of distributions whose law fulfills the conditional independence statements made by the graph. A member of this family may satisfy any number of additional conditional independence statements, but not less than those prescribed by the graph. In general, we will consider graphs $\mathcal{G} = (V, E)$ whose n vertices index a set of n random variables $(X_1, \ldots, X_n)$. The random variables all take their values in a common state space Λ. The random vector $(X_1, \ldots, X_n)$ then takes values in a *configuration space* $\Omega_n = \Lambda^n$. We will denote values of the random vector $(X_1, \ldots, X_n)$ simply by $x = (x_1, \ldots, x_n)$. The notation $X_{V \setminus I}$ will denote the set of variables excluding those whose indices lie in the set I. Let P be a probability measure on the configuration space. We will study the interplay between conditional independence properties of P and its factorization properties.

There are, broadly, two kinds of graphical models: directed and undirected. We first consider the case of undirected models. Fig. 2.1 illustrates an undirected graphical model with ten variables.

Random Fields and Markov Properties

Graphical models are very useful because they allow us to read off conditional independencies of the distributions that satisfy these models from the graph itself. Recall that we wish to study the relation between conditional independence of a distribution with respect to a graphical model, and its factorization.

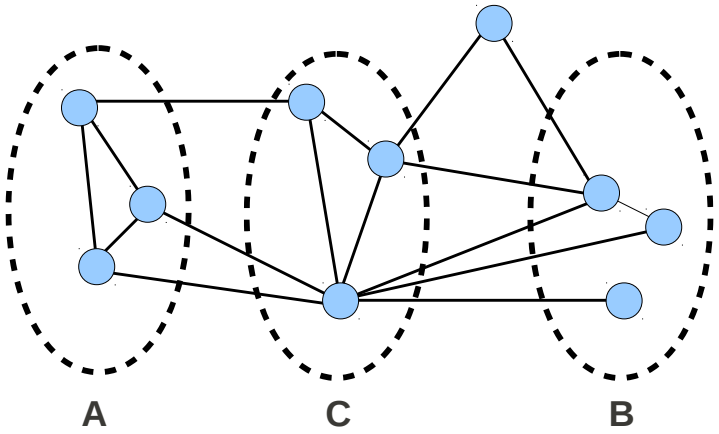

Figure 2.1: An undirected graphical model. Each vertex represents a random variable. The vertices in set A are separated from those in set B by set C. For random variables to satisfy the global Markov property relative to this graphical model, the corresponding sets of random variables must be conditionally independent. Namely, $A \perp\!\!\!\perp B \mid C$.

Towards that end, one may write increasingly stringent conditional independence properties that a set of random variables satisfying a graphical model may possess, with respect to the graph. In order to state these, we first define two graph theoretic notions — those of a general neighborhood system, and of separation.

Definition 2.3. Given a set of variables S known as *sites*, a *neighborhood system* $\mathcal{N}_S$ on S is a collection of subsets $\{\mathcal{N}_i \colon 1 \leq i \leq n\}$ indexed by the sites in S that satisfy

1. a site is not a neighbor to itself (this also means there are no self-loops in the induced graph): $s_i \notin \mathcal{N}_i$, and

2. the relationship of being a neighbor is mutual: $s_i \in \mathcal{N}_j \Leftrightarrow s_j \in \mathcal{N}_i$.

In many applications, the sites are vertices on a graph, and the neighborhood system $\mathcal{N}_i$ is the set of neighbors of vertex s_i on the graph. We will often be interested in *homogeneous neighborhood systems* of S on a graph in which, for

each $s_i \in S$, the neighborhood $\mathcal{N}_i$ is defined as

$$\mathcal{G}_i := \{s_j \in S \colon d(s_i, s_j) \leq r\}.$$

Namely, in such neighborhood systems, the neighborhood of a site is simply the set of sites that lie in the radius r ball around that site. Note that a *nearest neighbor system* that is often used in physics is just the case of $r = 1$. We will need to use the general case, where r will be determined by considerations from logic that will be introduced in the next two chapters. We will use the term "variable" freely in place of "site" when we move to logic.

Definition 2.4. Let A, B, C be three disjoint subsets of the vertices V of a graph $\mathcal{G}$. The set C is said to *separate* A and B if every path from a vertex in A to a vertex in B must pass through C.

Now we return to the case of the vertices indexing random variables $(X_1, \ldots, X_n)$ and the vector $(X_1, \ldots, X_n)$ taking values in a configuration space Ω_n. A probability measure P on Ω_n is said to satisfy certain Markov properties with respect to the graph when it satisfies the appropriate conditional independencies with respect to that graph. We will study the following two Markov properties, and their relation to factorization of the distribution.

Definition 2.5.

1. *The local Markov property.* The distribution X_i (for every i) is conditionally independent of the rest of the graph given just the variables that lie in the neighborhood of the vertex. In other words, the influence that variables exert on any given variable is completely described by the influence that is exerted through the neighborhood variables alone.

2. *The global Markov property.* For any disjoint subsets A, B, C of V such that C separates A from B in the graph, it holds that

$$A \perp\!\!\!\perp B \,|\, C.$$

We are interested in distributions that do satisfy such properties, and will examine what effect these Markov properties have on the factorization of the

distributions. For most applications, this is done in the context of *Markov random fields*.

We motivate a Markov random field with the simple example of a Markov chain $\{X_n \colon n \geq 0\}$. The Markov property of this chain is that any variable in the chain is conditionally independent of all other variables in the chain given just its immediate neighbors:

$$X_n \perp\!\!\!\perp \{x_k \colon k \notin \{n-1, n, n+1\} \mid X_{n-1}, X_{n+1}\}.$$

A Markov random field is the natural generalization of this picture to higher dimensions and more general neighborhood systems.

Definition 2.6. The collection of random variables $X_1, \ldots, X_n$ is a *Markov random field* with respect to a neighborhood system on $\mathcal{G}$ if and only if the following two conditions are satisfied.

1. The distribution is positive on the space of configurations: $P(x) > 0$ for $x \in \Omega_n$.

2. The distribution at each vertex is conditionally independent of all other vertices given just those in its neighborhood:

$$P(X_i \mid X_{V \setminus i}) = P(X_i \mid X_{\mathcal{N}_i})$$

   These local conditional distributions are known as *local characteristics* of the field.

The second condition says that Markov random fields satisfy the local Markov property with respect to the neighborhood system. Thus, we can think of interactions between variables in Markov random fields as being characterized by "piecewise local" interactions. Namely, the influence of far away vertices must "factor through" local interactions. This may be interpreted as:

> *The influence of far away variables is limited to that which is transmitted through the interspersed intermediate variables — there is no "direct" influence of far away vertices beyond that which is factored through such intermediate interactions.*

However, through such local interactions, a vertex may influence any other arbitrarily far away. Notice though, that this is a considerably simpler picture than having to consult the joint distribution over all variables for all interactions, for here, we need only know the local joint distributions and use these to infer the correlations of far away variables. We shall see in later chapters that this picture, with some additional caveats, is at the heart of polynomial time computations.

Note the positivity condition on Markov random fields. With this positivity condition, the complete set of conditionals given by the local characteristics of a field determine the joint distribution [Bes74].

Markov random fields satisfy the global Markov property as well.

Theorem 2.7. *Markov random fields with respect to a neighborhood system satisfy the global Markov property with respect to the graph constructed from the neighborhood system.*

Markov random fields originated in statistical mechanics [Dob68], where they model probability measures on configurations of interacting particles, such as Ising spins. See [KS80] for a treatment that focusses on this setting. Their local properties were later found to have applications to analysis of images and other systems that can be modelled through some form of spatial interaction. This field started with [Bes74] and came into its own with [GG84] which exploited the Markov-Gibbs correspondence that we will deal with shortly. See also [Li09].

2.2.1 Gibbs Random Fields and the Hammersley-Clifford Theorem

We are interested in how the Markov properties of the previous section translate into factorization of the distribution. Note that Markov random fields are characterized by a local condition — namely, their local conditional independence characteristics. We now describe another random field that has a global characterization — the Gibbs random field.

Definition 2.8. A *Gibbs random field* (or *Gibbs distribution*) with respect to a neighborhood system $\mathcal{N}_{\mathcal{G}}$ on the graph $\mathcal{G}$ is a probability measure on the set of configurations Ω_n having a representation of the form

$$P(x_1, \ldots, x_n) = \frac{1}{Z} \exp(-\frac{U(x)}{T}),$$

where

1. Z is the *partition function* and is a normalizing factor that ensures that the measure sums to unity,

$$Z = \sum_{x \in \Omega_n} \exp(-\frac{U(x)}{T}).$$

   Evaluating Z explicitly is hard in general since it is a summation over each of the Λ^n configurations in the space.

2. T is a constant known as the "Temperature" that has origins in statistical mechanics. It controls the sharpness of the distribution. At high temperatures, the distribution tends to be uniform over the configurations. At low temperatures, it tends towards a distribution that is supported only on the lowest energy states.

3. $U(x)$ is the "energy" of configuration x and takes the following form as a sum

$$U(x) = \sum_{c \in \mathcal{C}} V_c(x).$$

   over the set of cliques $\mathcal{C}$ of $\mathcal{G}$. The functions $V_c \colon c \in \mathcal{C}$ are the *clique potentials* such that the value of $V_c(x)$ depends only on the coordinates of x that lie in the clique c. These capture the interactions between vertices in the clique.

Thus, a Gibbs random field has a probability distribution that factorizes into its constituent "interaction potentials." This says that the probability of a configuration depends only on the interactions that occur between the variables, broken up into cliques. For example, let us say that in a system, each particle

interacts with only 2 other particles at a time, (if one prefers to think in terms of statistical mechanics) then the energy of each state would be expressible as a sum of potentials, each of whom had just three variables in its support. Thus, the Gibbs factorization carries in it a faithful representation of the underlying interactions between the particles. This type of factorization obviously yields a "simpler description" of the distribution. The precise notion is that of *independent parameters* it takes to specify the distribution. Factorization into conditionally independent interactions of scope k means that we can specify the distribution in $O(\gamma^k)$ parameters rather than $O(\gamma^n)$. We will return to this at the end of this chapter.

Definition 2.9. Let P be a Gibbs distribution whose energy function $U(x) = \sum_{c \in \mathcal{C}} V_c(x)$. The *support* of the potential V_c is the cardinality of the clique c. The *degree* of the distribution P, denoted by $\deg(P)$, is the maximum of the supports of the potentials. In other words, the degree of the distribution is the size of the largest clique that occurs in its factorization.

One may immediately see that the degree of a distribution is a measure of the complexity of interactions in the system since it is the size of the largest set of variables whose interaction cannot be split up in terms of smaller interactions between subsets. One would expect this to be the hurdle in efficient algorithmic applications.

The Hammersley-Clifford theorem relates the two types of random fields.

Theorem 2.10 (Hammersley-Clifford)**.** *X is Markov random field with respect to a neighborhood system $\mathcal{N}_\mathcal{G}$ on the graph $\mathcal{G}$ if and only if it is a Gibbs random field with respect to the same neighborhood system.*

The theorem appears in the unpublished manuscript [HC71] and uses a certain "blackening algebra" in the proof. The first published proofs appear in [Bes74] and [Mou74].

Note that the condition of positivity on the distribution (which is part of the definition of a Markov random field) is essential to state the theorem in full generality. The following example from [Mou74] shows that relaxing this

condition allows us to build distributions having the Markov property, but not the Gibbs property.

EXAMPLE 2.11. Consider a system of four binary variables $\{X_1, X_2, X_3, X_4\}$. Each of the following combinations have probability $1/8$, while the remaining combinations are disallowed.

$$(0,0,0,0) \quad (1,0,0,0) \quad (1,1,0,0) \quad (1,1,1,0)$$
$$(0,0,0,1) \quad (0,0,1,1) \quad (0,1,1,1) \quad (1,1,1,1).$$

We may check that this distribution has the global Markov property with respect to the 4 vertex cycle graph. Namely we have

$$X_1 \perp\!\!\!\perp X_3 \,|\, X_2, X_4 \text{ and } X_2 \perp\!\!\!\perp X_4 \,|\, X_1, X_3.$$

However, the distribution does not factorize into Gibbs potentials.

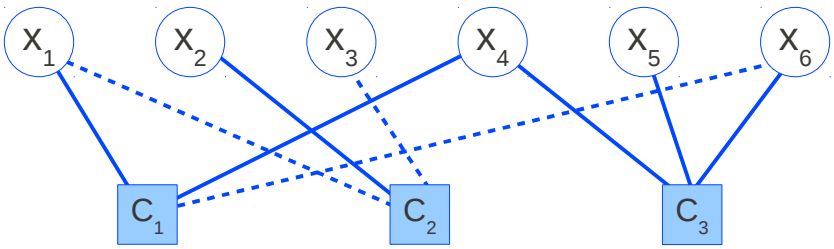

Figure 2.2: A factor graph showing the three clause 3-SAT formula $(X_1 \vee X_4 \vee \neg X_6) \wedge (\neg X_1 \vee X_2 \vee \neg X_3) \wedge (X_4 \vee X_5 \vee X_6)$. A dashed line indicates that the variable appears negated in the clause.

The distribution modelled by this factor graph will show a factorization as follows

$$p(x_1, \ldots, x_6) = \frac{1}{Z}\varphi_1(x_1, x_4, x_6)\varphi_2(x_1, x_2, x_3)\varphi(x_4, x_5, x_6), \tag{2.1}$$

$$\text{where } Z = \sum_{x_1,\ldots,x_6} \varphi_1(x_1, x_4, x_6)\varphi_2(x_1, x_2, x_3)\varphi(x_4, x_5, x_6). \tag{2.2}$$

Factor graphs offer a finer grained view of factorization of a distribution than Bayesian networks or Markov networks. One should keep in mind that this factorization is (in general) far from being a factorization into conditionals and does not express conditional independence. The system must embed each of these factors in ways that are global and not obvious from the factors. This global information is contained in the partition function. Thus, in general, these factors do not represent conditionally independent pieces of the joint distributions. In summary, the factorization above is not the one what we are seeking — it does not imply a series of conditional independencies in the joint distribution.

Factor graphs have been very useful in various applications, most notably perhaps in coding theory where they are used as graphical models that underlie various decoding algorithms based on forms of belief propagation (also known as the sum-product algorithm) that is an exact algorithm for computing marginals on tree graphs but performs remarkably well even in the presence of loops. See [KFaL98] and [AM00] for surveys of this field. As might be expected from the preceding comments, these do not focus on conditional independence, but rather on algorithmic applications of local features (such as locally tree like) of factor graphs.

A Hammersley-Clifford type theorem holds over the *completion* of a factor graph. A *clique* in a factor graph is a set of variable nodes such that every pair in the set is connected by a function node. The completion of a factor graph is obtained by introducing a new function node for each clique, and connecting it to all the variable nodes in the clique, and no others. Then, a positive distribution that satisfies the global Markov property with respect to a factor graph satisfies the Gibbs property with respect to its completion.

2.4 The Markov-Gibbs Correspondence for Directed Models

Consider first a directed acyclic graph (DAG), which is simply a directed graph without any directed cycles in it. Some specific points of additional terminology for directed graphs are as follows. If there is a directed edge from x to y, we say that x is a *parent* of y, and y is the child of x. The set of parents of x is denoted by $\mathrm{pa}(x)$, while the set of children of x is denoted by $\mathrm{ch}(a)$. The set of vertices from whom directed paths lead to x is called the *ancestor set* of x and is denoted $\mathrm{an}(x)$. Similarly, the set of vertices to whom directed paths from x lead is called the *descendant set* of x and is denoted $\mathrm{de}(x)$. Note that DAGs is allowed to have loops (and loopy DAGs are central to the study of iterative decoding algorithms on graphical models). Finally, we often assume that the graph is equipped with a distance function $d(\cdot, \cdot)$ between vertices which is just the length of the shortest path between them. A set of random variables whose interdependencies may be represented using a DAG is known as a *Bayesian network* or a *directed Markov field*. The idea is best illustrated with a simple example.

Consider the DAG of Fig. 2.3 (left). The corresponding factorization of the joint density that is induced by the DAG model is

$$p(x_1, \ldots, x_6) = p(x_1)p(x_2)p(x_3)p(x_4 \,|\, x_1)p(x_5 \,|\, x_2, x_3, x_4).$$

Thus, every joint distribution that satisfies this DAG factorizes as above.

Given a directed graphical model, one may construct an undirected one by a process known as *moralization*. In moralization, we (a) replace a directed edge from one vertex to another by an undirected one between the same two vertices and (b) "marry" the parents of each vertex by introducing edges between each pair of parents of the vertex at the head of the former directed edge. The process is illustrated in the figure below.

In general, if we denote the set of parents of the variable x_i by $\mathrm{pa}(x_i)$, then

Figure 2.3: The moralization of the DAG on the left to obtain the moralized undirected graph on the right.

the joint distribution of $(x_1, \ldots, x_n)$ factorizes as

$$p(x_1, \ldots, x_n) = \prod_{n=1}^{N} p(x_n \mid \mathrm{pa}_n).$$

We want, however, is to obtain a Markov-Gibbs equivalence for such graphical models in the same manner that the Hammersley-Clifford theorem provided for positive Markov random fields. We have seen that relaxing the positivity condition on the distribution in the Hammersley-Clifford theorem (Thm. 2.10) cannot be done in general. In some cases however, one may remove the positivity condition safely. In particular, [LDLL90] extends the Hammersley-Clifford correspondence to the case of arbitrary distributions (namely, dropping the positivity requirement) for the case of directed Markov fields. In doing so, they simplify and strengthen an earlier criterion for directed graphs given by [KSC84]. We will use the result from [LDLL90], which we reproduce next.

