On the Complexity of Constraint Satisfaction Problems
A Model Theoretic Approach

CO401J Project - Final Report

19th of June 2019

Imperial College London

Author
A. Papadopoulos

Supervisors
Dr. C. Kestner
Dr. Z. Ghadernezhad

Second Marker
Prof. K. Buzzard

Abstract

Constraint Satisfaction Problems (CSPs) have played a crucial role in recent developments in the field of Computational Complexity. The Dichotomy Conjecture of Finite Domain CSPs by Feder and Vardi [1] was finally proved two years ago (independently in [2] and [3]).

Ever since the conjecture was proposed, the study of CSPs using various techniques both from pure mathematics (such as Model Theory, Topology, and Group Theory) and theoretical computer science (for instance, Datalog and Descriptive Complexity) has flourished. Even though a vast amount of research is being carried out on the topic, the literature remains somewhat convoluted and in general inaccessible.

In this report, we attempt to give a concise presentation of recent works on the model theoretic approach to constraint satisfaction. Since the literature on Model Theory and Complexity Theory has been standardised over the last 30 years or so (the standard references include [4] for Model Theory, [5] for Classical Complexity Theory and [6] for Descriptive Complexity), we only present a short summary of the results we need from these areas. A more in-depth presentation of CSPs is given, in an attempt to provide a clearer entry point to current research.

Acknowledgements

I would like, first of all, to thank my supervisor, Zaniar not only for proposing this topic, but also for the long and interesting discussions and his overall helpful comments and immense support. Without his help, this project would not have existed.

I would also like to thank Ramon for his presence, Marios for his absence, and Myrto, for both. Finally, I would like to thank A. Doxiadis and C. Papadimitriou for writing Logicomix and my parents for buying it when I was 10. If it were not for this wonderful book I would never have studied maths in the first place.

Contents

Chapter 1

Introduction

Constraint Satisfaction Problems (CSPs) have played a crucial role in recent developments in the field of Computational Complexity. The Dichotomy Conjecture of Finite Domain CSPs by Feder and Vardi [1] was finally proved two years ago independently by Bulatov in [2] and Zhuk in [3].

Ever since the conjecture was proposed, the study of CSPs using various techniques both from pure mathematics (such as Model Theory, Topology, and Group Theory) and theoretical computer science (for instance, Datalog and Descriptive Complexity) has flourished. Even though a vast amount of research is being carried out on the topic, the literature remains somewhat convoluted and in general inaccessible.

1.1 Introduction

In general, many problems, from almost all areas of computer science can be cast as CSPs. At the same time, as we show, it is very natural to study CSPs using mathematical techniques. This means that CSPs lie at a very interesting intersection point of pure mathematics and theoretical computer science.

There are many approaches, originating from different branches of pure mathematics that can be used to study CSPs. In this report, we focus on one: Model Theory. We present an overview of some of the popular model theoretic techniques that have been used in the last decade to study CSPs.

1.2 Contributions

The literature on which the background of this report is based is standard. We have made attempted to include the core results from Model Theory and Complexity Theory that we need, but our goal was never to try and summarise, in full, these vast fields.

Instead, we focus our attention mainly on CSPs. There, we present results in more detail, always giving precise definitions, in an attempt to make the presentation self-contained. The main contributions of this project have been the following:

- A concise, organised, and relatively self-contained survey on some of the modern techniques in CSPs.
- Some minor corrections to arguments from [7] and [8], which are indicated.
- A discussion of some open questions about the descriptive complexity of certain classes of CSPs (see 4.2), not present in the current literature, and a proposal of certain new questions of similar flavour.

In some sense, this report is more a presentation of applications of model theoretic techniques in complexity theory than anything else.

1.3 Outline of the Report

The following chapters of the report are broken down as follows:

- Chapter 2 is split into two sections; one focusing on Model Theory (Section 2.1), and one on Complexity Theory (Section 2.2). The aim is to provide a quick overview of the technical background needed for the next chapter.
- In Chapter 3, which is based on [7] and [8], we discuss Constraint Satisfaction Problems in depth. The exploration is fairly self-contained, but proofs of very deep results are not given. Besides a general survey of the area, we completely show the following strict inclusions:

$$\texttt{CSP} \sim_{\mathbf{P}} MM\mathbf{SNP} \subsetneq \texttt{CSP}^{\star} \subsetneq M\mathbf{SNP} \subsetneq \mathbf{SNP} \sim_{\mathbf{P}} \mathbf{SO}\exists = \mathbf{NP}.$$

- In Chapter 4, we summarise the discussion of the report. We give an informal presentation of some other approaches that can be used in conjunction with Model Theory to study CSPs. We present some open questions and give some closing thoughts.

In general, we omit standard proofs with two exceptions. We give a complete proof of Fraïssé's Theorem (Theorem 2.1.8) and Fagin's Theorem (Theorem 2.2.1). These results lie at the heart of the model theoretic approach for CSPs, and their proofs, despite being technical, are elegant and offer insight on how various results arise.

We will only be discussing countable structures and one of the focal points of the report is the "jump" from finite to infinite structures. However, fundamentally, we are interested in computation and hence infinities are not really sensible in this context. As we will see, however, they may arise naturally even then.

Most of the infinite structures we care about will thus have some finiteness properties. In particular, they will be very homogeneous, this for us means that any isomorphism of finite substructures can be extended to an automorphism of the whole structure.

Since our structures are countable, there is an underlying philosophy of back and forth games being played. We do not actually need to formalise back and forth arguments, even though many proofs use arguments of their flavour. To this end, a section on Back and Forth is delegated to Appendix A.

Chapter 2

Background

Model Theory and Complexity Theory are somewhat "terminology heavy" areas, and in this chapter we attempt to summarise the machinery that will be used in the discussion of CSPs, in the next chapter.

2.1 Basic Model Theory

In this section, we introduce some preliminary terminology from Model Theory. Where we deviate from the treatment of [4], this will be indicated. In those cases, our approach will be more along the lines of the exposition in [7] and [8].

2.1.1 Structures, Signatures and Languages

We assume that the reader is familiar with first-order logic and basic model theory. In particular, we assume familiarity with the definitions of a *signature*, a *structure*, and of a *language*.

Our standard notation will be τ, σ, ... for finite or countable signatures. We say that τ is *relational* if it contains no function or constant symbols. Note that the discussion on constraint satisfaction will focus on finite relational signatures, that is, signatures consisting only of relation symbols $R_1, \ldots, R_n$. Signatures will always include a binary relation symbol $=$ for equality which is always interpreted as "true equality".

Where possible, we develop the theory for signatures with countably many relation symbols $R_1, R_2, \ldots$ and countably many constant symbols $c_1, c_2, \ldots$. This is because in certain arguments we will need to use expanded signatures that contain constant symbols naming the elements of a structure. Therefore, unless otherwise stated, we assume that signatures are of this form.

In general, we write Γ, Δ, ... for relational τ-structures to indicate that we think of them as (hyper)graphs, otherwise we may write $A, B, \ldots$. In particular, if a signature τ has a single binary relation symbol we interpret τ-structures as (not necessarily directed) graphs. We write $\mathrm{dom}(\Gamma)$ for the domain of Γ.

Unlike [4], and most of the literature on Model Theory, in general, we will make no distinction between signatures and first-order languages. When discussing first-order formulas, we mean finite length formulas with finitely many variables. We are not directly interested in infinitary languages, and so we restrict ourselves to discussing signatures. In this section, when we discuss τ-formulas we will mean first-order τ-formulas.

Syntactic restrictions of first-order formulas play an important role in characterising classes of problems (See Section 3.2.4). We summarise some important classes here. In Chapter 3 we will give more characterisations of this form.

Definition 2.1.1. Let τ be a fixed signature.

- An *atomic τ-formula* is a formula of the form $R(k_1, \ldots, k_n)$, where R is an n-ary relation symbol from τ and each k_i is either a variable, x_i, or a constant symbol c_i of τ.
- A first-order τ-formula is called a *τ-sentence* if it has no free variables.
- A τ-formula ϕ is said to be in *Conjunctive Normal Form* (CNF) if its quantifier-free part is a conjunction of disjunctions of literals (i.e. atomic or negated atomic formulas). It is in *negation normal form* if its

quantifier-free part is a conjunction of conjunctions of literals, that is, it is in CNF and each disjunction of literals is negated.

- A τ-formula is said to be *positive* if its quantifier-free part contains no negations, and *negative* if its quantifier-free part is $\neg\psi$ for some positive formula ψ.
- A τ-formula ϕ is said to be *universal* if it is of the form $\forall\bar{x}\phi(\bar{x})$, where ϕ is quantifier-free, and *existential* if it is of the form $\exists\bar{x}\phi(\bar{x})$, where ϕ is quantifier-free.

We follow the Tarskian definition of truth in a structure for first-order logic. See [4] for an explicit definition. Of more importance for the purposes of this report is the notation that will be used.

Let τ be a signature, Γ a τ-structure and ϕ a τ-sentence. If ϕ is true in Γ we write $\Gamma \models \phi$. In this case, we may also say that Γ is a *model* for ϕ. If ϕ has free variables amongst the variables in $\bar{x} = (x_1, \ldots, x_n)$ we indicate this by writing $\phi(\bar{x})$. If $\bar{a} = (a_1, \ldots, a_n)$ are elements of $\mathrm{dom}(\Gamma)$ we write $\phi(\bar{a})$ for the τ-formula ψ given by $\phi(\bar{x})$ with x_i replaced by a_i for each $i \leq n$.

Of importance is the notion of signature *expansion* and *reduction*. Let $\tau^- \subseteq \tau^+$ be signatures and let Γ be a τ^+-structure. We can turn Γ into a τ^--structure Γ' by simply "forgetting" the symbols in $\tau^+ \setminus \tau^-$. Of course, no elements are removed from Γ, but some elements may no longer be named by constant symbols of τ^-. In this notation we say that Γ' is the τ^--*reduct* of Γ. If Γ is a τ^+-structure and Δ is its τ^--reduct we call Γ the τ^+-*expansion* of Δ. Note that a τ^--structure may have many different τ^+-expansions.

Let τ and τ' be relational signatures and let Γ, Γ' be τ and τ' structures, respectively, with $\mathrm{dom}(\Gamma) = \mathrm{dom}(\Gamma')$. We say that Γ' is *definable* in Γ if for each n-ary relation symbol R' of τ' there exists a τ-formula ϕ in n free variables such that:

$$R^{\Gamma'} = \{\bar{a} \in \Gamma^n \mid \Gamma \models \phi(\bar{a})\},$$

that is, if every relation on Γ' can be defined from Γ using τ-formulas.

A stronger notion than definability is that of interpretability. We follow here the definition of [7]. Let τ be a relational signature and Γ a τ-structure. Let τ' be another relational signature and Γ' be a τ'-structure. We say that Γ' is *interpretable* in Γ if there exist:

- A natural number m, called the *dimension* of the interpretation;
- A τ-formula $\delta(x_1, \ldots, x_m)$, called the *domain formula*;
- For each n-ary relation R in τ' a τ-formula $\phi_R(\bar{x}_1, \ldots, \bar{x}_n)$, called the *defining formula*, where $\bar{x}$ are m-tuples of distinct variables;
- A surjective map $f : \{\bar{a} \in \mathrm{dom}(\Gamma)^m \mid \Gamma \models \delta(\bar{a})\} \to \mathrm{dom}(\Gamma')$, called the *coordinate map*;

such that for each n-ary relation symbols R in τ' and all $\bar{a}_i \in \{\bar{a} \in \mathrm{dom}(\Gamma)^m \mid \Gamma \models \delta(\bar{a})\}$ we have that:

$$\Gamma' \models R(f(\bar{a}_i), \ldots, f(\bar{a_n})) \text{ if, and only if } \Gamma \models \phi_R(\bar{a}_1, \ldots, \bar{a}_n).$$

2.1.2 Theories and Models

Let τ be a fixed signature. A τ-*theory* is just a set of τ-sentences. We say that Σ is *consistent* if it has a model, i.e. if there exists a τ-structure Γ such that $\Gamma \models \phi$ for all $\phi \in \Sigma$. We say that Σ is *complete* if it is consistent and for any τ-sentence ϕ we have that $\Sigma \models \phi$ or $\Sigma \models \neg\phi$.

We say a τ-theory that Σ *entails* a τ-formula ϕ (or that ϕ is a *logical consequence* of Σ) if every model of Σ is a model of ϕ. Soundness and completeness of first-order logic allow us to overload notation and write $\Sigma \models \phi$ in this case. Note that if Σ is not consistent, then $\Sigma \models \phi$ for any τ-sentence ϕ, trivially.

Let $\mathcal{C}$ be a class of τ-structures. We say that $\mathcal{C}$ is τ-*definable* if $\mathcal{C} = \{\Gamma \mid \Gamma \models \phi\}$, where ϕ is a τ-sentence. We say that a τ-theory Σ *axiomatises* a class of τ-structures $\mathcal{C}$ if $\mathcal{C}$ is the class of all τ-structures that are models of Σ.

Given a τ-theory Σ, the class of structures $\mathcal{C}$ that it axiomatises is completely determined by Σ. We write $\mathcal{C} = \mathrm{Mod}(\Sigma)$, to mean that Σ axiomatises $\mathcal{C}$. Let Σ and T be τ-theories. We say that Σ axiomatises T if $\mathrm{Mod}(\Sigma) = \mathrm{Mod}(T)$.

Conversely, given a τ-structure Γ we define the *(first-order) theory* of Γ, denoted $\mathrm{Th}(\Gamma)$, to be the set of all first-order sentences that are true in Γ.

If Γ is a τ-structure and Σ is a first-order theory then we say that Σ axiomatises Γ if the first-order sentences that are true in Γ are exactly those that are true in every model of Σ, that is, if $\mathrm{Th}(\Gamma) = \mathrm{Mod}(\Sigma)$.