Definition 2.12. A measure p admits a *recursive factorization* according to graph $\mathcal{G}$ if there exist non-negative functions, known as *kernels*, $k^v(.,.)$ for $v \in V$ defined on $\Lambda \times \Lambda^{|\,\mathrm{pa}(v)|}$ where the first factor is the state space for X_v and the second for $X_{\mathrm{pa}(v)}$, such that

$$\int k^v(y_v, x_{\mathrm{pa}(v)}) \mu_v(dy_v) = 1$$

and

$$p = f.\mu \text{ where } f(x) = \prod_{v \in V} k^v(x_v, x_{\mathrm{pa}(v)}).$$

In this case, the kernels $k^v(., x_{\mathrm{pa}(v)})$ are the conditional densities for the distribution of X_v conditioned on the value of its parents $X_{\mathrm{pa}(v)} = x_{\mathrm{pa}(v)}$. Now let $\mathcal{G}^m$ be the moral graph corresponding to $\mathcal{G}$.

Theorem 2.13. *If p admits a recursive factorization according to $\mathcal{G}$, then it admits a factorization (into potentials) according to the moral graph $\mathcal{G}^m$.*

D-separation

We have considered the notion of separation on undirected models and its effect on the set of conditional independencies satisfied by the distributions that factor according to the model. For directed models, there is an analogous notion of separation known as *D-separation*. The notion is what one would expect intuitively if one views directed models as representing "flows" of probabilistic influence.

We simply state the property and refer the reader to [KF09, §3.3.1] and [Bis06, §8.2.2] for discussion and examples. Let A,B, and C be sets of vertices on a directed model. Consider the set of all directed paths coming from a node in A and going to a node in B. Such a path is said to be *blocked* if one of the following two scenarios occurs.

1. Arrows on the path meet head-to-tail or tail-to-tail at a node in C.
2. Arrows meet head-to-head at a node, and neither the node nor any of its descendants is in C.

If all paths from A to B are blocked as above, then C is said to *D-separate A* from B, and the joint distribution must satisfy $A \perp\!\!\!\perp B \mid C$.

2.5 $\mathcal{I}$-maps and $\mathcal{D}$-maps

We have seen that there are two broad classes of graphical models — undirected and directed — which may be used to represent the interaction of variables in a system. The conditional independence properties of these two classes are obtained differently.

Definition 2.14. A graph (directed or undirected) is said to be a *$\mathcal{D}$-map* ('dependencies map') for a distribution if every conditional independence statement of the form $A \perp\!\!\!\perp B \mid C$ for sets of variables A, B, and C that is satisfied by the distribution is reflected in the graph. Thus, a completely disconnected graph having no edges is trivially a $\mathcal{D}$-map for any distribution.

A $\mathcal{D}$-map may express more conditional independencies than the distribution possesses.

Definition 2.15. A graph (directed or undirected) is said to be a *$\mathcal{I}$-map* ('independencies map') for a distribution if every conditional independence statement of the form $A \perp\!\!\!\perp B \mid C$ for sets of variables A, B, and C that is expressed by the graph is also satisfied by the distribution. Thus, a completely connected graph is trivially a $\mathcal{I}$-map for any distribution.

A $\mathcal{I}$-map may express less conditional independencies than the distribution possesses.

Definition 2.16. A graph that is both an $\mathcal{I}$-map and a $\mathcal{D}$-map for a distribution is called its *$\mathcal{P}$-map* ('perfect man').

In other words a $\mathcal{P}$-map expresses precisely the set of conditional independencies that are present in the distribution.

Not all distributions have $\mathcal{P}$-maps. Indeed, the class of distributions having directed $\mathcal{P}$-maps is itself distinct from the class having undirected $\mathcal{P}$-maps and neither equals the class of all distributions (see [Bis06, §3.8.4] for examples).

2.6 Parametrization

We now come to a central theme in our work. Consider a system of n binary covariates $(X_1, \ldots, X_n)$. To specify their joint distribution $p(x_1, \ldots, x_n)$ completely in the absence of any additional information, we would have to give the probability mass function at each of the 2^n configurations that these n variables can take jointly. The only constraint we have on these probability masses is that they must sum up to 1. Thus, if we had the function value at $2^n - 1$ configurations, we could find that at the remaining configuration. This means that in the absence of any additional information, n covariates require $2^n - 1$ parameters to specify their joint distribution.

Compare this to the case where we are provided with one critical piece of extra information — that the n variates are independent of each other. In that case, we would need 1 parameter to specify each of their individual distributions — namely, the probability that it takes the value 1. These n parameters then specify the joint distribution simply because the distribution *factorizes* completely into factors whose scopes are single variables (namely, just the $p(x_i)$), as a result of the independence. Thus, we go from exponentially many independent parameters to linearly many if we know that the variates are independent.

As noted earlier, it is not often that complex systems of n interacting variables have complete independence between some subsets. What is far more frequent is that there are *conditional independencies* between certain subsets given some intermediate subset. In this case, the joint will factorize into factors each of whose scope is a subset of $(X_1, \ldots, X_n)$. If the factorization is into conditionally independent factors, each of whose scope is of size at most k , then we can parametrize the joint distribution with at most $n2^n$ independent parameters. We should emphasize that the factors must give us conditional independence for this to be true. For example, factor graphs give us a factorization, but it is, in general, not a factorization into conditional independents, and so we cannot conclude anything about the number of independent parameters by just examining the factor graph. From our perspective, a major feature of directed graphical models is that their factorizations are already globally normalized once they

are locally normalized, meaning that there is a recursive factorization of the joint into conditionally independent pieces. The conditional independence in this case is from all non-descendants, given the parents. Therefore, if each node has at most k parents, we can parametrize the distribution using at most $n2^k$ independent parameters. We may also moralize the graph and see this as a factorization over cliques in the moralized graph. Note that such a factorization (namely, starting from a directed model and moralizing) holds even if the distribution is not positive in contrast with those distributions which do not factor over directed models and where we have to invoke the Hammersley-Clifford theorem to get a similar factorization. See [KF09] for further discussion on parameterizations for directed and undirected graphical models.

Our proof scheme aims to distinguish distributions based on the size of the irreducible direct interactions between subsets of the covariates. Namely, we would like to distinguish distributions where there are $O(n)$ such covariates whose joint interaction cannot be factored through smaller interactions (having less than $O(n)$ covariates) chained together by conditional independencies. We would like to contrast such distributions from others which *can* be so factored through factors having only $\mathrm{poly}(\log n)$ variates in their scope. The measure that we have which allows us to make this distinction is the number of independent parameters it takes to specify the distribution. When the size of the smallest irreducible interactions is $O(n)$, then we need $O(c^n)$ parameters where $c > 1$. On the other hand, if we were able to demonstrate that the distribution factors through interactions which always have scope $\mathrm{poly}(\log n)$, then we would need only $O(c^{\mathrm{poly}(\log n)})$ parameters.

Let us consider the example of a Markov random field. By Hammersley-Clifford, it is also a Gibbs random field over the set of maximal cliques in the graph encoding the neighborhood system of the Markov random field. This Gibbs field comes with conditional independence assurance, and therefore, we have an upper bound on the number of parameters it takes to specify the distribution. Namely, it is just $\sum_{c \in \mathcal{C}} 2^{|c|}$. Thus, if at most $k < n$ variables interact directly at a time, then the largest clique size would be k, and this would give

us a more economical parameterization than the one which requires $2^n - 1$ parameters.

In this work, we will build machinery that shows that if a problem lies in $\mathbf{P}$, the factorization of the distribution of solutions to that problem causes it to have economical parametrization, precisely because variables do not interact all at once, but rather in smaller subsets in a directed manner that gives us conditional independencies between sets that are of size $\mathrm{poly}(\log n)$.

We now begin the process of building that machinery.

3. Logical Descriptions of Computations

Work in finite model theory and descriptive complexity theory — a branch of finite model theory that studies the expressive power of various logics in terms of complexity classes — has resulted in machine independent characterizations of various complexity classes. In particular, over ordered structures, there is a precise and highly insightful characterization of the class of queries that are computable in polynomial time, and those that are computable in polynomial space. In order to keep the treatment relatively complete, we begin with a brief précis of this theory. Readers from a finite model theory background may skip this chapter.

We quickly set notation. A *vocabulary*, denoted by σ, is a set consisting of finitely many relation and constant symbols,

$$\sigma = \langle R_1, \ldots, R_m, c_1, \ldots, c_s \rangle.$$

Each relation has a fixed arity. We consider only *relational* vocabularies in that there are no function symbols. This poses no shortcomings since functions may be encoded as relations. A *σ-structure* $\mathfrak{A}$ consists of a set A which is the *universe* of $\mathfrak{A}$, interpretations $R^{\mathfrak{A}}$ for each of the relation symbols in the vocabulary, and interpretations $c^{\mathfrak{A}}$ for each of the constant symbols in the vocabulary. Namely,

$$\mathfrak{A} = \langle A, R_1^{\mathfrak{A}}, \ldots, R_m^{\mathfrak{A}}, c_1^{\mathfrak{A}}, \ldots, c_s^{\mathfrak{A}} \rangle.$$

An example is the vocabulary of graphs which consists of a single relation symbol having arity two. Then, a graph may be seen as a structure over this

vocabulary, where the universe is the set of nodes, and the relation symbol is interpreted as an edge. In addition, some applications may require us to work with a graph vocabulary having two constants interpreted in the structure as source and sink nodes respectively.

We also denote by σ_n the extension of σ by n additional constants, and denote by $(\mathfrak{A}, \mathbf{a})$ the structure where the tuple $\mathbf{a}$ has been identified with these additional constants.

3.1 Inductive Definitions and Fixed Points

The material in this section is standard, and we refer the reader to [Mos74] for the first monograph on the subject, and to [EF06, Lib04] for detailed treatments in the context of finite model theory. See [Imm99] for a text on descriptive complexity theory. Our treatment is taken mostly from these sources, and stresses the facts we need.

Inductive definitions are a fundamental primitive of mathematics. The idea is to build up a set in *stages*, where the defining relation for each stage can be written in the *first order* language of the underlying structure and uses elements added to the set in previous stages. In the most general case, there is an underlying structure $\mathfrak{A} = \langle A, R_1, \ldots, R_m \rangle$ and a formula

$$\phi(S, \mathbf{x}) \equiv \phi(S, x_1, \ldots, x_n)$$

in the first-order language of $\mathfrak{A}$. The variable S is a second-order relation variable that will eventually hold the set we are trying to build up in stages. At the ξ^{th} stage of the induction, denoted by I_ϕ^ξ, we insert into the relation S the tuples according to

$$\mathbf{x} \in I_\phi^\xi \Leftrightarrow \phi(\bigcup_{\eta<\xi} I_\phi^\eta, \mathbf{x}).$$

We will denote the stage that a tuple enters the relation in the induction defined by ϕ by $|\cdot|_\phi^{\mathfrak{A}}$. The decomposition into its various stages is a central characteristic of inductively defined relations. We will also require that ϕ have only positive occurrences of the n-ary relation variable S, namely all occurrences of S be

within the scope of an even number of negations. Such inductions are called *positive elementary.* In the most general case, a transfinite induction may result. The least ordinal κ at which $I_\phi^\kappa = I_\phi^{\kappa+1}$ is called the *closure ordinal* of the induction, and is denoted by $|\phi^{\mathfrak{A}}|$. When the underlying structures are finite, this is also known as the *inductive depth*. Note that the cardinality of the ordinal κ is at most $|A|^n$.

Finally, we define the relation

$$I_\phi = \bigcup_\xi I_\phi^\xi.$$

Sets of the form I_ϕ are known as *fixed points* of the structure. Relations that may be defined by

$$R(\mathbf{x}) \Leftrightarrow I_\phi(\mathbf{a}, \mathbf{x})$$

for some choice of tuple $\mathbf{a}$ over A are known as *inductive relations*. Thus, inductive relations are sections of fixed points.

Note that there are definitions of the set I_ϕ that are equivalent, but can be stated only in the second order language of $\mathfrak{A}$. Note that the definition above is

1. elementary at each stage, and

2. constructive.

We will use both these properties throughout our work.

We now proceed more formally by introducing operators and their fixed points, and then consider the operators on structures that are induced by first order formulae. We begin by defining two classes of operators on sets.

Definition 3.1. Let A be a finite set, and $\mathcal{P}(A)$ be its power set. An operator F on A is a function $F : \mathcal{P}(A) \to \mathcal{P}(A)$. The operator F is *monotone* if it respects subset inclusion, namely, for all subsets X, Y of A, if $X \subseteq Y$, then $F(X) \subseteq F(Y)$. The operator F is *inflationary* if it maps sets to their supersets, namely, $X \subseteq F(X)$.

Next, we define sequences induced by operators, and characterize the sequences induced by monotone and inflationary operators.

Definition 3.2. Let F be an operator on A. Consider the sequence of sets $F^0, F^1, \ldots$ defined by

$$\begin{aligned} F^0 &= \emptyset, \\ F^{i+1} &= F(F^i). \end{aligned} \tag{3.1}$$

This sequence (F^i) is called *inductive* if it is increasing, namely, if $F^i \subseteq F^{i+1}$ for all i. In this case, we define

$$F^\infty := \bigcup_{i=0}^{\infty} F^i. \tag{3.2}$$

Lemma 3.3. *If F is either monotone or inflationary, the sequence (F^i) is inductive.*

Now we are ready to define fixed points of operators on sets.

Definition 3.4. Let F be an operator on A. The set $X \subseteq A$ is called a *fixed point of F* if $F(X) = X$. A fixed point X of F is called its *least fixed point*, denoted $\mathsf{LFP}(F)$, if it is contained in every other fixed point Y of F, namely, $X \subseteq Y$ whenever Y is a fixed point of F.

Not all operators have fixed points, let alone least fixed points. The Tarski-Knaster guarantees that monotone operators do, and also provides two constructions of the least fixed point for such operators: one "from above" and the other "from below." The latter construction uses the sequences (3.1).

Theorem 3.5 (Tarski-Knaster)**.** *Let F be a monotone operator on a set A.*

1. *F has a least fixed point $\mathsf{LFP}(F)$ which is the intersection of all the fixed points of F. Namely,*
$$\mathsf{LFP}(F) = \bigcap \{Y : Y = F(Y)\}.$$
2. *$\mathsf{LFP}(F)$ is also equal to the union of the stages of the sequence (F^i) defined in (3.1). Namely,*
$$\mathsf{LFP}(F) = \bigcup F^i = F^\infty.$$

However, not all operators are monotone; therefore we need a means of constructing fixed points for non-monotone operators.

Definition 3.6. For an inflationary operator F, the sequence F^i is inductive, and hence eventually stabilizes to the fixed point F^∞. For an arbitrary operator G, we associate the inflationary operator G_{infl} defined by $G_{\text{infl}}(Y) \triangleq Y \cup G(Y)$. The set ${G_{\text{infl}}}^\infty$ is called the *inflationary fixed point* of G, and denoted by $\mathsf{IFP}(G)$.

Definition 3.7. Consider the sequence (F^i) induced by an arbitrary operator F on A. The sequence may or may not stabilize. In the first case, there is a positive integer n such that $F^{n+1} = F^n$, and therefore for all $m > n$, $F^m = F^n$. In the latter case, the sequence F^i does not stabilize, namely, for all $n \leq 2^{|A|}, F^n \neq F^{n+1}$. Now, we define the *partial fixed point* of F, denoted $\mathsf{PFP}(F)$, as F^n in the first case, and the empty set in the second case.

3.2 Fixed Point Logics for P and PSPACE

We now specialize the theory of fixed points of operators to the case where the operators are defined by means of first order formulae.

Definition 3.8. Let σ be a relational vocabulary, and R a relational symbol of arity k that is not in σ. Let $\varphi(R, x_1, \ldots, x_n) = \varphi(R, \mathbf{x})$ be a formula of vocabulary $\sigma \cup \{R\}$. Now consider a structure $\mathfrak{A}$ of vocabulary σ. The formula $\varphi(R, \mathbf{x})$ defines an operator $F_\varphi : \mathcal{P}(A^k) \to \mathcal{P}(A^k)$ on A^k which acts on a subset $X \subseteq A^k$ as

$$F_\varphi(X) = \{\mathbf{a} \mid \mathfrak{A} \models \varphi(X/R,\ \mathbf{a}\}, \tag{3.3}$$

where $\varphi(X/R,\ \mathbf{a}\}$ means that R is interpreted as X in φ.

We wish to extend FO by adding fixed points of operators of the form F_ϕ, where ϕ is a formula in FO. This gives us fixed point logics which play a central role in descriptive complexity theory.

Definition 3.9. Let the notation be as above.

1. The logic $\mathsf{FO}(\mathsf{IFP})$ is obtained by extending FO with the following formation rule: if $\varphi(R, \mathbf{x})$ is a formula and $\mathbf{t}$ a k-tuple of terms, then $[\mathsf{IFP}_{R,\mathbf{x}}\varphi(R, \mathbf{x})](\mathbf{t})$

is a formula whose free variables are those of $\mathbf{t}$. The semantics are given by

$$\mathfrak{A} \models [\mathsf{IFP}_{R,\mathbf{x}}\varphi(R,\mathbf{x})](\mathbf{a}) \text{ iff } \mathbf{a} \in \mathsf{IFP}(F_\varphi).$$

2. The logic $\mathsf{FO}(\mathsf{PFP})$ is obtained by extending FO with the following formation rule: if $\varphi(R,\mathbf{x})$ is a formula and $\mathbf{t}$ a k-tuple of terms, then $[\mathsf{PFP}_{R,\mathbf{x}}\varphi(R,\mathbf{x})](\mathbf{t})$ is a formula whose free variables are those of $\mathbf{t}$. The semantics are given by

$$\mathfrak{A} \models [\mathsf{PFP}_{R,\mathbf{x}}\varphi(R,\mathbf{x})](\mathbf{a}) \text{ iff } \mathbf{a} \in \mathsf{PFP}(F_\varphi).$$

We cannot define the closure of FO under taking least fixed points in the above manner without further restrictions since least fixed points are guaranteed to exist only for monotone operators, and testing for monotonicity is undecidable. If we were to form a logic by extending FO by least fixed points without further restrictions, we would obtain a logic with an undecidable syntax. Hence, we make some restrictions on the formulae which guarantee that the operators obtained from them as described by (3.3) will be monotone, and thus will have a least fixed point. We need a definition.