Definition 2.1.2. Let Σ be a τ-theory and $\phi(\bar{x})$, $\psi(\bar{x})$ be τ-formulas. Then $\phi(\bar{x})$ is *equivalent to* $\psi(\bar{x})$ *modulo* Σ if for every model Γ of Σ and every sequence $\bar{a}$ of elements of Γ we have that $\Gamma \vDash \phi(\bar{a})$ if, and only if $\Gamma \vDash \psi(\bar{a})$.

Equivalently, we have that $\phi(\bar{x})$ and $\psi(\bar{x})$ are equivalent modulo Σ if $\Sigma \vDash \forall \bar{x}(\phi(\bar{x}) \leftrightarrow \psi(\bar{x}))$. We say that $\psi(\bar{x})$ and $\psi(\bar{x})$ are *logically equivalent* if they are equivalent modulo the empty theory, i.e. if they are equivalent in every structure.

2.1.3 Morphisms and Substructures

Of equal importance to the objects that we will be discussing are the ways in which said objects relate to each other. In particular, CSPs, in the eyes of Model Theory, are *"homomorphism problems"*. We thus introduce these notions carefully.

Homomorphisms and Embedding

Definition 2.1.3. Let τ be fixed signature with no function symbols and Γ, Δ be τ-structures. A *homomorphism* from Γ to Δ is a (set-theoretic) function $f : \mathrm{dom}(\Gamma) \to \mathrm{dom}(\Delta)$ such that:

- For every relation symbol R_i of arity n_i in τ and every $(a_1, \ldots, a_{n_i}) \in \mathrm{dom}(\Gamma)^{n_i}$, if $(a_1, \ldots, a_{n_i}) \in R_i^\Gamma$ then $(f(a_1) \ldots, f(a_{n_i})) \in R_i^\Delta$.
- For every constant symbol c_k of τ we have $f(c_k^\Gamma) = c_k^\Delta$

We write $f : \Gamma \to \Delta$.

We are also interested in a special kind of homomorphism:

Definition 2.1.4. An *embedding* $f : \Gamma \to \Delta$ is an injective homomorphism such that for every relation symbol R_i of arity n_i in τ and every $(a_1, \ldots, a_{n_i}) \in \mathrm{dom}(\Gamma)^{n_i}$ we have $(a_1, \ldots, a_{n_i}) \in R_i^\Gamma$ if, and only if $(f(a_1) \ldots, f(a_{n_i})) \in R_i^\Delta$.

An *isomorphism* is a surjective embedding. If Γ and Δ are isomorphic τ-structures we write $\Gamma \cong \Delta$. An *endomorphism* is a homomorphism $f : \Gamma \to \Gamma$. An *automorphism* is an endomorphism that is an embedding.

It follows immediately from the definitions that the automorphisms of a τ-structure Γ form a group under composition. We denote this by $\mathrm{Aut}(\Gamma)$. In Section 2.1.12 we discuss how model theoretic properties of Γ translate to group theoretic properties of $\mathrm{Aut}(\Gamma)$.

Substructures

A very core idea in model theory is finding structures that "sit" inside other structures.

Definition 2.1.5. Let τ be a signature and Γ, Δ be τ-structures such that $\mathrm{dom}(\Gamma) \subseteq \mathrm{dom}(\Delta)$. Let $\iota : \mathrm{dom}(\Gamma) \to \mathrm{dom}(\Delta)$ be the inclusion map. If ι is an embedding then we say that Γ is an *induced substructure* or just *substructure* of Δ. If ι is just a homomorphism then we say that Γ is a *weak substructure* of Δ.

Since, in graph theory the notion of a subgraph H of a graph G is that of a weak substructure of G and the terminology can sometimes become confusing, we make it as explicit as possible.

In the notation of Definition 2.1.5 above, we have that Γ is an induced substructure of Δ if, and only if:

- For each relation symbol R_i of arity n_i in τ we have that $R_i^\Gamma = R_i^\Delta \cap \mathrm{dom}(\Gamma)^{n_i}$.
- For each constant symbol c_k in τ we have that $c_k^\Gamma = c_k^\Delta$.

and that Γ is a weak substructure of Δ if, and only if:

- For each relation symbol R_i of arity n_i in τ we have that $R_i^\Gamma \subseteq R_i^\Delta \cap \mathrm{dom}(\Gamma)^{n_i}$.
- For each constant symbol c_k in τ we have that $c_k^\Gamma = c_k^\Delta$.

Indeed, not every set of elements of a structure forms an induced substructure, but if we have a finite relational signature then every subset of a structure forms a weak substructure (such substructures of relational signatures can be expanded to induced substructures by adding relations between elements of their domain).

Note here, that since we are interested in relational signatures, we do not need to discuss finitely generated substructures. In fact, any finitely generated substructure is necessarily finite, in this case. If τ has countably many constant symbols then finitely generated substructures are no longer finite, but they are given precisely by the finite generating set and the constant symbols.

To avoid confusion, when discussing finitely generated substructures, letting $\bar{a}$ be a finite sequence of elements of Γ we write $\langle \bar{a} \rangle_\Gamma$ to mean the substructure of Γ generated by $\bar{a}$.

The following definition (deviating slightly from the notation of [8]) gives us an important class of structures:

Definition 2.1.6. Let τ be relational signature and $\mathcal{N}$ a class of finite τ-structures. We define the class of *strongly* $\mathbb{N}$*-free structures*, written Forb($\mathcal{N}$) to be the class of all finite τ-structures that do not contain any structures from $\mathcal{N}$ as induced substructures.

For a relational signature τ and τ-structures Γ_1 and Γ_2 we define the *disjoint union* of Γ_1 and Γ_2, written $\Gamma_1 \sqcup \Gamma_2$, with domain consisting of a union of disjoint isomorphic copies of Γ_1 and Γ_2 and relations $R^{\Gamma_1 \sqcup \Gamma_2} = R^{\Gamma_1} \cup R^{\Gamma_2}$, for each R in τ.

2.1.4 Preserving Formulas

The notion of morphisms preserving first-order formulas plays a crucial role in the study of CSPs, and to this end, we give a more detailed presentation.

Definition 2.1.7. Let τ be a signature, Γ and Δ be τ-structures, $f : \Gamma \to \Delta$ a homomorphism and $\phi(\bar{x})$ be a τ-formula. We say that f *preserves* $\phi(\bar{x})$ if for every sequence $\bar{a}$ of elements of Γ we have that $\Gamma \vDash \phi(\bar{a})$ implies $\Delta \vDash \phi(f(\bar{a}))$. We say that a formula $\phi(\bar{x})$ is *preserved in substructures* if whenever $\Gamma \subseteq \Delta$ are τ-structures then $\Gamma \vDash \phi(\bar{a})$ for some sequence of elements $\bar{a}$ from Γ then $\Gamma \vDash \phi(\bar{a})$.