Definition 3.10. Let notation be as earlier. Let φ be a formula containing a relational symbol R. An occurrence of R is said to be *positive* if it is under the scope of an even number of negations, and *negative* if it is under the scope of an odd number of negations. A formula is said to be *positive in* R if all occurrences of R in it are positive, or there are no occurrences of R at all. In particular, there are no negative occurrences of R in the formula.

Lemma 3.11. *Let notation be as earlier. If the formula $\varphi(R,\mathbf{x})$ is positive in R, then the operator obtained from φ by construction (3.3) is monotone.*

Now we can define the closure of FO under least fixed points of operators obtained from formulae that are positive in a relational variable.

Definition 3.12. The logic $\mathsf{FO}(\mathsf{LFP})$ is obtained by extending FO with the following formation rule: if $\varphi(R, \mathbf{x})$ is a formula that is positive in the k-ary relational variable R, and $\mathbf{t}$ is a k-tuple of terms, then $[\mathsf{LFP}_{R,\mathbf{x}}\varphi(R, \mathbf{x})](\mathbf{t})$ is a formula whose free variables are those of $\mathbf{t}$. The semantics are given by

$$\mathfrak{A} \models [\mathsf{LFP}_{R,\mathbf{x}}\varphi(R, \mathbf{x})](\mathbf{a}) \text{ iff } \mathbf{a} \in \mathsf{LFP}(F_\varphi).$$

As earlier, the stage at which the tuple $\mathbf{a}$ enters the relation R is denoted by $|\mathbf{a}|^{\mathfrak{A}}_{\varphi}$, and inductive depths are denoted by $|\varphi^{\mathfrak{A}}|$. This is well defined for least fixed points since a tuple enters a relation only once, and is never removed from it after. In fixed points (such as partial fixed points) where the underlying formula is not necessarily positive, this is not true. A tuple may enter and leave the relation being built multiple times.

Next, we informally state two well-known results on the expressive power of fixed point logics. First, adding the ability to do simultaneous induction over several formulae does not increase the expressive power of the logic, and secondly $\mathsf{FO}(\mathsf{IFP}) = \mathsf{FO}(\mathsf{LFP})$ over finite structures. See [Lib04, §10.3, p. 184] for details.

We have introduced various fixed point constructions and extensions of first order logic by these constructions. We end this section by relating these logics to various complexity classes. These are the central results of descriptive complexity theory.

Fagin [Fag74] obtained the first machine independent logical characterization of an important complexity class. Here, $\exists\mathsf{SO}$ refers to the restriction of second-order logic to formulae of the form

$$\exists X_1 \cdots \exists X_m \varphi,$$

where φ does not have any second-order quantification.

Theorem 3.13 (Fagin)**.**

$$\exists\mathsf{SO} = \mathbf{NP}.$$

Immerman [Imm82] and Vardi [Var82] obtained the following central result that captures the class $\mathbf{P}$ on ordered structures.

Theorem 3.14 (Immerman-Vardi)**.** *Over finite, ordered structures, the queries expressible in the logic* $\mathsf{FO}(\mathsf{LFP})$ *are precisely those that can be computed in polynomial time. Namely,*

$$\mathsf{FO}(\mathsf{LFP}) = \mathbf{P}.$$

A characterization of **PSPACE** in terms of PFP was obtained in [AV91, Var82].

Theorem 3.15 (Abiteboul-Vianu, Vardi)**.** *Over finite, ordered structures, the queries expressible in the logic* $\mathsf{FO}(\mathsf{PFP})$ *are precisely those that can be computed in polynomial space. Namely,*

$$\mathsf{FO}(\mathsf{PFP}) = \mathbf{PSPACE}.$$

Note: We will often use the term LFP generically instead of $\mathsf{FO}(\mathsf{LFP})$ when we wish to emphasize the fixed point *construction* being performed, rather than the language.

4. The Link Between Polynomial Time Computation and Conditional Independence

In Chapter 2 we saw how certain joint distributions that encode interactions between collections of variables "factor through" smaller, simpler interactions. This necessarily affects the type of influence a variable may exert on other variables in the system. Thus, while a variable in such a system can exert its influence throughout the system, this influence must necessarily be bottlenecked by the simpler interactions that it must factor through. In other words, the influence must propagate with bottlenecks at each stage. In the case where there are conditional independencies, the influence can only be "transmitted through" the values of the intermediate conditioning variables.

In this chapter, we will uncover a similar phenomenon underlying the logical description of polynomial time computation on ordered structures. The fundamental observation is the following:

> *Least fixed point computations "factor through" first order computations, and so limitations of first order logic must be the source of the bottleneck at each stage to the propagation of information in such computations.*

The treatment of LFP versus FO in finite model theory centers around the fact that FO can only express local properties, while LFP allows non-local properties such as transitive closure to be expressed. *We are taking as given the non-local capability of* LFP*, and asking how this non-local nature factors at each step, and what is the effect of such a factorization on the joint distribution of* LFP *acting upon ensembles.*

Fixed point logics allow variables to be non-local in their influence, but this non-local influence must factor through first order logic at each stage. This is a very similar underlying idea to the statistical mechanical picture of random fields over spaces of configurations that we saw in Chapter 2, but comes cloaked in a very different garb — that of logic and operators. The sequence (F^i_φ) of operators that construct fixed points may be seen as the propagation of influence in a structure by means of setting values of "intermediate variables". In this case, the variables are set by inducting them into a relation at various stages of the induction. We want to understand the stage-wise bottleneck that a fixed point computation faces at each step of its execution, and tie this back to notions of conditional independence and factorization of distributions. In order to accomplish this, we must understand the limitations of each stage of a LFP computation and understand how this affects the propagation of long-range influence in relations computed by LFP. Namely, we will bring to bear ideas from statistical mechanics and message passing to the logical description of computations.

It will be beneficial to state this intuition with the example of transitive closure.

EXAMPLE 4.1. The transitive closure of an edge in a graph is the standard example of a non-local property that cannot be expressed by first order logic. It can be expressed in FO(LFP) as follows. Let E be a binary relation that expresses the presence of an edge between its arguments. Then we can see that iterating the positive first order formula $\varphi(R, x, y)$ given by

$$\varphi(R, x, y) \equiv E(x, y) \vee \exists z(E(x, z) \wedge R(z, y)).$$

builds the transitive closure relation in stages.

Notice that the decision of whether a vertex enters the relation is based on the immediate neighborhood of the vertex. In other words, the relation is built stage by stage, and at each stage, vertices that have entered a relation make other vertices that are adjacent to them eligible to enter the relation at the next stage. Thus, *though the resulting property is non-local, the information flow used to*

compute it is stage-wise local. The computation *factors through* a local property at each stage, but by chaining many such local factors together, we obtain the non-local relation of transitive closure. This picture relates to a Markov random field, where such local interactions are chained together in a way that variables can exert their influence to arbitrary lengths, but the factorization of that influence (encoded in the joint distribution) reveals the stage-wise local nature of the interaction. There are important differences however — the flow of LFP computation is directed, whereas a Markov random field is undirected, for instance. We have used this simple example just to provide some preliminary intuition. We will now proceed to build the requisite framework.

4.1 The Limitations of LFP

Many of the techniques in model theory break down when restricted to finite models. A notable exception is the Ehrenfeucht-Fraïssé game for first order logic. This has led to much research attention to game theoretic characterizations of various logics. The primary technique for demonstrating the limitations of fixed point logics in expressing properties is to consider them a segment of the logic $\mathcal{L}^k_{\infty\omega}$, which extends first order logic with infinitary connectives, and then use the characterization of expressibility in this logic in terms of k-pebble games. This is however not useful for our purpose (namely, separating $\mathbf{P}$ from $\mathbf{NP}$) since $\mathbf{NP} \subseteq \mathbf{PSPACE}$ and the latter class is captured by PFP, which is also a segment of $\mathcal{L}^k_{\infty\omega}$.

One of the central contributions of our work is demonstrating a completely different viewpoint of LFP computations in terms of the concepts of conditional independence and factoring of distributions, both of which are fundamental to statistics and probability theory. In order to arrive at this correspondence, we will need to understand the limitations of first order logic. Least fixed point is an iteration of first order formulas. The limitations of first order formulae mentioned in the previous section therefore appear at each step of a least fixed point computation.

Viewing LFP as "stage-wise first order" is central to our analysis. Let us pause for a while and see how this fits into our global framework. We are interested in factoring complex interactions between variables into their smallest constituent irreducible factors. Viewed this way, LFP has a natural factorization into its stages, which are all described by first order formulae.

Let us now analyze the limitations of the LFP computation through this viewpoint.

4.1.1 Locality of First Order Logic

The local properties of first order logic have received considerable research attention and expositions can be found in standard references such as [Lib04, Ch. 4], [EF06, Ch. 2], [Imm99, Ch. 6]. The basic idea is that first order formulae can only "see" up to a certain distance away from their free variables. This distance is determined by the quantifier rank of the formula.

The idea that first order formulae are local has been formalized in essentially two different ways. This has led to two major notions of locality — Hanf locality [Han65] and Gaifman locality [Gai82]. Informally, Hanf locality says that whether or not a first order formula φ holds in a structure depends only on its multiset of isomorphism types of spheres of radius r. Gaifman locality says that whether or not φ holds in a structure depends on the number of elements of that structure having pairwise disjoint r-neighborhoods that fulfill first order formulae of quantifier depth d for some fixed d (which depends on φ). Clearly, both notions express properties of combinations of neighborhoods of fixed size.

In the literature of finite model theory, these properties were developed to deal with cases where the neighborhoods of the elements in the structure had bounded diameters. In particular, some of the most striking applications of such properties are in graphs with bounded degree, such as the linear time algorithm to evaluate first order properties on bounded degree graphs [See96]. In contrast, we will use some of the normal forms developed in the context of locality properties in finite model theory, but in the scenario where neighborhoods of elements have unbounded diameter. Thus, it is not only the locality

that is of interest to us, but the exact specification of the finitary nature of the first order computation. We will see that what we need is that first order logic can only exploit a *bounded* number of *local* properties. We will need both these properties in our analysis.

Recall the notation and definitions from the previous chapter. We need some definitions in order to state the results.

Definition 4.2. The *Gaifman graph* of a σ-structure $\mathfrak{A}$ is denoted by $G_{\mathfrak{A}}$ and defined as follows. The set of nodes of $G_{\mathfrak{A}}$ is A. There is an edge between two nodes a_1 and a_2 in $G_{\mathfrak{A}}$ if there is a relation R in σ and a tuple $\mathbf{t} \in R^{\mathfrak{A}}$ such that both a_1 and a_2 appear in $\mathbf{t}$.

With the graph defined, we have a notion of *distance* between elements a_i, a_j of A, denoted by $d(a_i, a_j)$, as simply the length of the shortest path between a_i and a_j in $G_{\mathfrak{A}}$. We extend this to a notion of distance between tuples from A as follows. Let $\mathbf{a} = (a_1, \ldots, a_n)$ and $\mathbf{b} = (b_1, \ldots, b_m)$. Then

$$d_{\mathfrak{A}}(\mathbf{a}, \mathbf{b}) = \min\{d_{\mathfrak{A}}(a_i, b_j) \colon 1 \leq i \leq n, 1 \leq j \leq m\}.$$

There is no restriction on n and m above. In particular, the definition above also applies to the case where either of them is equal to one. Namely, we have the notion of distance between a tuple and a singleton element. We are now ready to define neighborhoods of tuples. Recall that σ_n is the expansion of σ by n additional constants.

Definition 4.3. Let $\mathfrak{A}$ be a σ-structure and let $\mathbf{a}$ be a tuple over A. The ball of radius r around $\mathbf{a}$ is a set defined by

$$B_r^{\mathfrak{A}}(\mathbf{a}) = \{b \in A \colon d_{\mathfrak{A}}(\mathbf{a}, b) \leq r\}.$$

The r-neighborhood of $\mathbf{a}$ in $\mathfrak{A}$ is the σ_n-structure $N_r^{\mathfrak{A}}(\mathbf{a})$ whose universe is $B_r^{\mathfrak{A}}(\mathbf{a})$; each relation R is interpreted as $R^{\mathfrak{A}}$ restricted to $B_r^{\mathfrak{A}}(\mathbf{a})$; and the n additional constants are interpreted as $a_1, \ldots, a_n$.

We recall the notion of a *type*. Informally, if L is a logic (or language), the L-*type* of a tuple is the sum total of the information that can be expressed about it

in the language L. Thus, the first order type of a m-tuple in a structure is defined as the set of all FO formulae having m free variables that are satisfied by the tuple. Over finite structures, this notion is far too powerful since it characterizes the structure $(\mathfrak{A}, \mathbf{a})$ up to isomorphism. A more useful notion is the *local type* of a tuple. In particular, a neighborhood is a σ_n-structure, and a *type of a neighborhood* is an equivalence class of such structures up to isomorphism. Note that any isomorphism between $N_r^{\mathfrak{A}}(a_1, \ldots, a_n)$ and $N_r^{\mathfrak{B}}(b_1, \ldots, b_n)$ must send a_i to b_i for $1 \leq i \leq n$.

Definition 4.4. Notation as above. The *local r-type* of a tuple $\mathbf{a}$ in $\mathfrak{A}$ is the type of $\mathbf{a}$ in the substructure induced by the r-neighborhood of $\mathbf{a}$ in $\mathfrak{A}$, namely in $N_r(\mathbf{a})$.

In what follows, we may drop the superscript if the underlying structure is clear. The following three notions of locality are used in stating the results.

Definition 4.5.

1. Formulas whose truth at a tuple $\mathbf{a}$ depends only on $B_r(\mathbf{a})$ are called *r-local*. In other words, quantification in such formulas is restricted to the structure $N_r(\mathbf{x})$.

2. Formulas that are r-local around their variables for some value of r are said to be *local*.

3. Boolean combinations of formulas that are local around the various coordinates x_i of $\mathbf{x}$ are said to be *basic local*.

As mentioned earlier, there are two broad flavors of locality results in literature – those that follow from Hanf's theorem, and those that follow from Gaifman's theorem. The first relates two different structures.

Theorem 4.6 ([Han65])**.** *Let $\mathfrak{A}, \mathfrak{B}$ be σ-structures and let $m \in \mathbb{N}$. Suppose that for some $e \in \mathbb{N}$, the 3^m-balls in $\mathfrak{A}$ and $\mathfrak{B}$ have less than e elements, and for each 3^m-neighborhood type τ, either of the following holds.*

1. *Both $\mathfrak{A}$ and $\mathfrak{B}$ have the same number of elements of type τ.*

2. *Both $\mathfrak{A}$ and $\mathfrak{B}$ have more than me elements of type τ.*

Then $\mathfrak{A}$ and $\mathfrak{B}$ satisfy the same first order formulae up to quantifier rank m, written $\mathfrak{A} \equiv_m \mathfrak{B}$.

Note that in clause 1 above, the number of elements may be zero. In other words, the same set of types may be absent in both structures.

The Hanf locality lemma for formulae having a single free variable has a simple form and is an easy consequence of Thm. 4.6.

Lemma 4.7. *Notation as above. Let $\varphi(x)$ be a formula of quantifier depth q. Then there is a radius r and threshold t such that if $\mathfrak{A}$ and $\mathfrak{B}$ have the same multiset of local types up to threshold t, and the elements $a \in \mathfrak{A}$ and $b \in \mathfrak{B}$ have the same local type up to radius r, then*

$$\mathfrak{A} \models \varphi(a) \leftrightarrow \mathfrak{B} \models \varphi(b).$$

See [Lin05] for an application to computing simple monadic fixed points on structures of bounded degree in linear time.

Next we come to Gaifman's version of locality.

Theorem 4.8 ([Gai82])**.** *Every* FO *formula $\varphi(\mathbf{x})$ over a relational vocabulary is equivalent to a Boolean combination of*

1. *local formula around $\mathbf{x}$, and*
2. *sentences of the form*

$$\exists x_1, \ldots, x_s \left(\bigwedge_{i=1}^{s} \phi(x_i) \wedge \bigwedge_{1 \le i \le j \le s} d^{>2r}(x_i, x_j) \right),$$

where the ϕ are r-local.

In words, for every first order formula, there is an r such that the truth of the formula on a structure depends only on the number of elements having disjoint r-neighborhoods that satisfy certain local formulas. This again expresses the *bounded number of local properties* feature that limits first order logic.

The following normal form for first order logic that was developed in an attempt to merge some of the ideas from Hanf and Gaifman locality.

Theorem 4.9 ([SB99])**.** *Every first-order sentence is logically equivalent to one of the form*

$$\exists x_1 \cdots \exists x_l \forall y \varphi(\mathbf{x}, y),$$

where φ is local around y.

4.2 Simple Monadic LFP and Conditional Independence

In this section, we exploit the limitations described in the previous section to build conceptual bridges from least fixed point logic to the Markov-Gibbs picture of the preceding section. At first, this may seem to be an unlikely union. But we will establish that there are fundamental conceptual relationships between the directed Markovian picture and least fixed point computations. The key is to see the constructions underlying least fixed point computations through the lens of influence propagation and conditional independence. In this section, we will demonstrate this relationship for the case of simple monadic least fixed points. Namely, a $\mathsf{FO}(\mathsf{LFP})$ formula without any nesting or simultaneous induction, and where the LFP relation being constructed is monadic. In later sections, we show how to deal with complex fixed points as well.

We wish to build a view of fixed point computation as an information propagation algorithm. In order to do so, let us examine the geometry of information flow during an LFP computation. At stage zero of the fixed point computation, none of the elements of the structure are in the relation being computed. At the first stage, some subset of elements enters the relation. This changes the local neighborhoods of these elements, and the vertices that lie in these local neighborhoods change their local type. Due to the global changes in the multiset of local types, more elements in the structure become eligible for inclusion into the relation at the next stage. This process continues, and the changes "propagate" through the structure. Thus, *the fundamental vehicle of this information propagation is that a fixed point computation $\varphi(R, x)$ changes local neighborhoods of elements at*

each stage of the computation.

This propagation is

1. directed, and

2. relies on a bounded number of local neighborhoods *at each stage*.

In other words, we observe that

> *The influence of an element during* LFP *computation propagates in a similar manner to the influence of a random variable in a directed Markov field.*

This correspondence is important to us. Let us try to uncover the underlying principles that cause it. The directed property comes from the positivity of the first order formula that is being iterated. This ensures that once an element is inserted into the relation that is being computed, it is never removed. Thus, influence flows in the direction of the stages of the LFP computation. Furthermore, this influence flow is local in the following sense: the influence of an element can propagate throughout the structure, but only through its influence on various local neighborhoods.

This correspondence is most striking in the case of bounded degree structures. In that case, we have only $O(1)$ local types.

Lemma 4.10. *On a graph of bounded degree, there is a fixed number of non-isomorphic neighborhoods with radius r. Consequently, there are only a fixed number of local r-types.*

In order to determine whether an element in a structure satisfies a first order formula we need (a) the multiset of local r-types in the structure (also known as its global type) for some value of r, and (b) the local type of the element. Furthermore, by threshold Hanf, we only need to know the multiset of local types up to a certain threshold.

For large enough structures, we will cross the Hanf threshold for the multiset of r-types. At this point, we will be making a decision of whether an element enters the relation based *solely* on its local r-type. This type potentially changes

with each stage of the LFP. At the time when this change renders the element eligible for entering the relation, it will do so. Once it enters the relation, it changes the local r-type of all those elements which lie within a r-neighborhood of it, and such changes render them eligible, and so on. This is how the computation proceeds, in a purely stage-wise local manner. This is a Markov property: the influence of an element upon another must factor entirely through the local neighborhood of the latter.

In the more general case where degrees are not bounded, we still have factoring through local neighborhoods, except that we have to consider all the local neighborhoods in the structure. However, here the bounded nature of FO comes in. The FO formula that is being iterated can only express a property about some bounded number of such local neighborhoods. For example, in the Gaifman form, there are s distinguished disjoint neighborhoods that must satisfy some local condition.

Remark 4.11. The same concept can be expressed in the language of *sufficient statistics*. Namely, knowing some information about certain local neighborhoods renders the rest of the information about variable values that have entered the relation in previous stages of the graph superfluous. In particular, Gaifman's theorem says that for first order properties, *there exists a sufficient statistic that is gathered locally at a bounded number of elements*. Knowing this statistic gives us conditional independence from the values of other elements that have already entered the relation previously, but not from elements that will enter the relation subsequently. This is similar to the *directed* Markov picture where there is conditional independence of any variable from non-descendants given the value of the parents.

At this point, we have exhibited a correspondence between two apparently very different formalisms. This correspondence is illustrated in Fig. 4.1.

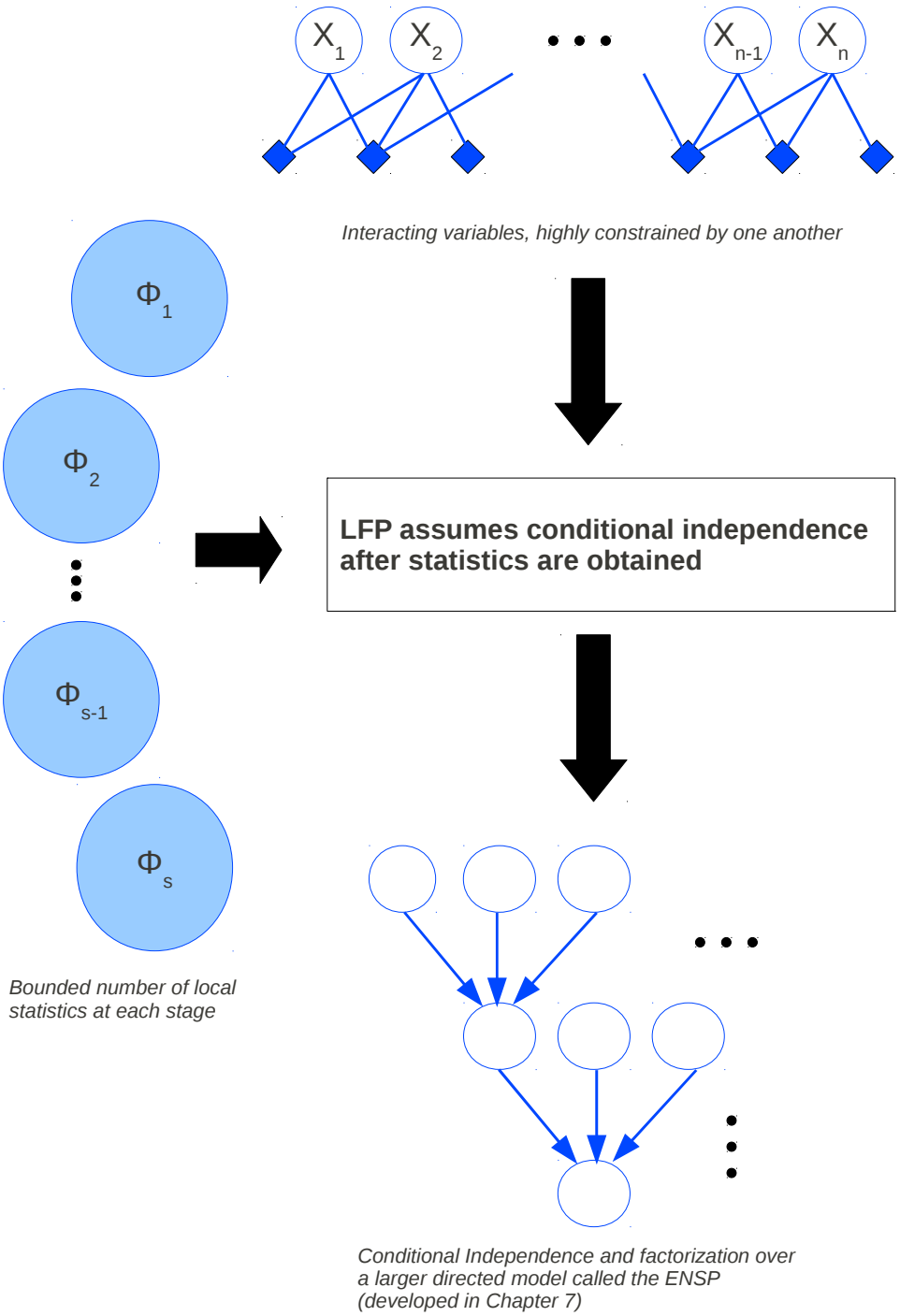

Figure 4.1: The LFP computation process viewed as conditional independencies.

4.3 Conditional Independence in Complex Fixed Points

In the previous sections, we showed that the natural "factorization" of LFP into first order logic, coupled with the bounded local property of first order logic can be used to exhibit conditional independencies in the relation being computed.

But the argument we provided was for simple fixed points having one free variable, namely, for monadic least fixed points. How can we show that this picture is the same for complex fixed points? We accomplish this in stages.

1. First, we use the transitivity theorem for fixed point logic to move nested fixed points into simultaneous fixed points without nesting.

2. Next, we use the simultaneous induction lemma for fixed point logic to encode the relation to be computed as a "section" of a single LFP relation of higher arity.

3. At this point, the picture of the preceding sections applies, except that we have to bookkeep for a k-ary relation that is being computed. The property of "bounded number of local neighborhoods" holds at each stage, except the conditions on the neighborhoods could be expressed in terms of k coordinates instead of just one.

   Alternatively, we could work over a product structure where LFP captures the class of polynomial time computable queries. In other words, we have to work in a structure whose elements are k-tuples of our original structure. In this way, a k-ary LFP over the original structure would be a monadic LFP over this structure.

Steps 1 and 2 involve standard constructions in finite model theory, which we recall in Appendix A. See also [EF06, §8.2]. In order to accomplish step 3, we simply have to ensure that our original structure has a relation that allows an order to be established on k-tuples. In particular this does not pose a problem for encoding instances of k-SAT. The basic nature of information gathering and processing in LFP does not change when the arity of the computation rises. It merely adds the ability to gather polynomially more information at each stage,

but this information is still "bounded number of local neighborhoods at each stage."

Remark 4.12. Note that there are elegant ways to work with the space of equivalence classes of k-tuples with equivalence under first order logic with k-variables. For instance, one can consider a construction known as the *canonical structure* due originally to [DLW95] who used it to provide a model theoretic proof of the important theorem in [AV95] that $\mathbf{P} = \mathbf{PSPACE}$ if and only if $\mathsf{LFP} = \mathsf{PFP}$. Note that this is for all structures, not just for ordered structures.

The issue one faces is that there is a linear order on the canonical structure, which renders the Gaifman graph trivial (totally connected). See [Lib04, §11.5] for more details on canonical structures. The simple scheme described above suffices for our purposes.

4.4 Aggregate Properties of LFP over Ensembles

We have shown that any polynomial time computation will update its relation according to a certain Markov type property on the space of k-types of the underlying structure, after extracting a statistic from the local neighborhoods of the underlying structure. Thus far, there is no probabilistic picture, or a distribution that we can analyze. We are only describing a fully deterministic computation.

The distribution we seek will arise when we examine the aggregate behavior of LFP over *ensembles of structures* that come from ensembles of constraint satisfaction problems (CSPs) such as random k-SAT. When we examine the properties in the aggregate of LFP running over ensembles, we will find that the "bounded number of local" property of each stage of LFP computation manifests as conditional independencies in the distribution. This gives us the setting where we can exploit the full machinery of graphical models of Chapter 2.

Before we examine the distributions arising from LFP acting on ensembles of structures, we will bring in ideas from statistical physics into the proof. We begin this in the next chapter.

5. The 1RSB Ansatz of Statistical Physics

5.1 Ensembles and Phase Transitions

The study of random ensembles of various constraint satisfaction problems (CSPs) is over two decades old, dating back at least to [CF86]. While a given CSP — say, 3-SAT— might be **NP**-complete, many instances of the CSP might be quite easy to solve, even using fairly simple algorithms. Furthermore, such "easy" instances lay in certain well defined regimes of the CSP, while "harder" instances lay in clearly separated regimes. Thus, researchers were motivated to study randomly generated ensembles of CSPs having certain parameters that would specify which regime the instances of the ensemble belonged to. We will see this behavior in some detail for the specific case of the ensemble known as random k-SAT.

An instance of k-SAT is a propositional formula in conjunctive normal form

$$\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$$

having m clauses C_i, each of whom is a disjunction of k literals taken from n variables $\{x_1, \ldots, x_n\}$. The decision problem of whether a satisfying assignment to the variables exists is **NP**-complete for $k \geq 3$. The ensemble known as random k-SAT consists of instances of k-SAT generated randomly as follows. An instance is generated by drawing each of the m clauses $\{C_1, \ldots, C_m\}$ uniformly from the $2^k \binom{n}{k}$ possible clauses having k variables. The entire ensemble of random k-SAT having m clauses over n literals will be denoted by $\mathrm{SAT}_k(n, m)$,

and a single instance of this ensemble will be denoted by $\Phi_k(n, m)$. The *clause density*, denoted by α and defined as $\alpha := m/n$ is the single most important parameter that controls the geometry of the solution space of random k-SAT. Thus, we will mostly be interested in the case where every formula in the ensemble has clause density α. We will denote this ensemble by $\text{SAT}_k(n, \alpha)$, and an individual formula in it by $\Phi_k(n, \alpha)$.

Random CSPs such as k-SAT have attracted the attention of physicists because they model disordered systems such as *spin glasses* where the Ising spin of each particle is a binary variable ("up" or "down") and must satisfy some constraints that are expressed in terms of the spins of other particles. The energy of such a system can then be measured by the number of unsatisfied clauses of a certain k-SAT instance, where the clauses of the formula model the constraints upon the spins. The case of zero energy then corresponds to a solution to the k-SAT instance. The following formulation is due to [MZ97]. First we translate the Boolean variables x_i to Ising variables S_i in the standard way, namely $S_i = -(-1)^{x_i}$. Then we introduce new variables C_{li} as follows. The variable C_{li} is equal to 1 if the clause C_l contains x_i, it is -1 if the clause contains $\neg x_i$, and is zero if neither appears in the clause. In this way, the sum $\sum_{i=1}^{n} C_{li}S_i$ measures the satisfiability of clause C_l. Specifically, if $\sum_{i=1}^{n} C_{li}S_i - k > 0$, the clause is satisfied by the Ising variables. The energy of the system is then measured by the Hamiltonian

$$H = \sum_{i=1}^{m} \delta(\sum_{i=1}^{n} C_{li}S_i, -K).$$

Here $\delta(i, j)$ is the Kronecker delta. Thus, satisfaction of the k-SAT instance translates to vanishing of this Hamiltonian. Statistical mechanics then offers techniques such as replica symmetry, to analyze the macroscopic properties of this ensemble.

Also very interesting from the physicist's point of view is the presence of a sharp *phase transition* [CKT91, MSL92] (see also [KS94]) between satisfiable and unsatisfiable regimes of random k-SAT. Namely, empirical evidence suggested that the properties of this ensemble undergoes a clearly defined transition when the clause density is varied. This transition is conjectured to be as follows. For

each value of k, there exists a transition threshold $\alpha_c(k)$ such that with probability approaching 1 as $n \to \infty$ (called the *Thermodynamic limit* by physicists),

- if $\alpha < \alpha_c(k)$, an instance of random k-SAT is satisfiable. Hence this region is called the *SAT phase*.
- If $\alpha > \alpha_c(k)$, an instance of random k-SAT is unsatisfiable. This region is known as the *unSAT phase*.

There has been intense research attention on determining the numerical value of the threshold between the SAT and unSAT phases as a function of k. [Fri99] provides a sharp but non-uniform construction (namely, the value α_c is a function of the problem size, and is conjectured to converge as $n \to \infty$). Functional upper bounds have been obtained using the first moment method [MA02] and improved using the second moment method [AP04] that improves as k gets larger.

5.2 The d1RSB Phase

More recently, another thread on this crossroad has originated once again from statistical physics and is most germane to our perspective. This is the work in the progression [MZ97], [BMW00], [MZ02], and [MPZ02] that studies the evolution of the solution space of random k-SAT as the constraint density increases towards the transition threshold. In these papers, physicists have conjectured that there is a *second* threshold that divides the SAT phase into two — an "easy" SAT phase, and a "hard" SAT phase. In both phases, there is a solution with high probability, but while in the easy phase one giant connected cluster of solutions contains almost all the solutions, in the hard phase this giant cluster shatters into exponentially many communities that are far apart from each other in terms of least Hamming distance between solutions that lie in distinct communities. Furthermore, these communities shrink and recede maximally far apart as the constraint density is increased towards the SAT-unSAT threshold. As this threshold is crossed, they vanish altogether.

As the clause density is increased, a picture known as the "1RSB hypothesis" emerges that is illustrated in Fig. 5.1, and described below.

RS For $\alpha < \alpha_d$, a problem has many solutions, but they all form one giant cluster within which going from one solution to another involves flipping only a finite (bounded) set of variables. This is the *replica symmetric* phase.

d1RSB At some value of $\alpha = \alpha_d$ which is below α_c, it has been observed that the space of solutions splits up into "communities" of solutions such that solutions within a community are close to one another, but are far away from the solutions in any other community. This effect is known as *shattering* [ACO08]. Within a community, flipping a bounded finite number of variable assignments on one satisfying takes one to another satisfying assignment. But to go from one satisfying assignment in one community to a satisfying assignment in another, one has to flip a fraction of the set of variables and therefore encounters what physicists would consider an "energy barrier" between states. This is the *dynamical one step replica symmetry breaking* phase.

unSAT Above the SAT-unSAT threshold, the formulas of random k-SAT are unsatisfiable with high probability.

Using statistical physics methods, [KMRT+07] obtained another phase that lies between d1RSB and unSAT. In this phase, known as 1RSB (*one step replica symmetry breaking*), there is a "condensation" of the solution space into a sub-exponential number of clusters, and the sizes of these clusters go to zero as the transition occurs, after which there are no more solutions. This phase has not been proven rigorously thus far to our knowledge and we will not revisit it in this work.

The 1RSB hypothesis has been proven rigorously for high values of k. Specifically, the existence of the d1RSB phase has been proven rigorously for the case of $k > 8$, starting with [MMZ05] (see also [DMMZ08]) who showed the existence of clusters in a certain region of the SAT phase using first moment methods. Later, [ART06] rigorously proved that there exist exponentially many clus-

ters in the d1RSB phase and showed that within any cluster, the fraction of variables that take the same value in the entire cluster (the so-called *frozen variables*) goes to one as the SAT-unSAT threshold is approached. Further [ACO08] obtained analytical expressions for the threshold at which the solution space of random k-SAT (as also two other CSPs — random graph coloring and random hypergraph 2-colorability) shatters, as well as confirmed the $O(n)$ Hamming separation between clusters.

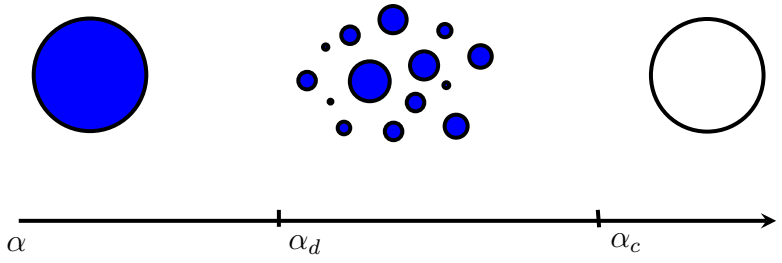

Figure 5.1: The clustering of solutions just before the SAT-unSAT threshold. Below α_d, the space of solution is largely connected. Between α_d and α_c, the solutions break up into exponentially many communities. Above α_c, there are no more solutions, which is indicated by the unfilled circle.

In summary, in the region of constraint density $\alpha \in [\alpha_d, \alpha_c]$, the solution space is comprised of exponentially many communities of solutions which require a fraction of the variable assignments to be flipped in order to move between each other.

5.2.1 Cores and Frozen Variables

In this section, we reproduce results about the distribution of variable assignments within each cluster of the d1RSB phase from [MMW07], [ART06], and [ACO08].