The proof of the following is immediate from the definitions:

Theorem 2.1.1. *Let τ be a signature, Γ, Δ be τ-structures and f a (set-theoretic) map $f : dom(\Gamma) \to dom(\Delta)$.*
(a) We have that f is a homomorphism if, and only if for every atomic τ-formula $\phi(\bar{x})$ and tuple $\bar{a}$ of Γ we have that $\Gamma \vDash \phi(\bar{a})$ implies that $\Delta \vDash \phi(f(\bar{a}))$.
(b) We have that f is an embedding if, and only if for every atomic τ-formula $\phi(\bar{x})$ and tuple $\bar{a}$ of Γ we have that $\Gamma \vDash \phi(\bar{a})$ if, and only if $\Delta \vDash \phi(f(\bar{a}))$.
In particular a homomorphism $f : dom(\Gamma) \to dom(\Delta)$ is an embedding if, and only if for every literal $\phi(\bar{x})$ of τ and sequence $\bar{a}$ from Γ we have that if $\Gamma \vDash \phi(\bar{a})$ then $\Delta \vDash \phi(\bar{b})$, for some $\bar{b}$ from Δ

So homomorphisms preserve atomic formulas and a homomorphism is an embedding if, and only if it preserves literals. In fact, it is easy to see that this implies that embeddings preserve existential formulas. In fact, we can say something stronger:

Theorem 2.1.2 (Loś-Tarski Theorem, Corollary 5.4.5 of Theorem 5.4.4 in [4])**.** *Let Σ be a first-order τ-theory. and ϕ a first-order τ-sentence. Then the following are equivalent:*
- *ϕ is preserved by embeddings between models of Σ.*
- *ϕ is equivalent to an existential formula.*

Moreover, we have:

Theorem 2.1.3 (Homomorphism Preservation Theorem, Theorem 1.3 in [9])**.** *Let τ be a signature and let $f : \Gamma \to \Delta$ a homomorphism of τ-structures. Then f preserves $\phi(\bar{x})$ if, and only if ϕ is equivalent to an existential positive formula.*

Note that the second theorem is true for *all* τ-structures, finite or infinite. A proof of this fact can be found in [9]. We stated 2.1.2 in terms of models of theories because as noted in [10], it is not true for finite structures.

Theorem 2.1.3 gives an important connection between classical Model Theory and Finite Model Theory and we will make good use of it in the next chapter.

2.1.5 Elementary Equivalence and Elementary Embeddings

Unlike other areas of pure mathematics, where *morphisms* serve as the primary tool of relating structures, Model Theory, as a branch of mathematical logic, has other, more logical methods of relating structures. We present two in this report. One is below and another other in Appendix A.

Definition 2.1.8. Let τ be a signature and Γ, Δ be τ-structures. We say that Γ and Δ are *elementarily equivalent* if for every first-order τ-sentence ϕ we have $\Gamma \vDash \phi$ if, and only if $\Delta \vDash \phi$. We write $\Gamma \equiv \Delta$.

Note that for τ-structures Γ, Δ we have that $\Gamma \cong \Delta$ implies $\Gamma \equiv \Delta$, but not vice versa. The converse does hold, for example, when Γ and Δ are finite. Note that a τ-theory Σ is complete if, and only if it is consistent and any two models of Σ are elementarily equivalent.

Let τ be a signature and $f : \Gamma \to \Delta$ a homomorphism of τ-structures. Let Φ be a class of first-order τ-formulas. We call f a *Φ-map* if f preserves all the formulas in Φ. If Φ be the class of all first-order formulas. and let f be a Φ-map. Then f is an *elementary embedding.*

Suppose that $f : \Gamma \to \Delta$ is an elementary embedding. Then for any first-order formula $\phi(\bar{x})$ we have that $\Gamma \vDash \phi(\bar{a})$ implies that $\Delta \vDash \phi(f(\bar{a}))$, in particular, this holds when $\phi(\bar{x})$ is a literal. Hence, elementary embeddings are embeddings. On the other hand, embeddings are just maps that preserve literals, so not all embeddings are necessarily elementary.

Definition 2.1.9. Let Γ, Δ be τ-structures. We call Δ an *elementary extension* of Γ, or to the same extent, Γ an *elementary substructure* of Δ, written $\Gamma \preccurlyeq \Delta$, if $\Gamma \subseteq \Delta$ and the inclusion map $\iota : \Gamma \to \Delta$ is an elementary embedding.

Note that if $\Gamma \preccurlyeq \Delta$ then $\Gamma \equiv \Delta$. On the other hand, it is not true in general that if $\Gamma \subseteq \Delta$ and $\Gamma \equiv \Delta$ then $\Gamma \preccurlyeq \Delta$, but we have a method to determine if $\Gamma \preccurlyeq \Delta$:

Theorem 2.1.4 (Tarski-Vaught Criterion, Theorem 2.5.1 in [4])**.** *Let Γ, Δ be τ-structures with $\Gamma \subseteq \Delta$. Then $\Gamma \preccurlyeq \Delta$ if, and only if for every τ-formula $\psi(\bar{x}, y)$ and all tuples $\bar{a}$ from Γ we have that if $\Delta \vDash \exists y(\psi(\bar{a}, y))$ then $\Delta \vDash \psi(\bar{a}, c)$ with $c \in dom(\Gamma)$.*

2.1.6 Categoricity

Recall the following, basic result of Model Theory:

Theorem 2.1.5 (Compactness Theorem, Theorem 5.1.1 in [4])**.** *Let τ be an arbitrary signature and Σ a τ-theory. Then Σ has a model if, and only if every finite subset of Σ has a model.*

Using Compactness (for the up direction) and Skolemisations (for the down direction) one can easily prove:

Theorem 2.1.6 (Löwenheim-Skolem Theorems, Corollaries 5.1.4 and 3.1.4 in [4], respectively)**.** *Let τ be an arbitrary signature and let λ be any infinite cardinal.*

- (Upwards). *Suppose that τ has cardinality at most λ. Let Γ be a τ-structure and suppose that $|dom(\Gamma)| \leq \lambda$. Then Γ has an elementary expansion of cardinality exactly λ.*
- (Downwards). *Suppose that Γ is a τ-structure, X a subset of $dom(\Gamma)$ and that $|\tau| + |X| \leq \lambda \leq |\Gamma|$. Then Γ has an elementary substructure $\Delta \preccurlyeq \Gamma$ such that $X \subseteq dom(\Delta)$.*

In general, we are interested in how models of a fixed theory are related. We have the following definition:

Definition 2.1.10. A theory Σ is *categorical* if Σ is consistent and all models of Σ are isomorphic.

It is clear that the only categorical first-order theories are complete theories of finite structures[1]. This follows immediately from the Löwenheim-Skolem Theorems, since any theory of an infinite structure will have a model of every cardinality. To make sense of the notion of categoricity in first-order logic we need to ask for something stronger:

[1]This is not the case for second-order theories. It is known that the Peano Axioms (in second-order logic with standard semantics) are categorical [11].

Definition 2.1.11. Let λ be a cardinal. A class of τ-structures $\mathcal{C}$ is called λ-*categorical* if all τ-structures in $\mathcal{C}$ of cardinality λ are isomorphic. A theory Σ is λ-*categorical* if the class of all its models is λ-categorical.

A structure Γ is λ-categorical if, and only if $\mathrm{Th}(\Gamma)$ is λ-categorical. Of particular importance, for our purposes, is the case of ω-categoricity, where ω is the first countable cardinal.