We first need the notion of the *core* of a cluster. Given any solution in a cluster, one may obtain the core of the cluster by "peeling away" variable assignments that, loosely speaking occur only in clauses that are satisfied by other

variable assignments. This process leads to the core of the cluster.

To get a formal definition, first we define a *partial assignment* of a set of variables $(x_1, \ldots, x_n)$ as an assignment of each variable to a value in $\{0, 1, *\}$. The $*$ assignment is akin to a "joker state" which can take whichever value is most useful in order to satisfy the k-SAT formula.

Next, we say that a variable in a partial assignment is *free* when each clause it occurs in has at least one other variable that satisfies the clause, or has as assignment to $*$.

Finally, to obtain the core of a cluster, we repeat the following starting with any solution in the cluster: if a variable is free, assign it a $*$.

This process will eventually lead to a fixed point, and that is the *core* of the cluster. We may easily see that the core is not dependent upon the choice of the initial solution.

What does the core of a cluster look like? Recall that the core is itself a partial assignment, with each variable being assigned a $0, 1$ or a $*$. Of obvious interest are those variables that are assigned 0 or 1. These variables are said to be *frozen*. Note that since the core can be arrived at starting from any choice of an initial solution in the cluster, it follows that frozen variables take the same value throughout the cluster. For example, if the variable x_i takes value 1 in the core of a cluster, then every solution lying in the cluster has x_i assigned the value 1. The *non-frozen* variables are those that are assigned the value $*$ in the core. These take both values 0 and 1 in the cluster. Clearly the number of $*$ variables is a measure of the internal entropy (and therefore the size) of a cluster since it is only these variables whose values vary within the cluster.

Apriori, we have no way to tell that the core will not be the all $*$ partial assignment. Namely, we do not know whether there are any frozen variables at all. However, [ART06] proved that for high enough values of k, with probability going to 1 in the thermodynamic limit, almost every variable in a core is frozen as we increase the clause density towards the SAT-unSAT threshold.

Theorem 5.1 ([ART06]). *For every $r \in [0, \frac{1}{2}]$ there is a constant k_r such that for all $k \geq k_r$, there exists a clause density $\alpha(k, r) < \alpha_c$ such that for all $\alpha \in [\alpha(k, r), \alpha_c]$,*

asymptotically almost surely

1. *every cluster of solutions of* $\Phi_k(n, \alpha n)$ *has at least* $(1 - r)n$ *frozen variables,*

2. *fewer than* rn *variables take the value* $*$.

We end this section with a physical picture of what forms a core. If a formula Φ has a core with C clauses, then these clauses must have literals that come from a set of at most C variables. By bounding the probability of this event, [MMW07] obtained a lower bound on the size of cores. The bound is linear, which means that when cores do exist ([ART06] proved their existence for sufficiently high k), they must involve a fraction of all the variables in the formula. In other words, a core may be thought of as the onset of a large single interaction of degree $O(n)$ among the variables. As the reader may imagine after reading the previous chapters, this sort of interaction cannot be dealt with by LFP algorithms. We will need more work to make this precise, but informally cores are too large to pass through the bottlenecks that the stage-wise first order LFP algorithms create.

This may also be interpreted as follows. Algorithms based on LFP can tackle long range interactions between variables, but only when they can be factored into interactions of degree $\mathrm{poly}(\log n)$. The exact degree is determined by the LFP algorithm — those that take more time to complete can deal with higher degrees, but it is always $\mathrm{poly}(\log n)$. But the appearance of cores is equivalent to the onset of $O(n)$ degree interactions which cannot be further factored into $\mathrm{poly}(\log n)$ degree interactions. Such large interactions, caused by increasing the clause density sufficiently, cannot be dealt with using an LFP algorithm.

We have already noted that this is because LFP algorithms factor through first order computations, and in a first order computation, the decision of whether an element is to enter the relation being computed is based on information collected from local neighborhoods and combined in a bounded fashion. This bottleneck is too small for a core to factor through. The precise statement of this intuitive picture will be provided in the next chapter when we build our conditional independence hierarchies.

5.2.2 Performance of Known Algorithms

We end this chapter with a brief overview of the performance of known algorithms as a function of the clause density, and pointers to more detailed surveys.

Beginning with [CKT91] and [MSL92], there has been an understanding that hard instances of random k-SAT tend to occur when the constraint density α is near the *transition threshold*, and that this behavior was similar to phase transitions in spin glasses [KS94]. Now that we have surveyed the known results about the geometry of the space of solutions in this region, we turn to the question of how the two are related.

It has been empirically observed that the onset of the d1RSB transition seems to coincide with the constraint density where traditional solvers tend to exhibit exponential slowdown; see [ACO08] and [CO09]. See also [CO09] for the best current algorithm along with a comparison of various other algorithms to it. Thus, while both regimes in SAT have solutions with high probability, the ease of finding a solution differs quite dramatically on traditional SAT solvers due to a clustering of the solution space into numerous communities that are far apart from each other in terms of Hamming distance. In particular, for clause densities above $O(2^k/k)$, no algorithms are known to produce solutions in polynomial time with probability $\Omega(1)$. Compare this to the SAT-unSAT threshold, which is asymptotically $2^k \ln 2$. Thus, well below the SAT-unSAT threshold, in regimes where we know solutions exist, we are currently unable to find them in polynomial time. Our work will explain that indeed, this is fundamentally a limitation of polynomial time algorithms.

Incomplete algorithms are a class that do not always find a solution when it exists, nor do they indicate the lack of solution except to the extent that they were unable to find one. Incomplete algorithms are obviously very important for hard regimes of constraint satisfaction problems since we do not have complete algorithms in these regimes that have economical running times. More recently, a breakthrough for incomplete algorithms in this field came with [MPZ02] who used the cavity method from spin glass theory to construct an algorithm named *survey propagation* that does very well on instances of random

k-SAT with constraint density *above* the aforementioned clustering threshold, and continues to perform well very close to the threshold α_c for low values of k. Survey propagation seems to scale as $n \log n$ in this region. The algorithm uses the 1RSB hypothesis about the clustering of the solution space into numerous communities. The behavior of survey propagation for higher values of k is still being researched.

6. Random Graph Ensembles

We will use factor graphs as a convenient means to encode various properties of the random k-SAT ensemble. In this section we introduce the factor graph ensembles that represent random k-SAT. Our treatment of this section follows [MM09, Chapter 9].

Definition 6.1. The *random k-factor graph ensemble*, denoted by $\mathbb{G}_k(n, m)$, consists of graphs having n variable nodes and m function nodes constructed as follows. A graph in the ensemble is constructed by picking, for each of the m function nodes in the graph, a k-tuple of variables uniformly from the $\binom{n}{k}$ possibilities for such a k-tuple chosen from n variables.

Graphs constructed in this manner may have two function nodes connected to the same k-tuple of variables. In this ensemble, function nodes all have degree k, while the degree of the variable nodes is a random variable with expectation km/n.

Definition 6.2. The *random (k, α)-factor graph ensemble*, denoted by $\mathbb{G}_k(n, \alpha)$, consists of graphs constructed as follows. For each of the $\binom{n}{k}$ k-tuples of variables, a function node that connects to only these k variables is added to the graph with probability $\alpha n / \binom{n}{k}$.

In this ensemble, the number of function nodes is a random variable with expectation αn, and the degree of variable nodes is a random variable with expectation αk.

We will be interested in the case of the thermodynamic limit of $n, m \to \infty$ with the ratio $\alpha := m/n$ being held constant. In this case, both the ensembles converge in the properties that are important to us, and both can be seen as the

underlying factor graph ensembles to our random k-SAT ensemble $\text{SAT}_k(n, \alpha)$ (see Chapter 5 for definitions and our notation for random k-SAT ensembles).

With the definitions in place, we are ready to describe two properties of random graph ensembles that are pertinent to our problem.

6.1 Properties of Factor Graph Ensembles

The first property provides us with intuition on why algorithms find it so hard to put together local information to form a global perspective in CSPs.

6.1.1 Locally Tree-Like Property

We have seen in Chapter 4 that the propagation of influence of variables during a LFP computation is *stagewise-local*. This is really the fundamental limitation of LFP that we seek to exploit. In order to understand why this is a limitation, we need to examine what local neighborhoods of the factor graphs underlying **NP**-complete problems like k-SAT look like in hard phases such as d1RSB. In such phases, there are many extensive (meaning $O(n)$) correlations between variables that arise due to loops of sizes $O(\log n)$ and above.

However, remarkably, such graphs are *locally trivial*. By that we mean that there are no cycles in a $O(1)$ sized neighborhood of any vertex as the size of the graph goes to infinity [MM09, §9.5]. One may demonstrate this for the Erdos-Renyi random graph as follows. Here, there are n vertices, and there is an edge between any two with probability c/n where c is a constant that parametrizes the density of the graph. Edges are "drawn" uniformly and independently of each other. Consider the probability of a certain graph (V, E) occurring as a subgraph of the Erdos-Renyi graph. Such a graph can occur in $\binom{n}{|V|}$ positions. At each position, the probability of the graph structure occurring is

$$p^{|E|}(1-p)^{\frac{|V|}{2}-|E|}.$$

Applying Stirling's approximations, we see that such a graph occurs asymptotically $O(|V| - |E|)$ times. If the graph is connected, $|V| \leq |E| - 1$ with equality

only for trees. Thus, in the limit of $n \to \infty$, finite connected graphs have vanishing probability of occurring in finite neighborhoods of any element.

In short, if only local neighborhoods are examined, the two ensembles $\mathbb{G}_k(n, m)$ and $\mathbb{T}_k(n, m)$ are indistinguishable from each other.

Theorem 6.3. *Let G be a randomly chosen graph in the ensemble $\mathbb{G}_k(n, m)$, and i be a uniformly chosen node in G. Then the r-neighborhood of i in G converges in distribution to $\mathbb{T}_k(n, m)$ as $n \to \infty$.*

Let us see what this means in terms of the information such graphs divulge locally. The simplest local property is degrees of elements. These are, of course, available through local inspection. The next would be small connected subgraphs (triangles, for instance). But even this next step is not available. In other words, such random graphs do not provide any of their global properties through local inspection at each element.

Let us think about what this implies. We know from the onset of cores and frozen variables in the 1dRSB phase of k-SAT that there are strong correlations between blocks of variables of size $O(n)$ in that phase. However, these loops are *invisible* when we inspect local neighborhoods of a fixed finite size, as the problem size grows.

6.1.2 Degree Profiles in Random Graphs

The degree of a variable node in the ensemble $\mathbb{G}_k(n, m)$ is a random variable. We wish to understand the distribution of this random variable. The expected value of the fraction of variables in $\mathbb{G}_k(n, m)$ having degree d is the same as the expected value that a single variable node has degree d, both being equal to

$$P(\deg v_i = d) = \binom{m}{d} p^k (1-p)^{m-d} \qquad \text{where } p = \frac{k}{d}.$$

In the large graph limit we get

$$\lim_{n\to\infty} P(\deg v_i = d) = e^{-k\alpha} \frac{(k\alpha)^d}{n\,!}.$$

In other words, the degree is asymptotically a Poisson random variable.

A corollary is that the maximum degree of a variable node is almost surely less than $O(\log n)$ in the large graph case.

Lemma 6.4. *The maximum variable node degree in $\mathbb{G}_k(n, m)$ is asymptotically almost surely $O(\log n)$. In particular, it asymptotically almost surely satisfies the following*

$$\frac{d_{\max}}{k\alpha e} = \frac{z}{\log(z/\log z)}\left[1 + \Theta\left(\frac{\log\log z}{(\log z)^2}\right)\right]. \tag{6.1}$$

where $z = (\log n)/k\alpha e$.

Proof. See [MM09, p. 184] for a discussion of this upper bound, as well as a lower bound. ■

7. Separation of Complexity Classes

We have built a framework that connects ideas from graphical models, logic, statistical mechanics, and random graphs. We are now ready to begin our final constructions that will yield the separation of complexity classes.

7.1 Measuring Conditional Independence

The central concern of this work has been to understand what are the irreducible interactions between the variables in a system — namely, those that cannot be expressed in terms of interactions between smaller sets of variables with conditional independencies between them. Such irreducible interactions can be 2-interactions (between pairs), 3-interactions (between triples), and so on, up to n-interactions between all n variables simultaneously.

A joint distribution encodes the interaction of a system of n variables. What would happen if all the *direct* interactions between variables in the system were all of less than a certain finite range k, with $k < n$? In such a case, the "jointness" of the covariates really would lie at a lower "level" than n. We would like to measure the "level" of conditional independence in a system of interacting variables by inspecting their joint distribution. At level zero of this "hierarchy", the covariates should be independent of each other. At level n, they are coupled together n at a time, without the possibility of being decoupled. In this way, we can make statements about how deeply entrenched the conditional independence between the covariates is, or dually, about how large the set of direct interactions between variables is.

This picture is captured by the number of independent parameters required

to parametrize the distribution. When the largest irreducible interactions are k-interactions, the distribution can be parametrized with $n2^k$ independent parameters. Thus, in families of distributions where the irreducible interactions are of fixed size, the independent parameter space grows polynomially with n, whereas in a general distribution without any conditional independencies, it grows exponentially with n. The case of LFP lies in between — the interactions are not of fixed size, but they grow relatively slowly.

There are some technical issues with constructing such a hierarchy to measure conditional independence. The first issue would be how to measure the level of a distribution in this hierarchy. If, for instance, the distribution has a directed $\mathcal{P}$-map, then we could measure the size of the largest clique that appears in its moralized graph. However, as noted in Sec. 2.5, not all distributions have such maps. We may, of course, upper and lower bound the level using minimal $\mathcal{I}$-maps and maximal $\mathcal{D}$-maps for the distribution. In the case of ordered graphs, we should note that there may be different minimal $\mathcal{I}$-maps for the same distribution for different orderings of the variables. See [KF09, p. 80] for an example.

The insight that allows us to resolve the issue is as follows. If we could somehow *embed* the distribution of solutions generated by LFP into a larger distribution, such that

1. the larger distribution factorized recursively according to some directed graphical model, and

2. the larger distribution had only polynomially many more variates than the original one,

then we would have obtained a parametrization of our distribution that would reflect the factorization of the larger distribution, and would cost us only polynomially more, which does not affect us.

By pursuing the above course, we aim to demonstrate that distributions of solutions generated by LFP lie at a lower level of conditional independence than distributions that occur in the d1RSB phase of random k-SAT. Consequently,

they have more economical parametrizations than the space of solutions in the 1dRSB phase does.

We will return to the task of constructing such an embedding in Sec. 7.3. First we describe how we use LFP to create a distribution of solutions.

7.2 Generating Distributions from LFP

7.2.1 Encoding k-SAT into Structures

In order to use the framework from Chapters 3 and 4, we will encode k-SAT formulae as structures over a fixed vocabulary.

Our vocabularies are relational, and so we need only specify the set of relations, and the set of constants. We will use three relations.

1. The first relation R_C will encode the clauses that a SAT formula comprises. Since we are studying ensembles of random k-SAT, this relation will have arity k.

2. We need a relation in order to make FO(LFP) capture polynomial time queries on the class of k-SAT structures. We will not introduce a linear ordering since that would make the Gaifman graph a clique. Rather we will include a relation such that FO(LFP) can capture all the polynomial time queries on the structure. This will be a binary relation R_E.

3. Lastly, we need a relation R_P to hold "partial assignments" to the SAT formulae. We will describe these in the Sec. 7.2.3.

4. We do not require constants.

This describes our vocabulary

$$\sigma = \{R_C, R_E, R_P\}.$$

Next, we come to the universe. A SAT formula is defined over n variables, but they can come either in positive or negative form. Thus, our universe will

have $2n$ elements corresponding to the variables $x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n$. In order to avoid new notation, we will simply use the same notation to indicate the corresponding element in the universe. We denote by lower case x_i the literals of the formula, while the corresponding upper case X_i denotes the corresponding variable in a model.

Finally, we need to interpret our relations in our universe. We dispense with the superscripts since the underlying structure is clear. The relation R_C will consist of k-tuples from the universe interpreted as clauses consisting of disjunctions between the variables in the tuple. The relation R_E will be interpreted as an "edge" between successive variables. The relation R_P will be a partial assignment of values to the underlying variables.

Now we encode our k-SAT formulae into σ-structures in the natural way. For example, for $k = 3$, the clause $x_1 \vee \neg x_2 \vee \neg x_3$ in the SAT formula will be encoded by inserting the tuple $(x_1, \neg x_2, \neg x_3)$ in the relation R_C. Similarly, the pairs (x_i, x_{i+1}) and $(\neg x_i, \neg x_{i+1})$, both for $1 \leq i < n$, as well as the pair $(x_n, \neg x_1)$ will be in the relation R_E. This chains together the elements of the structure.

The reason for the relation R_E that creates the chain is that on such structures, polynomial time queries are captured by $\mathsf{FO(LFP)}$ [EF06, §11.2]. This is a technicality. Recall that an order on the structure enables the LFP computation (or the Turing machine the runs this computation) to represent tuples in a lexicographical ordering. In our problem k-SAT, it plays no further role. Specifically, the assignments to the variables that are computed by the LFP have nothing to do with their order. They depend only on the relation R_C which encodes the clauses and the relation R_P that holds the initial partial assignment that we are going to ask the LFP to extend. In other words, each stage of the LFP is *order-invariant*. It is known that the class of order invariant queries is also Gaifman local [GS00]. However to allow LFP to capture polynomial time on the class of encodings, we need to give the LFP something it can use to create an ordering. We could encode our structures with a linear order, but that would make the Gaifman graph fully connected. What we want is something weaker, that still suffices. Thus, we encode our structures as successor-type structures through

the relation R_E. This seems most natural, since it imparts on the structure an ordering based on that of the variables. Note also that SAT problems may also be represented as matrices (rows for clauses, columns for variables that appear in them), which have a well defined notion of order on them.

Ensembles of k-SAT Let us now create ensembles of σ-structures using the encoding described above. We will start with the ensemble $\mathrm{SAT}_k(n, \alpha)$ and encode each k-SAT instance as a σ-structure. The resulting ensemble will be denoted by $\mathfrak{S}_k(n, \alpha)$. The encoding of the problem $\Phi_k(n, \alpha)$ as a σ-structure will be denoted by $\mathfrak{P}_k(n, \alpha)$.

7.2.2 The LFP Neighborhood System

In this section, we wish to describe the neighborhood system that underlies the monadic LFP computations on structures of $\mathfrak{S}_k(n, \alpha)$. We begin with the factor graph, and build the neighborhood system through the Gaifman graph.

Let us recall the factor graph ensemble $\mathbb{G}_k(n, m)$. Each graph in this ensemble encodes an instance of random k-SAT. We encode the k-SAT instance as a structure as described in the previous section. Next, we build the Gaifman graph of each such structure. The set of vertices of the Gaifman graph are simply the set of variable nodes in the factor graph and their negations since we are using both variables and their negations for convenience (this is simply an implementation detail). For instance, the Gaifman graph for the factor graph of Fig 2.2 will have 12 vertices. Two vertices are joined by an edge in the Gaifman graph either when the two corresponding variable nodes were joined to a single function node (i.e., appeared in a single clause) of the factor graph or if they are adjacent to each other in the chain that relation R_E has created on the structure.

On this Gaifman graph, the simple monadic LFP computation induces a neighborhood system described as follows. The sites of the neighborhood system are the variable nodes. The neighborhood $\mathcal{N}_s$ of a site s is the set of all nodes that lie in the r-neighborhood of a site, where r is the locality rank of the first order formula φ whose fixed point is being constructed by the LFP computation.

Finally, we make the neighborhood system into a graph in the standard way. Namely, the vertices of the graph will be the set of sites. Each site s will be connected by an edge to every other site in $\mathcal{N}_s$. This graph will be called the *interaction graph* of the LFP computation. The ensemble of such graphs, parametrized by the clause density α, will be denoted by $\mathbb{I}_k(n, \alpha)$.

Note that this interaction graph has many more edges in general than the Gaifman graph. In particular, every node that was within the locality rank neighborhood of the Gaifman graph is now connected to it by a single edge. The resulting graph is, therefore, far more dense than the Gaifman graph.

What is the size of cliques in this interaction graph? This is not the same as the size of cliques in the factor graph, or the Gaifman graph, because the density of the graph is higher. The size of the largest clique is a random variable. What we want is an asymptotic almost sure (by this we mean with probability tending to 1 in the thermodynamic limit) upper bound on the size of the cliques in the distribution of the ensemble $\mathbb{I}_k(n, \alpha)$.

Note: From here on, all the statements we make about ensembles should be understood to hold asymptotically almost surely in the respective random ensembles. By that we mean that they hold with probability 1 as $n \to \infty$.

Lemma 7.1. *The size of cliques that appear in graphs of the ensemble $\mathbb{I}_k(n, \alpha)$ are upper bounded by $\mathrm{poly}(\log n)$ asymptotically almost surely.*

Proof. Let $d_{\max}$ be as in (6.1), and r be the locality rank of φ. The maximum degree of a node in the Gaifman graph is asymptotically almost surely upper bounded by $d_{\max} = O(\log n)$. The locality rank is a fixed number (roughly equal to 3^d where d is the quantifier depth of the first order formula that is being iterated). The node under consideration could have at most $d_{\max}$ others adjacent to it, and the same for those, and so on. This gives us a coarse $d_{\max}^r$ upper bound on the size of cliques. ■

Remark 7.2. While this bound is coarse, there is not much point trying to tighten it because any constant power factor (r in the case above) can always be introduced by computing a r-ary LFP relation. This bound will be sufficient for

us.

Remark 7.3. High degree nodes in the Gaifman graph become significant features in the interaction graph since they connect a large number of other nodes to each other, and therefore allow the LFP computation to access a lot of information through a neighborhood system of given radius. It is these high degree nodes that reduce factorization of the joint distribution since they represent direct interaction of a large number of variables with each other. Note that although the radii of neighborhoods are $O(1)$, the number of nodes in them is not $O(1)$ due to the Poisson distribution of the variable node degrees, and the existence of high degree nodes.

Remark 7.4. The relation being constructed is monadic, and so it does not introduce new edges into the Gaifman graph at each stage of the LFP computation. When we compute a k-ary LFP, we can encode it into a monadic LFP over a polynomially (n^k) larger product space, as is done in the canonical structure, for instance, but with the linear order replaced by a weaker successor type relation. Therefore, we can always chose to deal with monadic LFP. This is really a restatement of the transitivity principle for inductive definitions that says that if one can write an inductive definition in terms of other inductively defined relations over a structure, then one can write it directly in terms of the original relations that existed in the structure [Mos74, p. 16].

7.2.3 Generating Distributions

The standard scenario in finite model theory is to ask a query about a structure and obtain a Yes/No answer. For example, given a graph structure, we may ask the query "Is the graph connected?" and get an answer.

But what we want are *distributions* of solutions that are computed by a purported LFP algorithm for k-SAT. This is not generally the case in finite model theory. Intuitively, we want to generate solutions lying in exponentially many clusters of the solution space of SAT in the d1RSB phase. How do we do this? To generate these distributions, we will start with *partial assignments* to the set

of variables in the formula, and ask the question whether such a partial assignment can be extended to a satisfying assignment. Since the answer to such a question can be verified in polynomial time, such a query must be expressible in FO(LFP) itself on our encoding of k-SAT into structures if $\mathbf{P} = \mathbf{NP}$. In fact, through the self-reducibility of SAT, we can see that the resulting assignment will itself be expressible as a LFP computable global relation.

Since we want to generate exponentially many such solutions, we will have to partially assign $O(n)$ (a small fraction) of the variables, and ask the LFP to extend this assignment, whenever possible, to a satisfying assignment to all variables. Thus, we now see what the relation R_P in our vocabulary stands for. It holds the partial assignment to the variables. For example, suppose we want to ask whether the partial assignment $x_1 = 1, x_2 = 0, x_3 = 1$ can be extended to a satisfying assignment to the SAT formula, we would store this partial assignment in the tuple $(x_1, \neg x_2, x_3)$ in the relation R_P in our structure.

The output satisfying assignment will be computed as a unary relation which holds all the literals that are assigned the value 1. This means that x_i is in the relation if x_i has been assigned the value 1 by the LFP, and otherwise $\neg x_i$ is in the relation meaning that x_i has been assigned the value 0 by the LFP computation. This is the simplest case where the FO(LFP) formula is simple monadic. For more complex formulas, the output will be some section of a relation of higher arity (please see Appendix A for details), and we will view it as monadic over a polynomially larger structure.

Now we "initialize" our structure with different partial assignments and ask the LFP to compute complete assignments when they exist. If the partial assignment cannot be extended, we simply abort that particular attempt and carry on with other partial assignments until we generate enough solutions. By "enough" we mean rising exponentially with the underlying problem size. In this way we get a distribution of solutions that is exponentially numerous, and we now analyze it and compare it to the one that arises in the d1RSB phase of random k-SAT.

7.3 Disentangling the Interactions: The ENSP Model

Now that we have a distribution of solutions computed by LFP, we would like to examine its conditional independence characteristics. Does it factor through any particular graphical model, for instance?

In Chapter 2, we considered various graphical models and their conditional independence characteristics. Once again, our situation is not exactly like any of these models. We will have to build our own, based on the principles we have learnt. Let us first note two issues.

The first issue is that graphical models considered in literature are mostly *static*. By this we mean that

1. they are of fixed size, over a fixed set of variables, and
2. the relations between the variables encoded in the models are fixed.

In short, they model fixed interactions between a fixed set of variables. Since we wish to apply them to the setting of complexity theory, we are interested in *families* of such models, with a focus on how their structure changes with the problem size.

The second issue that faces us now is as follows. Even within a certain size n, we do *not* have a fixed graph on n vertices that will model all our interactions. The way a LFP computation proceeds through the structure will, in general, vary with the initial partial assignment. We would expect a different "trajectory" of the LFP computation for different clusters in the 1dRSB phase. So, if one initial partial assignment landed us in cluster X, and another in cluster Y, the way the LFP would go about assigning values to the unassigned variables would be, in general, quite different. Even within a cluster, the trajectories of two different initial partial assignments will not be the same, although we would expect them to be similar. How do we deal with this situation?

In order to model this dynamic behavior, let us build some intuition first.

1. We know that there is a "directed-ness" to LFP in that elements that are assigned values at a certain stage of the computation then go on to influ-

ence other elements who are as yet unassigned. Thus, there is a directed flow of influence as the LFP computation progresses. This is, for example, different from a Markov random field distribution which has no such direction.

2. There are two types of flows of information in a LFP computation. Consider simple monadic LFP. In one type of flow, neighborhoods across the structure influence the value an unassigned node will take. In the other type of flow, once an element is assigned a value, it changes the neighborhoods (or more precisely the local types of various other elements) in its vicinity. Note that while the first type of flow happens during a stage of the LFP, the second type is implicit. Namely, there is no separate stage of the LFP where it happens. It implicitly happens once any element enters the relation being computed.

3. Because the flow of information is as described above, we will not be able to express it using a simple DAG on either the set of vertices, or the set of neighborhoods. Thus, we have to consider building a graphical model on certain larger *product* spaces.

4. The stage-wise nature of LFP is central to our analysis, and the various stages cannot be bundled into one without losing crucial information. Thus, we do need a model which captures each stage separately.

5. In order to exploit the factorization properties of *directed* graphical models, and the resulting parametrization by potentials, we would like to avoid any closed directed paths.

Let us now incorporate this intuition into a model, which we will call a *Element-Neighborhood-Stage Product Model*, or ENSP model for short. This model appears to be of independent interest. We now describe the ENSP model for a simple monadic least fixed point computation. The model is illustrated in Fig. 7.1. It has two types of vertices.

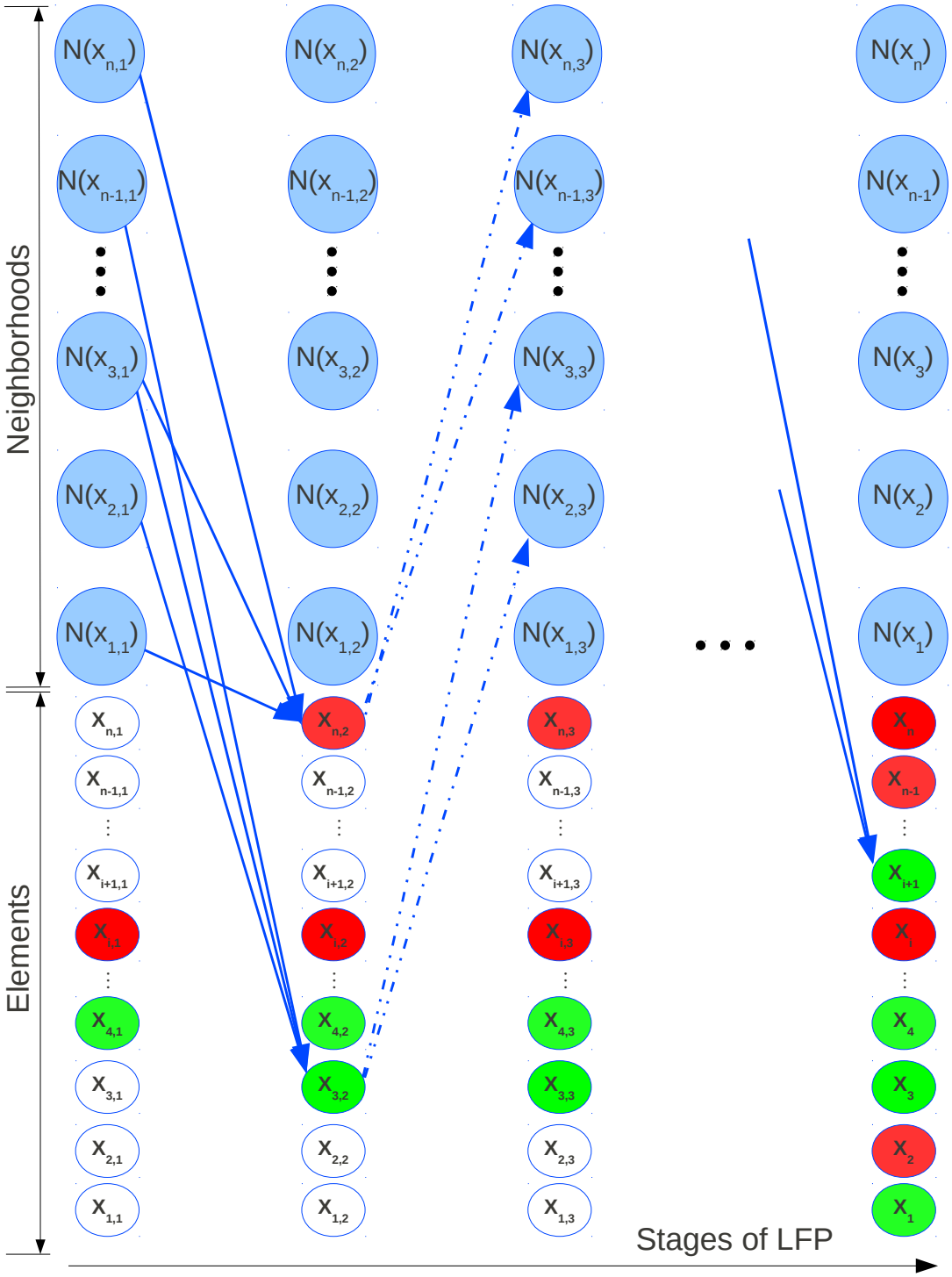

Figure 7.1: The Element-Neighborhood-Stage Product (ENSP) model for LFP_{φ}. See text for description.

Element Vertices These vertices, which encode the variables of the k-SAT instance, are represented by the smaller circles in Fig. 7.1. They therefore correspond to elements in the structure (recall that elements of the structure represent the literals in the k-SAT formula). However, *each variable in our original system $X_1, \ldots, X_n$ is represented by a different vertex at each stage of the computation*. Thus, each variable in the original system gives rise to $|\varphi^{\mathfrak{A}}|$ vertices in the ENSP model. Also recall that there are $2n$ elements in the k-SAT structure, where n is the number of variables in the SAT formula. However, in Fig 7.1, we have only shown one vertex per variable, and allowed it to be colored two colors - green indicating the variable has been assigned a value of $+1$, and red indicating the variable has been assigned the value -1. Since the underlying formula φ that is being iterated is positive, elements do not change their color once they have been assigned.

Neighborhood Vertices These vertices, denoted by the larger circles with blue shading in Fig. 7.1, represent the r-neighborhoods of the elements in the structure. Just like variables, each neighborhood is also represented by a different vertex at each stage of the LFP computation. Each of their possible values are the possible isomorphism types of the r-neighborhoods, namely, the local r-types of the corresponding element. These vertices may be thought of as vectors of size $\mathrm{poly}(\log n)$ corresponding to the cliques that occur in the neighborhood system we described in Sec. 7.2.2, or one may think of them as a single variable taking the value of the various local r-types.

Now we describe the stages of the ENSP. There are $2|\varphi^{\mathfrak{A}}|$ stages, starting from the leftmost and terminating at the rightmost. Each stage of the LFP computation is represented by two stages in the ENSP. Initially, at the start of the LFP computation, we are in the left-most stage. Here, notice that some variable vertices are colored green, and some red. In the figure, $X_{4,1}$ is green, and $X_{i,1}$ is red. This indicates that the initial partial assignment that we provided the LFP had variable X_4 assigned $+1$ and variable X_i assigned -1. In this way, a small

fraction $O(n)$ of the variables are assigned values. The LFP is asked to extend this partial assignment to a complete satisfying assignment on all variables (if it exists, and abort if not).

Let us now look at the transition to the second stage of the ENSP. At this stage, based on the conditions expressed by the formula φ in terms of their own local neighborhoods, and the existence of a bounded number of other local neighborhoods in the structure, some elements enter the relation. This means they get assigned $+1$ or -1. In the figure, the variable $X_{3,2}$ takes the color green based on information gathered from its own neighborhood $N(X_{3,1})$ and two other neighborhoods $N(X_{2,1})$ and $N(X_{n-1,1})$. This indicates that at the first stage, the LFP assigned the value $+1$ to the variable X_3. Similarly, it assigns the value -1 to variable X_n (remember that the first two stages in the ENSP correspond to the first stage of the LFP computation). The vertices that do not change state simply transmit their existing state to the corresponding vertices in the next stage by a horizontal arrow, which we do not show in the figure in order to avoid clutter.

Once some variables have been assigned values in the first stage, their neighborhoods, and the neighborhoods in their vicinity (meaning, the neighborhoods of other elements that are in their vicinity) change. This is indicated by the dotted arrows between the second and third stages of the ENSP. Note that this happens implicitly during LFP computation. That is why we have represented each stage of the actual LFP computation by two stages in the ENSP. The first stage is the explicit stage, where variables get assigned values. The second stage is the implicit stage, where variables "update their neighborhoods" and those neighborhoods in their vicinity. For example, once X_3 has been assigned the value $+1$, it updates its neighborhood and also the neighborhood of variable X_2 which lies in its vicinity (in this example). In this way, influence propagates through the structure during a LFP computation. There are two stages of the ENSP for each stage of the LFP. Thus, there are $2|\varphi^{\mathfrak{A}}|$ stages of the ENSP in all.

By the end of the computation, all variables have been assigned values, and we have a satisfying assignment. The variables at the last stage $X_{i,|\varphi^{\mathfrak{A}}|}$ are just

the original X_i. Thus, we recover our original variables $(X_1, \ldots, X_n)$ by simply looking only at the last (rightmost in the figure) level of the ENSP.

By introducing extra variables to represent each stage of each variable and each neighborhood in the SAT formula, we have accomplished our original aim. We have embedded our original set of variates into a *polynomially* larger product space, and obtained a directed graphical model on this larger space. This product space has a nice factorization due to the directed graph structure. This is what we will exploit.