2.1.7 Some Group Theory

Recall that $\mathrm{Aut}(\Gamma)$ is the group of automorphisms of a structure Γ. We regard it as a subgroup of $\mathrm{Sym}(\mathrm{dom}(\Gamma))$. Several properties of Γ translate to properties of its automorphism group. To make this precise we give some definitions from group theory.

Let Ω be any set and let G be a subgroup of $\mathrm{Sym}(\Omega)$. Let $n \in \mathbb{N}$. Then G can be viewed as a group of permutations of Ω^n, where $g(a_1, \ldots, a_n) = (g(a_1), \ldots, g(a_n))$. In this light, we can talk about the orbits of elements of Ω^n. Of course, if $n \geq 2$ then G is definitely not transitive on Ω^n.

Definition 2.1.12. We say that G is *oligomorphic* on Ω if for every positive integer n the number of orbits of G on Ω^n is finite.

Connecting this with Model Theory we have:

Theorem 2.1.7. *Let τ be an arbitrary, countable signature and Γ a countable τ-structure. Then Aut(Γ) is oligomorphic if, and only if Γ is ω-categorical.*

Proof. This is one of the equivalent conditions appearing in Theorem 2.1.12. □

2.1.8 Fraïssé's Theorem

Up until this point we have given a quick overview of various technical terms, but in this and the next section, we slow down the exposition and offer a more detailed discussion.

In this section we give a complete proof of *Fraïssé's Theorem*, originally proved in [12]. The proof we give is based on [4].

Setup Throughout this section, τ is assumed to be a countable signature with no function symbols. Note that in the proof of Fraïssé's Theorem the only important assumption we make about τ is that it is countable. Avoiding function symbols simplifies the discussion, as we will not need to distinguish between finite and finitely generated substructures. Since the argument does not actually assume that τ is relational we highlight this by shift notation and writing $A, B, \ldots$ for τ-structures.

Definition 2.1.13. Let A be a τ-structure. The *age* of A is the class $\mathcal{C}$ of all finitely generated τ-structures that are isomorphic to a substructure of A, taken up to isomorphism. We write $\mathcal{C} = \mathrm{Age}(A)$. We call $\mathcal{C}$ an *age* if it is the age of some τ-structure.

In other words, the age of A is the collection of all finitely generated substructures of A. Since this class is closed under isomorphisms, we can choose one representative for each isomorphism type. If our signature is countable, then there are countably many isomorphism types for each structure $B \in \mathrm{Age}(A)$ and working modulo isomorphism guarantees that $\mathrm{Age}(A)$ is countable.

We now define three very important properties that classes of τ-structures may exhibit:

Definition 2.1.14. Let $\mathcal{C}$ be a class of τ-structures. We define:

(i) The *Hereditary Property* (HP): If $A \in \mathcal{C}$ and $B \subseteq A$ is a finitely generated substructure then $B \in \mathcal{C}$.

(ii) The *Joint Embedding Property* (JEP) If $A, B \in \mathcal{C}$ then there exists some $C \in \mathcal{C}$ such that both A and B are can be embedded in C.

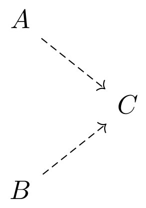


Figure 2.1: The Joint Embedding Property.

(iii) The *Amalgamation Property* (AP): If $A, B, C \in \mathcal{C}$ and $e : A \to B$, $f : A \to C$ are embeddings then there exists some $D \in \mathcal{C}$ and embeddings $g : B \to D$, $h : C \to D$ such that $g \circ e = h \circ f$.

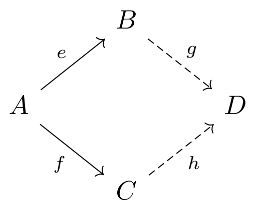


Figure 2.2: The Amalgamation Property.

Note that in general JEP is not a special case of AP unless, of course, $\mathcal{C}$ contains the empty structure. It should be clear that if $\mathcal{C}$ is an age, then $\mathcal{C}$ has JEP and HP.

Definition 2.1.15. Let A be a τ-structure. Then A is *ultrahomogeneous* (or just *homogeneous*) if every isomorphism between finitely generated substructures of A extends to an automorphism of A.

Theorem 2.1.9 says that for the age of A to have AP, A needs to be homogeneous.

Statement We can now state the Fraïssé's Theorem.

Theorem 2.1.8. *Let τ be a countable signature and $\mathcal{C}$ be a countable set of finitely generated τ-structures which is closed under isomorphisms and has HP, JEP and AP. Then there exists a countable τ-structure A such that A is ultrahomogeneous and $\mathcal{C}$ is the age of A. Moreover, A is unique, up to isomorphism.*

Definition 2.1.16. In the notation of Theorem 2.1.8 we say that A is the *Fraïssé Limit* of $\mathcal{C}$. We denote the Fraïssé limit of $\mathcal{C}$ by $\text{Flim}(\mathcal{C})$.

We will discuss several examples in later sections. In this section, we expand on Fraïssé's original example, *Dense Linear Orders* with no endpoints. The archetypal example of a dense linear order without endpoints is $(\mathbb{Q}, \leq)$. We will now use Theorem 2.1.8 to show that this is, in fact, the "only" example, in the eyes of first-order logic.

Example 2.1.1. Let $\mathcal{C}$ be the class of all finite linear orders. Then $\mathcal{C}$ has HP, AP and JEP. The fact that $\mathcal{C}$ has HP is obvious, as any finitely generated substructure of a finite linear order is a linear order. AP is also fairly clear, since for finite linear orders B, C, each of which contains a copy of some linear order A we can easily construct a linear order D making the diagram in Figure 2.2 commute. Note that this amalgamation is not free (see Definition 2.1.20), since we will need to introduce new relations. Finally, since the empty structure is in $\mathcal{C}$, JEP is, in this case an immediate consequence of AP. Now, Fraïssé's Theorem tells us that there exists a unique countable structure A which is ultrahomogeneous and has $\text{Age}(A) = \mathcal{C}$, that is, all structures in $\mathcal{C}$ can be embedded in A.

Consider $Q = (\mathbb{Q}, \leq)$. Clearly, we can identify any finite linear order with a substructure of Q and any finite substructure of Q is a finite linear order, so $\mathcal{C} = \text{Age}(Q)$. Moreover, if $A, B \subsetneq \mathbb{Q}$ are arbitrary finite subsets of $\mathbb{Q}$ and $f : A \to B$ is an isomorphism then we can extend this to an automorphism of Q. We do this by a back and forth argument, essentially constructing a back and forth system (see Example A.1.1). Instead of having to construct the map from "scratch", this time we will take the map $f_0 = f : \langle a_1, \ldots, a_k \rangle_Q \to \langle b_1, \ldots, b_k \rangle_Q$ as our starting point. The back and forth argument is given in a slightly more general setting in Example A.1.1. Since $\mathbb{Q}$ is countable we can, by induction, get an automorphism of Q, extending f. So Q is ultrahomogeneous. It follows that Q is isomorphic to the Fraïssé Limit of $\mathcal{C}$, so any countable dense linear order without endpoints is isomorphic to Q.