Remark 7.5. The explicit stages of the ENSP also perform the task of propagating the local constraints placed by the various factors in the underlying factor graph outward into the larger graphical model. For example, in our case of the factors encoding clauses of a k-SAT formula, the local constraint placed by a clause is that the global assignment must evade exactly one restriction to a specified set of k coordinates. For example, in the case of $k = 3$ the clause $x_1 \vee x_2 \vee \neg x_3$ permits all global assignments except those whose first three coordinates are $(-1, -1, +1)$. In contrast, if the factor were a XORSAT clause, the local restrictions are all in the form of linear spaces, and so the global solution is an intersection of such spaces. k-SAT asks a question about whether certain spaces of the form

$$\{\omega \colon (\omega_{i_1}, \ldots, \omega_{i_k}) \neq (\nu_1, \ldots, \nu_k)\}$$

have non-empty intersections, where $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ and the prohibited ν_i are ± 1. Note that these are $O(1)$ local constraints per factor. In contrast, XORSAT asks the question about whether certain *linear* spaces have a non-empty intersection. Linearity is a global constraint. Of course, all messages are coded into the formula φ. Thus, the end result of multiple runs of the LFP will be a space of solutions conditioned upon the requirements. So, for instance, if we were to try to solve XORSAT formulae, we would obtain a space that would be linear.

Thus, we have a directed graph with $2n + n = 3n$ vertices at each stage, and $2|\varphi^{\mathfrak{A}}|$ stages. Since the LFP completes its computation in under a fixed polynomial number of steps, this means that we have managed to represent

the LFP computation on a structure as a directed model using a polynomial overhead in the number of parameters of our representation space. In other words, by embedding the covariates into a polynomially larger space, we have been able to put a common structure on various computations done by LFP on them. In the ENSP model, the covariates are recovered by restricting to the first n vertices in the bottom row, but now we have a graphical model that can represent the underlying computation done in order to get to that final state. Note that without embedding the covariates into a larger space, we would not be able to place the various computations done by LFP into a single graphical model. The insight that we can afford to incur a polynomial cost in order to obtain a common graphical model on a larger product space was key to this section.

7.4 Parametrization of the ENSP

Our goal is to demonstrate the following.

> *If* LFP *were able to compute solutions to the d1RSB phase of random* k-SAT*, then the distribution of the entire space of solutions would have a substantially simpler parametrization than we know it does.*

In order to accomplish this, we need to measure the growth in the dimension of independent parameters it requires to parametrize the distribution of solutions that we have just computed using LFP.

In order to do this, we have embedded our variates into a polynomially larger space that has factorization according to a directed model — the ENSP. We have seen that the cliques in the ENSP are of size $\mathrm{poly}(\log n)$. By employing the version of Hammersley-Clifford for directed models, Theorem 2.13, we also know that we can parameterize the distribution by specifying a system of potentials over its cliques, automatically ensuring conditional independence.

The directed nature of the ENSP also means that we can factor the resulting distribution into conditional probability distributions (CPDs) at each vertex of

the model of the form $P(x \mid \mathrm{pa}(x))$, and then normalize each CPD. Once again, each CPD will have scope only $\mathrm{poly}(\log n)$. From our perspective, the major benefit of directed graphical models is that we can do this always, without any added positivity constraints. Recall that positivity is required in order to apply the Hammersley-Clifford theorem to obtain factorizations for undirected models.

How do we compute the CPDs or potentials? We assign various initial partial assignments to the variables as described in Sec. 7.2.3 and let the LFP computations run. We only consider successful computations, namely those where the LFP was able to extend the partial assignment to a full satisfying assignment to the underlying k-SAT formula. We represent each stage of the LFP computation on the corresponding two stages of the ENSP and thus obtain one full instantiation of the representation space. We do this exponentially numerous times, and build up our local CPDs by simply recording local statistics over all these runs. This gives us the factorization (over the expanded representation space) of our distribution, assuming that $\mathbf{P} = \mathbf{NP}$.

The ENSP for different runs of the LFP will, in general, be different. This is because the flow of influences through the stages of the ENSP will, in general, depend on the initial partial assignment. What is important is that each such model will have some properties — such as largest clique size, which determines the order of the number of parameters — in common. Let us inspect these properties that determine the parametrization of the ENSP model.

1. There are polynomially many more vertices in the ENSP model than elements in the underlying structure.

2. Lemma 7.1 gives us a $\mathrm{poly}(\log n)$ upper bound on the size of the neighborhoods. The number of local r-types whose value each neighborhood vertex can take is $2^{\mathrm{poly}(\log n)}$.

3. By Theorem 4.8 there is a fixed constant s such that there must exist s neighborhoods in the structure satisfying certain local conditions for the formula to hold. Remember, we are presently analyzing a single stage of

the LFP. This again gives us $\mathrm{poly}(n)$ ($O(n^s)$ in this case) different possibilities for each explicit stage of the ENSP. The same can also be arrived at by utilizing the normal form of Theorem 4.9. By the previous point, each of these possibilities can be parameterized by $2^{\mathrm{poly}(\log n)}$ parameters, giving us a total of $2^{\mathrm{poly}(\log n)}$ parameters required.

4. At each implicit stage of the ENSP, we have to update the types of the neighborhoods that were affected by the induction of elements at the previous explicit stage. There are only n neighborhoods, and each has $\mathrm{poly}(\log n)$ elements at most.

The ENSP *is an interaction model where direct interactions are of size* $\mathrm{poly}(\log n)$*, and are chained together through conditional independencies.*

Proposition 7.6. *A distribution that factorizes according to the* ENSP *can be parameterized with* $2^{\mathrm{poly}(\log n)}$ *independent parameters. The scope of the factors in the parametrization grows as* $\mathrm{poly}(\log n)$*.*

This also underscores the principle that the description of the parameter space is simpler because it only involves interactions between l variates at a time directly, and then chains these together through conditional independencies. In the case of the LFP neighborhood system, the size of the largest cliques are $\mathrm{poly}(\log n)$ for each single run of the LFP. This will not change if we were computing using complex fixed points since the space of k-types is only polynomially larger than the underlying structure.

The crucial property of the distribution of the ENSP is that it admits a recursive factorization. This is what drastically reduces the parameter space required to specify the distribution. It also allows us to parametrize the ENSP by simply specifying potentials on its maximal cliques, which are of size $\mathrm{poly}(\log n)$.

While the entire distribution obtained by LFP may not factor according to any one ENSP, it is a mixture of distributions each of whom factorizes as per some ENSP. Next, we analyze the features of such a mixture when exponentially many instantiations of it are provided. As the reader may intuit, when such a mixture is asked to provide exponentially many samples, these will show

features of scope $\mathrm{poly}(\log n)$. This is simply a statement about the paucity of independent parameters in the component distributions in the mixture.

7.5 Separation

The property of the ENSP that allows us to analyze the behavior of mixtures is that it is specified by local Gibbs potentials on its cliques. In other words, a variable interacts with the rest of the model only through the cliques that it is part of. These cliques are parametrized by potentials. We may think of the cliques as the building blocks of each ENSP. The cliques are also upper bounded in size by $\mathrm{poly}(\log n)$. Furthermore, a vertex may be in at most $O(\log n)$ such cliques. Therefore, a vertex displays collective behavior only of range $\mathrm{poly}(\log n)$. Thus, the mixture comprises distributions that can be parametrized by a subspace of $\mathbb{R}^{\mathrm{poly}(\log n)}$, in contrast to requiring the larger space $\mathbb{R}^{O(n)}$. This means that when exponentially many solutions are generated, the features in the mixture will be of size $\mathrm{poly}(\log n)$, not of size $O(n)$.

This explains why polynomial time algorithms fail when interactions between variable are $O(n)$ at a time, without the possibility of factoring into smaller pieces through conditional independencies. This also puts on rigorous ground the empirical observation that even **NP**-complete problems are easy in large regimes, and become hard only when the densities of constraints increase above a certain threshold. This threshold is precisely the value where $O(n)$ interactions first appear in almost all randomly constructed instances.

In case of random k-SAT in the d1RSB phase, these irreducible $O(n)$ interactions manifest through the appearance of cores which comprise clauses whose variables are coupled so tightly that one has to assign them "simultaneously." Cores arise when a set of $C = O(n)$ clauses have all their variables also lying in a set of size C. Once clause density is sufficiently high, cores cannot be assigned $\mathrm{poly}(\log n)$ at a time, and successive such assignments chained together through conditional independencies. Since cores do not factor through conditional independencies, this makes it impossible for polynomial time algorithms to assign

their variables correctly. Intuitively, variables in a core are so tightly coupled together that they can only vary jointly, without any conditional independencies between subsets. In other words, they represent irreducible interactions of size $O(n)$ which may not be factored any further. In such cases, parametrization over cliques of size only $\mathrm{poly}(\log n)$ is insufficient to specify the joint distribution.

However, we have shown that in the ENSP model, the size of the largest such irreducible interactions are $\mathrm{poly}(\log n)$, not $O(n)$. Furthermore, since the model is directed, it guarantees us conditional independencies at the level of its largest interactions. More precisely, it guarantees us that there will exist conditional independencies in sets of size larger than the largest cliques in its moral graph, which are $O(\mathrm{poly}(\log n))$. In other words, there will be independent variation *within* cores when conditioned upon values of intermediate variables that also lie within the core, should the core factorize as per the ENSP. This is illustrated in Fig. 7.2. This is contradictory to the known behaviour of cores for sufficiently high values of k and clause density in the d1RSB phase. In other words, while the space of solutions generated by LFP has features of size $\mathrm{poly}(\log n)$, the features present in cores in the d1RSB phase have size $O(n)$.

> *The framework we have constructed allows us to analyze the set of polynomial time algorithms simultaneously, since they can all be captured by some* LFP*, instead of dealing with each individual algorithm separately. It makes precise the notion that polynomial time algorithms can take into account only interactions between variables that grow as* $\mathrm{poly}(\log n)$*, not as* $O(n)$*.*

At this point, we are ready to state our main theorem.

Theorem 7.7. $\mathbf{P} \neq \mathbf{NP}$.

Proof. Consider the solution space of k-SAT in the d1RSB phase for $k > 8$ as recalled in Section.5.2.1. We know that for high enough values of the clause density α, we have $O(n)$ frozen variables in almost all of the exponentially many clusters. Let us consider the situation where these clusters were generated by a

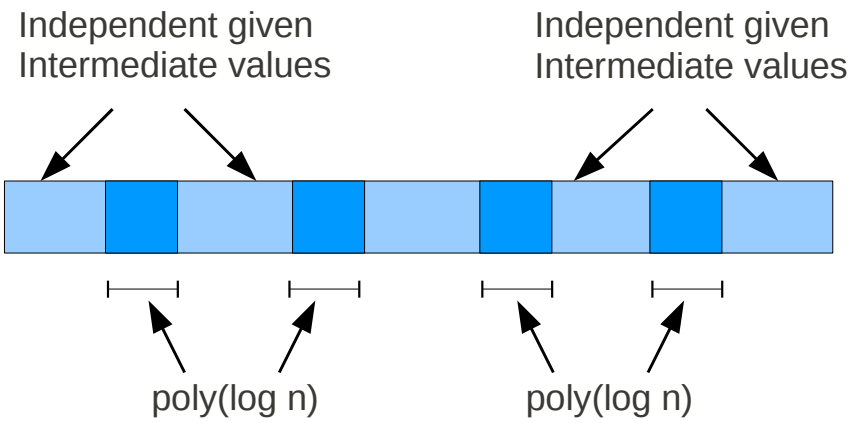

Figure 7.2: The factorization within a core due to potentials of size $\mathrm{poly}(\log n)$.

purported LFP algorithm for k-SAT. However, when exponentially many solutions have been generated from distributions having the parametrization of the ENSP model, we will see the effect of conditional independencies beyond range $\mathrm{poly}(\log n)$. Let $\alpha\beta\gamma$ be a representation of the variables in cliques α, β and γ, then given a value of β, we will see independent variation over all their possible conditional values in the variables of α and γ. If each set of such variables has scope at most $\mathrm{poly}(\log n)$, then this means that once more than $c^{\mathrm{poly}(\log n)}, c > 1$ many distinct solutions are generated, we have non-trivial conditional distributions conditioned upon values of β variables (this factor accounts for the possible orderings within the $\mathrm{poly}(\log n)$ variables as well). At this point, the conditional independence ensure that we will see cross terms of the form

$$\alpha_1\beta\gamma_1 \qquad \alpha_2\beta\gamma_2 \qquad \alpha_1\beta\gamma_2 \qquad \alpha_2\beta\gamma_1.$$

Note that since $O(n)$ variables have to be changed when jumping from one cluster to another, we may even chose our $\mathrm{poly}(\log n)$ blocks to be in overlaps of these variables. This would mean that with a $\mathrm{poly}(\log n)$ change in frozen variables of one cluster, we would get a solution in another cluster. But we know that in the highly constrained phases of d1RSB, we need $O(n)$ variable flips to get from one cluster to the next. This gives us the contradiction that we seek. ■

The basic question in analyzing such mixtures is: How many variables do

we need to condition upon in order to split the distribution into conditionally independent pieces? The answer is given by (a) the size of the largest cliques and (b) the number of such cliques that a single variable can occur in. In our case, these two give us a $\mathrm{poly}(\log n)$ quantity. When exponentially many solutions have been generated, there will be conditional distributions that exhibit conditional independence between blocks of variates size $\mathrm{poly}(\log n)$. Namely, there will be no effect of the values of one upon those of the other. This is what prevents the Hamming distance between solutions from being $O(n)$. This is shown pictorially in Fig. 7.2.

We may think of such mixtures as possessing only $c^{\mathrm{poly}(\log n)}$ "channels" to communicate directly with other variables. All long range correlations transmitted in such a distribution must pass through only these many channels. Therefore, exponentially many solutions cannot independently transmit $O(n)$ correlations (namely, the variables that have to be changed when jumping from one cluster to another). Their correlations must factor through this bottleneck, which gives us conditional independences after range $\mathrm{poly}(\log n)$. This means that blocks of size larger than this are now varying independently of each other conditioned upon some intermediate variables. This gives us the cross-terms described earlier, and prevents the Hamming distance from being $O(n)$ on the average over exponentially many solutions. Instead, it must be $\mathrm{poly}(\log n)$.

We can see that due to the limited parameter space that determines each variable, it can only display a limited behavior. This behavior is completely determined by $\mathrm{poly}(\log n)$ other variates, not by $O(n)$ other variates. Thus, the "jointness" in this distribution lies at a level $\mathrm{poly}(\log n)$. This is why when enough solutions have been generated by the LFP, the resulting distribution will start showing features that are at most of size $\mathrm{poly}(\log n)$. In other words, there will be solutions that show cross-terms between features whose size is $\mathrm{poly}(\log n)$.

We collect some observations in the following.

Remark 7.8. The $\mathrm{poly}(\log n)$ size of features and therefore Hamming distance between solutions tells us that polynomial time algorithms correspond to the

RS phase of the 1RSB picture, not to the d1RSB phase.

Remark 7.9. We can see from the preceding discussion that the number of independent parameters required to specify the distribution of the entire solution space in the d1RSB phase (for $k > 8$) rises as $c^n, c > 1$. This is because it takes that many parameters to specify the exponentially many $O(n)$ variable "jumps" between the clusters. These jumps are independent, and cannot be factored through $\mathrm{poly}(\log n)$ sized factors since that would mean conditional independence of pieces of size $\mathrm{poly}(\log n)$ and would ensure that the Hamming distance between solutions was of that order.

Remark 7.10. Note that the central notion is that of the number of independent parameters, not frozen variables. For example, frozen variables would occur even in low dimensional parametrizations in the presence of additional constraints placed by the problem. This is what happens in XORSAT, where the linearity of the problem causes frozen variables to occur. The frozen variables in XORSAT do not arise due to a high dimensional parameterization, but simply because the 2-core percolates [MM09, §18.3]. Each cluster is a linear space tagged on to a solution for the 2-core, which is also why the clusters are all of the same size. Linear spaces always admit a simple description as the linear span of a *basis*, which takes the order of $\log$ of the size of the space.

Remark 7.11. It is tempting to think that there will be such a parametrization whenever the algorithmic procedure used to generate the solutions is stage-wise local. This is not so. We need the added requirement that "mistakes" are not allowed. Namely, we cannot change a decision that has been made. Otherwise, even PFP has the stage-wise bounded local property, but it can give rise to distributions without any conditional independence factorizations whose factors are of size $\mathrm{poly}(\log n)$. When placed in the ENSP, we see that there is factorization, but over an exponentially larger space, where clique sizes are of exponential size. One might observe that the requirement that we not make any trial and error at all that limits LFP computations in a fundamentally different manner than the locality of information flows. See [Put65] for an interesting related notion of "trial and error predicates" in computability theory.

7.6 Some Perspectives

The following perspectives are reinforced by this work.

1. The most natural object of study for constraint satisfaction problems is the *entire* space of solutions. It is this space where the dependencies and independencies that the CSP imposes upon covariates that satisfy it manifest.

2. There is an intimate relation between the geometry of the space and its parametrization. Studying the parametrization of the space of solutions is a worthwhile pursuit.

3. The view that an algorithm is a means to generate *one* solution is limited in the sense that it is oblivious to the geometry of the space of *all* solutions. It may, of course, be the appropriate approach in many applications. But there are applications where requiring algorithms to generate numerous solutions and approximate with increasing accuracy the entire space of solutions seems more natural.

4. Conditional independence over factors of small scope is at the heart of resolving CSPs by means of polynomial time algorithms. In other words, polynomial time algorithms succeed by successively "breaking up" the problem into smaller subproblems that are joined to each other through conditional independence. Consequently, polynomial time algorithms cannot solve problems in regimes where blocks whose order is the same as the underlying problem instance require simultaneous resolution.

5. Polynomial time algorithms resolve the variables in CSPs in a certain order, and with a certain structure. This structure is important in their study. In order to bring this structure under study, we may have to embed the space of covariates into a larger space (as done by the ENSP).