Of course, since first-order logic cannot "count" above the first limit cardinal, the observation that in the eyes of first-order logic Q is the only dense linear order with no endpoints makes sense, as any other dense linear order with no endpoints will clearly be elementarily equivalent to Q.

Proof The proof is split into two parts, existence and uniqueness. Uniqueness will be proved first, since we will use the an argument from the uniqueness proof in the existence proof.

As the proof is fairly long and the argument is somewhat technical, we start with a short intuitive summary of the main ideas in the proof:

- *Uniqueness*: The crucial assumption here is that A, where $\mathcal{C} = \text{Age}(A)$, is countable. We will essentially show that if B is any other countable homogeneous τ-structure whose age is $\mathcal{C}$ then A and B are back and forth equivalent. Then by we will use an argument like the one in the proof of Theorem A.1.2, to show that A and B are isomorphic.
- *Existence*: In the proof we will make use of JEP and AP to start with an arbitrary structure in $\mathcal{C}$ and inductively extend it, constructing a chain of structures in $\mathcal{C}$, until we get a structure whose age is $\mathcal{C}$. Ultra-homogeneity will be a consequence of AP.

Now that the main ideas in the proof have been discussed, to some extent, we will present the argument (based on [4]) in its full formal technicality.

Uniqueness First, we need one more definition:

Definition 2.1.17. Let A be a τ-structure. Then A is *weakly homogeneous* if for every pair of finitely generated substructures $B, C \subseteq A$ such that $B \subseteq C$ and for every embedding $f : B \to A$ there exists an embedding $g : C \to A$ that extends f.

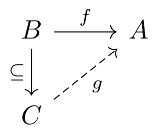


Figure 2.3: Weak Homogeneity.

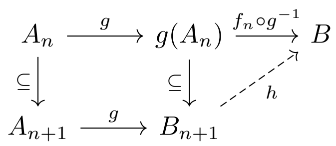
We will see that in the case of countable structures this definition amounts to nothing new. Of course, it is immediate that if a structure A is ultrahomogeneous it is weakly homogeneous, as the names would suggest, but we will in fact show that for countable structures the converse is also true.

Lemma 2.1.1. *Let A and B be countable τ-structures such that Age$(A) \subseteq$ Age(B) and B is weakly homogeneous. Then there exists an embedding $f : A \to B$. In fact, any embedding from a finitely generated substructure of A into B can be extended to an embedding of A into B.*

Proof. Let $A_0 \subseteq A$ be a finitely generated substructure of A. Since $A_0 \in \text{Age}(B)$ we can find an embedding $f_0 : A_0 \to B$. We will extend this to an embedding $f : A \to B$ inductively.

First, since A is countable we can write $A = \cup_{n\in\mathbb{N}} A_n$ where $(A_i : i \in \mathbb{N})$ is a chain of finitely generated substructures of A, i.e. $A_i \subseteq A_{i+1}$, for all $i \in \mathbb{N}$, starting with the fixed A_0, from above.

Now, we want to build an increasing chain of embeddings $f_n : A_n \to B$. Suppose we have $f_n : A_n \to B$. Then, since $A_n \in \text{Age}(A) \subseteq \text{Age}(B)$ there exists some finitely generated substructure $B_{n+1} \subseteq B$ which is isomorphic to A_{n+1}. Let $g : A_{n+1} \to B_{n+1}$ be the isomorphism and consider $f_n \circ g^{-1}$. This map gives an embedding of $g(A_n) \subseteq B_{n+1}$ into B, and since B is weakly homogeneous there exists an embedding $h : B_{n+1} \to B$ extending $f_n \circ g^{-1}$. We have:



Then letting $f_{n+1} = h \circ g$ gives us an embedding which extends f_n. The result follows by induction and taking $f = \bigcup_{n\in\mathbb{N}}(f_n)$. □

Definition 2.1.18. Let B be a countable τ-structure and let $\mathcal{C} = \mathrm{Age}(B)$. We say that B is *universal* for $\mathcal{C}$ if for every countable structure A such that $\mathrm{Age}(A) \subseteq \mathcal{C}$ there exists an embedding $f : A \to B$.

With this terminology, by the lemma above, weakly homogeneous substructures are universal for their age. Now, suppose that we have two weakly homogeneous substructures of the same age. Then by repeating the argument above, since both are universal for their age they will be isomorphic. This is the following lemma:

Lemma 2.1.2. *Let A and B be countable, weakly homogeneous τ-structures of the same age. Then $A \cong B$. In fact if $C \subseteq A$ is a finitely generated substructure of A and $f : C \to B$ an embedding then f can be extended to an isomorphism from A to B.*

Proof. Just like the previous proof we will start by expressing A and B as union of a chain of countably many finitely generated substructures. So let $(A_n : n \in \mathbb{N})$ and $(B_n : n \in \mathbb{N})$ be such that $A = \bigcup_{n\in\mathbb{N}} A_n$ and $B = \bigcup_{n\in\mathbb{N}} B_n$. Using the proof of the previous lemma we define a chain of isomorphisms between finitely generated substructures of A and B, $(f_n : n \in \mathbb{N})$, such that for each $n \in \mathbb{N}$ we have that $A_n \subseteq \mathrm{dom}(f_{2n})$ and $B_n \subseteq \mathrm{im}(f_{2n+1})$. Then the union $f = \cup_{n\in\mathbb{N}} f_n$ is an isomorphism from A to B. □

Note that it if A is a countable, weakly homogeneous structure then we can use the proof of the lemma above taking $B = A$ to extend any isomorphism of between finitely generated substructures of A to an automorphism of A, so as claimed, if A is countable then it is ultrahomogeneous if, and only if it is weakly homogeneous.

But if A is countable and ultrahomogeneous then it is the Fraïssé limit for its age and by Lemma 2.1.2 it is unique, up to isomorphism as required. Note that the proofs in this and the next section take full advantage of the countability of A and this is for good reason. The results fail when A is uncountable.

This connects very nicely with some notions discussed in Appendix A. In particular:

Proposition 2.1.3. *Suppose that A and B are arbitrary weakly homogeneous τ-structures of the same age. The A and B are back and forth equivalent.*

Proof. Using the proof of the previous lemma we can construct countably many partial isomorphisms between A and B. These partial isomorphisms give us a back and forth system between A and B, so the two are back and forth equivalent. □

Existence Now that we have showed uniqueness, we are on the right track, so we move to existence.

Lemma 2.1.4. *Let $\mathcal{C}$ be a countable class of finitely generated τ-structures and assume that $\mathcal{C}$ has HP, JEP and AP. Then there exists a countable, ultrahomogeneous τ-structure A such that Age$(A) = \mathcal{C}$*

Proof. Note, first of all, that if $\mathcal{D}$ is any class of finitely generated τ-structures and $(A_i : i \leq \alpha)$ a chain of τ-structures such that for each $i \leq \alpha$ we have that $\mathrm{Age}(A_i) \subseteq \mathcal{D}$ then $\mathrm{Age}(\bigcup_{i\leq\alpha} A_i) \subseteq \mathcal{D}$. In particular, if $\mathrm{Age}(A_i) = \mathcal{D}$ for all i then $\mathrm{Age}(\bigcup_{i\leq\alpha} A_i) = \mathcal{D}$.