A. Reduction to a Single LFP Operation

A.1 The Transitivity Theorem for LFP

We now gather a few results that will enable us to cast any LFP into one having just one application of the LFP operator. Since we use this construction to deal with complex fixed points, we reproduce it in this appendix. The presentation here closely follows [EF06, Ch. 8].

The first result, known as the transitivity theorem, tells us that nested fixed points can always be replaced by simultaneous fixed points. Let $\varphi(\mathbf{x}, X, Y)$ and $\psi(\mathbf{y}, X, Y)$ be first order formulas positive in X and Y. Moreover, assume that no individual variable free in $\mathsf{LFP}_{\mathbf{y},Y}\psi(\mathbf{y}, X, Y)$ gets into the scope of a corresponding quantifier or LFP operator in A.1.

$$[\mathsf{LFP}_{\mathbf{x},X}\varphi(\mathbf{x}, X, [\mathsf{LFP}_{\mathbf{y},Y}\psi(\mathbf{y}, X, Y)])]\mathbf{t} \tag{A.1}$$

Then A.1 is equivalent to a formula of the form

$$\exists(\forall)u[\mathsf{LFP}_{\mathbf{z},Z,\chi}(\mathbf{z}, Z)]\mathbf{u},$$

where χ is first order.

A.2 Sections and the Simultaneous Induction Lemma for LFP

Next we deal with simultaneous fixed points. Recall that simultaneous inductions do not increase the expressive power of LFP. The proof utilizes a coding procedure whereby each simultaneous induction is embedded as a section in a single LFP operation of higher arity. First, we introduce the notion of a section.

Definition A.1. Let R be a relation of arity $(k+l)$ on A and $\mathbf{a} \in A^k$. Then the $\mathbf{a}$-section of R, denoted by $R_{\mathbf{a}}$, is given by

$$R_{\mathbf{a}} := \{\mathbf{b} \in A^k | R(\mathbf{ba})\}$$

Next we see how sections can be used to encode multiple simultaneous operators producing relations of lower arity into a single operator producing a relation of higher arity. Let m operators $F_1, \ldots, F_m$ act as follows:

$$\begin{aligned} F_1 &\colon (A^{k_1}) \times \cdots \times (A^{k_m}) \to (A^{k_1}) \\ F_2 &\colon (A^{k_1}) \times \cdots \times (A^{k_m}) \to (A^{k_2}) \\ &\vdots \\ F_m &\colon (A^{k_1}) \times \cdots \times (A^{k_m}) \to (A^{k_m}) \end{aligned} \tag{A.2}$$

We wish to embed these operators as sections of a "larger" operator, which is known as their *simultaneous join*.

We will denote a tuple consisting only of a's by $\tilde{a}$. The length of $\tilde{a}$ be clear from context.

Definition A.2. Let $F_1, \ldots, F_m$ be operators acting as above. Set

$$k := \max\{k_1, \ldots, k_m\} + m + 1.$$

The simultaneous join of $F_1, \ldots, F_m$, denoted by $J(F_1, \ldots, F_m)$, is an operator acting as

$$J(F_1, \ldots, F_m) \colon (A^k) \to (A^k)$$

such that for any $a, b \in A$, the $\tilde{a}b^i$-section (where the length of $\tilde{a}$ here is $k - i + 1$) of the n^{th} power of J is the n^{th} power of the operator F_i. Concretely, the simultaneous join is given by

$$J(R) := \bigcup_{a,b \in A, a \neq b} ((F_1(R_{\tilde{a}b^1}, \ldots, R_{\tilde{a}b^m}) \times \{\tilde{a}b^1\}) \cup \cdots \cdots \cup (F_m(R_{\tilde{a}b^1}, \ldots, R_{\tilde{a}b^m}) \times \{\tilde{a}b^m\})). \quad \text{(A.3)}$$

The simultaneous join operator defined above has properties we will need to use. These are collected below.

Lemma A.3. *The i^{th} power J^i of the simultaneous join operator satisfies*

$$J^i = \bigcup_{a,b \in A, a \neq b} ((F_1^i \times \{\tilde{a}b^1\}) \cup \cdots \cup (F_m^i \times \{\tilde{a}b^m\})). \quad \text{(A.4)}$$

The following corollaries are now immediate.

Corollary A.4. *The fixed point J^∞ of the simultaneous join of operators $(F_1, \ldots, F_m)$ exists if and only if their simultaneous fixed point $(F_1^\infty, \ldots, F_m^\infty)$ exists.*

Corollary A.5. *The simultaneous join of inductive operators is inductive.*

Finally, we need to show that the simultaneous join can itself be expressed as a LFP computation. We need formulas that will help us define sections of a simultaneous induction. Since the sections are coded using tuples of the form $a^{k-i+k_i+1}b^i$, we will need formulas that can express this.

Definition A.6. For $\ell \geq 1$ and $i = 1, \ldots, \ell$, the section formulas $\delta_i^l(x_1, \ldots, x_l, v, w)$

$$\delta_i^l(x_1, \ldots, x_l, v, w) := \begin{cases} \neg(v = w) \wedge (x_1 = \cdots = x_\ell = v) & i = 1 \\ \neg(v = w) \wedge (x_1 = \cdots = x_{\ell-i+1} = v) \wedge & \\ \quad (x_{\ell-i+2} = \cdots = w) & i > 1. \end{cases} \quad \text{(A.5)}$$

For distinct $a, b \in \mathfrak{A}$, $\mathfrak{A} \models \delta_i[\tilde{a}b^j ab]$ if and only if $i = j$.

Now we are ready to show that simultaneous fixed-point inductions of formulas can be replaced by the fixed point induction of a single formula.

Definition A.7. Let

$$\varphi_1(R_1, \ldots, R_m, \mathbf{x}_1), \ldots, \varphi_m(R_1, \ldots, R_m, \mathbf{x}_m)$$

be formulas of LFP. As always, we let R_i be a k_i-ary relation and $\mathbf{x}_i$ be a k_i-tuple. Furthermore, let $\varphi_1, \ldots, \varphi_m$ be positive in $R_1, \ldots, R_m$. Set $k := \max\{k_1, \ldots, k_m\} + m + 1$. Define a new first order formula χ_J having k variables and computing a single k-ary relation Z by

$$\begin{aligned}\chi_J(Z, z_1, \ldots, z_k) := \exists v \exists w (\neg v = w \wedge \\ ((\varphi_1(Z_{\tilde{v}w^1}, \ldots, Z_{\tilde{v}w^m}, z_1, \ldots, z_k) \wedge \delta_1^k(z_1, \ldots, z_k, v, w)) \\ \vee (\varphi_2(Z_{\tilde{v}w^1}, \ldots, Z_{\tilde{v}w^m}, z_1, \ldots, z_k) \wedge \delta_2^k(z_1, \ldots, z_k, v, w)) \\ \vdots \\ \vee (\varphi_m(Z_{\tilde{v}w^1}, \ldots, Z_{\tilde{v}w^m}, z_1, \ldots, z_k) \wedge \delta_m^k(z_1, \ldots, z_k, v, w)))) \quad \text{(A.6)}\end{aligned}$$

Then, the relation computed by the least fixed point of χ_J contains all the individual least fixed points computed by the simultaneous induction as its sections.

Bibliography

[ACO08] D. Achlioptas and A. Coja-Oghlan. Algorithmic barriers from phase transitions. *arXiv:0803.2122v2 [math.CO]*, 2008.

[AM00] Srinivas M. Aji and Robert J. McEliece. The generalized distributive law. *IEEE Trans. Inform. Theory*, 46(2):325–343, 2000.

[AP04] Dimitris Achlioptas and Yuval Peres. The threshold for random k-SAT is $2^k \log 2 - O(k)$. *J. Amer. Math. Soc.*, 17(4):947–973 (electronic), 2004.

[ART06] Dimitris Achlioptas and Federico Ricci-Tersenghi. On the solution-space geometry of random constraint satisfaction problems. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 130–139. ACM, New York, 2006.

[AV91] Serge Abiteboul and Victor Vianu. Datalog extensions for database queries and updates. *J. Comput. Syst. Sci.*, 43(1):62–124, 1991.

[AV95] Serge Abiteboul and Victor Vianu. Computing with first-order logic. *Journal of Computer and System Sciences*, 50:309–335, 1995.

[BDG95] José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural complexity. I.* Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, second edition, 1995.

[Bes74] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Statist. Soc. Ser. B*, 36:192–236, 1974. With dis-

cussion by D. R. Cox, A. G. Hawkes, P. Clifford, P. Whittle, K. Ord, R. Mead, J. M. Hammersley, and M. S. Bartlett and with a reply by the author.

[BGS75] Theodore Baker, John Gill, and Robert Solovay. Relativizations of the $\mathcal{P} =? \mathcal{NP}$ question. *SIAM J. Comput.*, 4(4):431–442, 1975.

[Bis06] Christopher M. Bishop. *Pattern recognition and machine learning*. Information Science and Statistics. Springer, New York, 2006.

[BMW00] G Biroli, R Monasson, and M Weigt. A variational description of the ground state structure in random satisfiability problems. *PHYSICAL JOURNAL B*, 568:551–568, 2000.

[CF86] Ming-Te Chao and John V. Franco. Probabilistic analysis of two heuristics for the 3-satisfiability problem. *SIAM J. Comput.*, 15(4):1106–1118, 1986.

[CKT91] Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Where the really hard problems are. In *IJCAI*, pages 331–340, 1991.

[CO09] A. Coja-Oghlan. A better algorithm for random k-sat. *arXiv:0902.3583v1 [math.CO]*, 2009.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, New York, NY, USA, 1971. ACM Press.

[Coo06] Stephen Cook. The P versus NP problem. In *The millennium prize problems*, pages 87–104. Clay Math. Inst., Cambridge, MA, 2006.

[Daw79] A. P. Dawid. Conditional independence in statistical theory. *J. Roy. Statist. Soc. Ser. B*, 41(1):1–31, 1979.

[Daw80] A. Philip Dawid. Conditional independence for statistical operations. *Ann. Statist.*, 8(3):598–617, 1980.

[DLW95] Anuj Dawar, Steven Lindell, and Scott Weinstein. Infinitary logic and inductive definability over finite structures. *Inform. and Comput.*, 119(2):160–175, 1995.

[DMMZ08] Hervé Daudé, Marc Mézard, Thierry Mora, and Riccardo Zecchina. Pairs of sat-assignments in random boolean formulæ. *Theor. Comput. Sci.*, 393(1-3):260–279, 2008.

[Dob68] R. L. Dobrushin. The description of a random field by means of conditional probabilities and conditions on its regularity. *Theory Prob. Appl.*, 13:197–224, 1968.

[Edm65] Jack Edmonds. Minimum partition of a matroid into independents subsets. *Journal of Research of the National Bureau of Standards*, 69:67–72, 1965.

[EF06] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, enlarged edition, 2006.

[Fag74] Ronald Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In *Complexity of computation (Proc. SIAM-AMS Sympos. Appl. Math., New York, 1973)*, pages 43–73. SIAM–AMS Proc., Vol. VII. Amer. Math. Soc., Providence, R.I., 1974.

[Fri99] E. Friedgut. Necessary and sufficient conditions for sharp thresholds and the k-sat problem. *J. Amer. Math. Soc.*, 12(20):1017–1054, 1999.

[Gai82] Haim Gaifman. On local and nonlocal properties. In *Proceedings of the Herbrand symposium (Marseilles, 1981)*, volume 107 of *Stud. Logic Found. Math.*, pages 105–135, Amsterdam, 1982. North-Holland.

[GG84] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Trans-*

actions on Pattern Analysis and Machine Intelligence, 6(6):721–741, November 1984.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and intractability*. W. H. Freeman and Co., San Francisco, Calif., 1979. A guide to the theory of NP-completeness, A Series of Books in the Mathematical Sciences.

[GS00] Martin Grohe and Thomas Schwentick. Locality of order-invariant first-order formulas. *ACM Trans. Comput. Log.*, 1(1):112–130, 2000.

[Han65] William Hanf. Model-theoretic methods in the study of elementary logic. In *Theory of Models (Proc. 1963 Internat. Sympos. Berkeley)*, pages 132–145. North-Holland, Amsterdam, 1965.

[HC71] J. M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. 1971.

[HH76] J. Hartmanis and J. E. Hopcroft. Independence results in computer science. *SIGACT News*, 8(4):13–24, 1976.

[Hod93] Wilfrid Hodges. *Model theory*, volume 42 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1993.

[Imm82] Neil Immerman. Relational queries computable in polynomial time (extended abstract). In *STOC '82: Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 147–152, New York, NY, USA, 1982. ACM.

[Imm86] Neil Immerman. Relational queries computable in polynomial time. *Inform. and Control*, 68(1-3):86–104, 1986.

[Imm99] Neil Immerman. *Descriptive complexity*. Graduate Texts in Computer Science. Springer-Verlag, New York, 1999.

[Kar72] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[KF09] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[KFaL98] Frank R. Kschischang, Brendan J. Frey, and Hans andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 1998.

[KMRT+07] Florent Krząkała, Andrea Montanari, Federico Ricci-Tersenghi, Guilhem Semerjian, and Lenka Zdeborová. Gibbs states and the set of solutions of random constraint satisfaction problems. *Proc. Natl. Acad. Sci. USA*, 104(25):10318–10323 (electronic), 2007.

[KS80] R. Kinderman and J. L. Snell. Markov random fields and their applications. *American Mathematical Society*, 1:1–142, 1980.

[KS94] Scott Kirkpatrick and Bart Selman. Critical behavior in the satisfiability of random boolean formulae. *Science*, 264:1297–1301, 1994.

[KSC84] Harri Kiiveri, T. P. Speed, and J. B. Carlin. Recursive causal models. *J. Austral. Math. Soc. Ser. A*, 36(1):30–52, 1984.

[Lau96] Steffen L. Lauritzen. *Graphical models*, volume 17 of *Oxford Statistical Science Series*. The Clarendon Press Oxford University Press, New York, 1996. Oxford Science Publications.

[LDLL90] S. L. Lauritzen, A. P. Dawid, B. N. Larsen, and H.-G. Leimer. Independence properties of directed Markov fields. *Networks*, 20(5):491–505, 1990. Special issue on influence diagrams.

[Lev73] Leonid A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3), 1973.

[Li09] Stan Z. Li. *Markov random field modeling in image analysis*. Advances in Pattern Recognition. Springer-Verlag London Ltd., London, third edition, 2009. With forewords by Anil K. Jain and Rama Chellappa.

[Lib04] Leonid Libkin. *Elements of finite model theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2004.

[Lin05] S. Lindell. Computing monadic fixed points in linear time on doubly linked data structures. *available online at http://citeseerx.ist.psu.edu/doi=10.1.1.122.1447*, 2005.

[MA02] Cristopher Moore and Dimitris Achlioptas. Random k-sat: Two moments suffice to cross a sharp threshold. *FOCS*, pages 779–788, 2002.

[MM09] Marc Mézard and Andrea Montanari. *Information, physics, and computation*. Oxford Graduate Texts. Oxford University Press, Oxford, 2009.

[MMW07] Elitza Maneva, Elchanan Mossel, and Martin J. Wainwright. A new look at survey propagation and its generalizations. *J. ACM*, 54(4):Art. 17, 41 pp. (electronic), 2007.

[MMZ05] M. Mézard, T. Mora, and R. Zecchina. Clustering of solutions in the random satisfiability problem. *Phys. Rev. Lett.*, 94(19):197–205, May 2005.

[Mos74] Yiannis N. Moschovakis. *Elementary induction on abstract structures*. North-Holland Publishing Co., Amsterdam, 1974. Studies in Logic and the Foundations of Mathematics, Vol. 77.

[Mou74] John Moussouris. Gibbs and Markov random systems with constraints. *J. Statist. Phys.*, 10:11–33, 1974.

[MPV87] Marc Mézard, Giorgio Parisi, and Miguel Angel Virasoro. *Spin glass theory and beyond*, volume 9 of *World Scientific Lecture Notes in Physics*. World Scientific Publishing Co. Inc., Teaneck, NJ, 1987.

[MPZ02] M Mèzard, G Parisi, and R Zecchina. Analytic and Algorithmic Satisfiability Problems. *Science*, 297(August):812–815, 2002.

[MSL92] David Mitchell, Bart Selman, and Hector Levesque. Hard and easy distributions of sat problems. In *AAAI*, pages 459–465, 1992.

[MZ97] Rémi Monasson and Riccardo Zecchina. Statistical mechanics of the random k-satisfiability model. *Phys. Rev. E*, 56(2):1357–1370, Aug 1997.

[MZ02] Marc Mézard and Riccardo Zecchina. Random k-satisfiability problem: From an analytic solution to an efficient algorithm. *Phys. Rev. E*, 66(5):056126, Nov 2002.

[Put65] Hilary Putnam. Trial and error predicates and the solution to a problem of mostowski. *J. Symb. Log.*, 30(1):49–57, 1965.

[RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. System Sci.*, 55(1, part 1):24–35, 1997. 26th Annual ACM Symposium on the Theory of Computing (STOC '94) (Montreal, PQ, 1994).

[SB99] Thomas Schwentick and Klaus Barthelmann. Local normal forms for first-order logic with applications to games and automata. In *Discrete Mathematics and Theoretical Computer Science*, pages 444–454. Springer Verlag, 1999.

[See96] Detlef Seese. Linear time computable problems and first-order descriptions. *Math. Structures Comput. Sci.*, 6(6):505–526, 1996. Joint COMPUGRAPH/SEMAGRAPH Workshop on Graph Rewriting and Computation (Volterra, 1995).

[Sip92] Michael Sipser. The history and status of the p versus NP question. pages 603–618, 1992.

[Sip96] Michael Sipser. *Introduction to the Theory of Computation*. Course Technology, December 1996.

[Var82] Moshe Y. Vardi. The complexity of relational query languages (extended abstract). In *STOC '82: Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 137–146, New York, NY, USA, 1982. ACM.

[Wig07] Avi Wigderson. P, NP, and Mathematics - a computational complexity perspective. *Proceedings of the ICM 2006*, 1:665–712, 2007.