Now, suppose that $\mathcal{C}$ is countable and has HP, JEP and AP. Since we are only interested in the *isomorphism types* of structures in $\mathcal{C}$ (of course, since τ is countable, we have at most countably many isomorphism types of finitely generated τ-structures), we can assume without loss that $\mathcal{C}$ is closed under taking isomorphic copies of its elements. This time we will construct a chain of structures $(A_n : n \in \mathbb{N})$ of structures in $\mathcal{C}$ with the following property:

Property. *For any $M, N \in \mathcal{C}$ with $M \subseteq N$ if there is an embedding $f : M \to A_i$ for some i then for some $j \geq i$ we can find an embedding $g : N \to A_j$ which extends f. That is, we have:*

$$\begin{array}{ccc} M & \xrightarrow{\subseteq} & N \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\ A_i & \xrightarrow{\subseteq} & A_j \end{array}$$

Suppose that such a chain exists and take $A = \bigcup_{n\in\mathbb{N}} A_n$. Since each A_i is already in $\mathcal{C}$ it follows that all finitely generated substructures of A are in $\mathcal{C}$, as $\mathcal{C}$ has HP. So by the previous discussion $\mathrm{Age}(A) \subseteq \mathcal{C}$. We have to show that $\mathrm{Age}(A) = \mathcal{C}$ and that A is ultrahomogeneous.

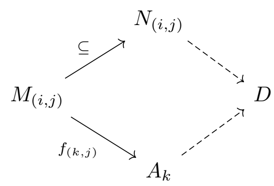
1. $\mathrm{Age}(A) = \mathcal{C}$. Let $M \in \mathcal{C}$. Since both M and A_0 are in $\mathcal{C}$ and $\mathcal{C}$ has JEP we can find a structure N in $\mathcal{C}$ such that $M \subseteq N$ and both M and A_0 can be embedded in N. Now, consider the identity map on A_0. By assumption, we can extend this to an embedding of N to some A_n, in particular, N is isomorphic to a finitely generated

substructure of A, so $N \in \text{Age}(A)$, but since $M \subseteq N$ is a finitely generated substructure and ages have HP we have that $M \in \text{Age}(A)$, so $\mathcal{C} = \text{Age}(A)$.

2. *A is ultrahomogeneous.* We will show, in fact that A is weakly homogeneous. Suppose that $B, C \subseteq A$ are finitely generated substructures of A and $B \subseteq C$. Then for any embedding $f : B \to A$ we can find some A_i such that $\text{im}(f) \subseteq A_i$ and by construction we can find some $j \geq i$ and embedding $g : C \to A_j$ extending f. Viewing g as an embedding $g : C \to A$ shows that A is weakly homogeneous and hence, since A is countable it is ultrahomogeneous, by the discussion after Lemma 2.1.2.

Now it remains to actually construct the chain $(A_n : n \in \mathbb{N})$. We will do so by a clever combinatorial trick. Recall that the property we want to show is that whenever $M \subseteq N \in \mathcal{C}$ and $f : M \to A_i$ is an embedding there is some $j \geq i$ such that $g : N \to A_i$ is an embedding extending f. We start by taking $\mathcal{P}$ to be the set of all isomorphism types of pairs (M, N) such that $M, N \in \mathcal{C}$ and $M \subseteq N$. Since $\mathcal{C}$ is at most countable we can choose $\mathbf{P}$ is countable. Then, we let $\pi : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be any bijection such that $\pi(i, j) \geq i$ for all pairs $(i, j) \in \mathbb{N} \times \mathbb{N}$. Let $A_0 \in \mathcal{C}$ be arbitrary. Suppose that we have A_k, we will use AP to define $A_{k+1} \in \mathcal{C}$.

This is where the property of the π we defined comes into place. At each stage, when A_k has been defined we list as $((f_{(k,j)}, M_{(k,j)}, N_{(k,j)}) : j \in \mathbb{N})$ all the triples such that $(M_{(k,j)}, N_{(k,j)}) \in \mathcal{P}$ (i.e. $M_{(k,j)} \subseteq N_{(k,j)}$) and $f_{(k,j)} : M_{(k,j)} \to A_k$ is an embedding. Note that by construction of π there exists a unique pair $(i, j) \in \mathbb{N} \times \mathbb{N}$ such that $\pi(i, j) = k$ and $i \leq k$. Then, we go back to the previous list we had, for i, this time and look at $(f_{(i,j)}, M_{(i,j)}, N_{(i,j)})$. We have that $f_{(i,j)} : M_{(i,j)} \to A_i$ is an embedding and since $\mathcal{C}$ has AP there exists some $D \in \mathcal{C}$ and an embedding from $N_{(i,j)}$ to D that extends $f_{(i,j)}$, that is, we can make the following diagram commute:

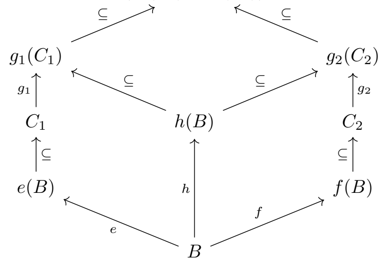


Let $A_{k+1} = D$. Then we have completed the (i, j)-th task. We proceed, inductively, in this manner and since π is a bijection all tasks will be completed, using AP, at each stage. Then, if $M, N \in \mathcal{C}$ are such that $M \subseteq N$ and $f : M \to A_i$ is an embedding, for some $i \in \mathbb{N}$, then (M, N) is a pair that is isomorphic to one of the pairs $(M_{(i,j)}, N_{(i,j)})$. Let $k = \pi(i, j)$. Then $k \geq i$ and by the construction above we can find an embedding $g : N \to A_k$ extending f. Therefore the chain has the required property. $\square$

This lemma concludes the proof of Fraïssé's Theorem. We will now show a "converse" to Theorem 2.1.8:

Theorem 2.1.9. *Let τ be a countable signature and A a countable ultrahomogeneous τ-structure. Then Age(A) has HP, JEP and AP.*

Proof. As we noted all ages have HP and JEP. The point is, that when A is ultrahomogeneous, $\text{Age}(A)$ has AP, as well. To show this, suppose that we have finitely generated substructures $B, C_1, C_2 \in \text{Age}(A)$ and embeddings $e : B \to C_1$ and $f : B \to C_2$. Let $h : B \to A$ be an embedding of B into a finitely generated substructure $h(B) \subseteq A$. Then $h \circ e^{-1} : e(B) \to A$ is an embedding of $e(B)$ into A. Since A is ultrahomogeneous, and hence weakly homogeneous, and $e(B) \subseteq A$, we can extend this to an embedding $g_1 : C_1 \to A$. We repeat the same argument to extend $h \circ f^{-1} : f(B) \to A$ to an embedding $g_2 : C_2 \to A$. This gives us finitely generated substructures $g_1(C_1), g_2(C_2) \subseteq A$ that contain the "same copy" of $h(B)$. In particular, the following diagram commutes:

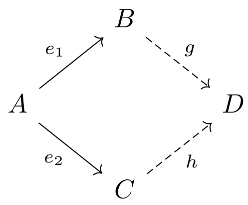


And $\langle g_1(C_1) \cup g_2(C_2)\rangle$ (the finitely generated substructure of A generated by $g_1(C_1) \cup g_2(C_2)$) together with the outer embeddings in the diagram above give us the required result. □

One of the main results we are building towards in this section is Theorem 2.1.12. This result will allow us to prove that when τ is finite and relational then any homogeneous structure (and in particular, any Fraïssé Limit) is ω-categorical. To state Theorem 2.1.12 completely we will first need to develop some terminology for types. Before we do so, we will take a fairly long detour to examine properties that the *age* of a countable structure may satisfy and to use Fraïssé's Theorem to construct some interesting structures.

2.1.9 Strong and Free Amalgamation

The two properties we will discuss are variants of AP. Let $\mathcal{C}$ be a class of τ-structures and suppose that $\mathcal{C}$ has AP. As in Definition 2.1.14 we have the following commutative diagram for $A, B, C \in \mathcal{C}$:



The two properties we are going to define are the *Strong Amalgamation Property* and the *Free Amalgamation Property*. In the notation above, let $f = g \circ e_1 = h \circ e_2$. Then:

Definition 2.1.19. We say that $\mathcal{C}$ has the *strong amalgamation property* if we also have that for all $b \in B$ and $c \in C$ such that $f(b) = f(c)$ there exists some $a \in A$ such that $e_1(a) = b$ and $e_2(a) = c$

The point is that we can "amalgamate" any two structures $B, C \in \mathcal{C}$ over their common substructure A without identifications outside A. This means that we never end up identifying elements in $B \setminus A$ with elements in $C \setminus A$. Of course, we may need to add relations between the elements of $B \setminus A$ and $C \setminus A$, but we can essentially think of this as "gluing" the structures over A.

Example 2.1.2. Let τ be the a signature for graphs. Then:
1. If $\mathcal{C}$ is the class of all finite graphs, then $\mathcal{C}$ has the strong amalgamation property.
2. Let $n > 2$ and denote by K_n the complete (directed or undirected) graph on n vertices. Then the class of K_n-free graphs has the strong amalgamation property.
3. All examples in Section 2.1.10 have the strong amalgamation property. For example if we let $\mathcal{C}$ be the class of tournaments (i.e. directed graphs where each node is connected to each other node just once) that forbids the structures in 2.4, then as we will see this has the strong amalgamation property.

We mention these classes here, since we will examine them in the next section in some detail.
On the other hand, the class of finite fields (in the natural signature for fields) of fixed characteristic p, which can easily be seen to be an amalgamation class, does not have the strong amalgamation property.

The idea behind free amalgamation is that we may be able to amalgamate two structures over a given structure without adding any new relations at all. More precisely, keeping the notation from above:

Definition 2.1.20. Let e_1 and e_2 be inclusions. The *free amalgam* of B and C, over A is the structure D, where $\mathrm{dom}(D) = \mathrm{dom}(B) \cup \mathrm{dom}(C)$ and for each n-ary relation R in the signature τ we have that $R^D = R^B \cup R^C$.
A class $\mathcal{C}$ of τ-structures has the *free amalgamation property* if for all $A, B, C \in \mathcal{C}$ such that $A \subseteq B$ and $A \subseteq C$ the free amalgam of B and C over A is in $\mathcal{C}$.

Clearly, if a class $\mathcal{C}$ has the free amalgamation property then it has the strong amalgamation property.

Example 2.1.3. As we noted when discussing the Fraïssé Limit finite linear orders the amalgamation there is not free. On the other hand, if we discuss the set of all K_n-free graphs, where K_n is the complete graph on n vertices this class very much has the free amalgamation property (its Fraïssé Limit is the *universal K_n-free graph*). An example of similar flavour is 3 from Example 2.1.2. This has the strong amalgamation property, but the amalgamation is clearly not free.

Note that we will prove in Section 2.1.12 that if τ is a finite, relational signature then any homogeneous structure is ω-categorical (Corollary 2.1.8). In our discussion, in the next section, all structures will thus be ω-categorical.

2.1.10 Countable Homogeneous Digraphs

In this section, we will apply Fraïssé's techniques to discuss homogeneous (see Definition 2.1.15) graphs. That is, homogeneous structures in a signature τ with a single binary relation E.

If we assume E to be irreflexive, then we will call τ-structures *digraphs* (that is, directed graphs). In this case we may write $u \to v$ or even $u < v$ instead of $E(u, v)$. If, on the other hand we assume that E is reflexive then we will call τ-structures *graphs*. In this section we will be careful with the terminology used, but in later sections we will be a bit more lax and assume that it is clear from the context.

Let G be a graph. We define the *complement* of G, G^c to be the graph on the same domain as G such that, for any $v, g \in \mathrm{dom}(G^c)$ we have that $G^c \models E(v, u)$ if, and only if $G \models \neg E(v, u)$. We may (informally) refer to edges in the complement as "negative edges".

Homogeneous Graphs The first example we will give is the *random graph*. Let $\mathcal{C}$ be the class of all finite graphs. We note that $\mathcal{C}$ is countable (modulo isomorphism) and is clearly an amalgamation class, since we can just use free amalgamation. Hence there exists a countable, homogeneous structure $\mathcal{R}$ such that $\mathrm{Age}(\mathcal{R}) = \mathcal{C}$. That is, there exists a countable, homogeneous graph $\mathcal{R}$ such that if G is any finite graph, there exists an embedding $f : G \to \mathcal{R}$. Fraïssé tells us, in addition to this, that $\mathcal{R}$ is unique, up to isomorphism. It is easy enough to see this without Fraïssé. In particular, what we can show using a back and forth argument and the fact that $\mathcal{R}$ is countable is that $\mathrm{Th}(\mathcal{R})$ (and so, $\mathcal{R}$) is ω-categorical and thus that the random graph is unique, up to isomorphism.

Let $n \geq 3$. As we mentioned in Example 2.1.2, the class $\mathcal{C}$ of K_n-free graphs has the strong amalgamation property (and in fact it has the free amalgamation property). It follows that there exists a unique countable homogeneous graph $\mathcal{R}_n$, such that $\mathrm{Age}(\mathcal{R}_n) = \mathcal{C}$. We call $\mathcal{R}_n$ the *universal K_n-free* graph.

In particular, Woodrow and Lachlan proved in [13] that if G is any infinite countable homogeneous graph then either G or G^c is isomorphic to one of $\mathcal{R}$, $\mathcal{R}_n$ for some $n \geq 3$ or $\bigsqcup_{i=1}^{\infty} K_n$, for a fixed $n \geq 3$. This gives us a complete classification of infinite countable homogeneous graphs. We will not strictly need this for the discussion on Constraint Satisfaction, but it serves as a good motivation for the results in the next paragraph, which are actually very important.

Homogeneous Tournaments Having given a brief overview of the homogeneous graphs we will now discuss homogeneous digraphs. We will start with *tournaments*, a special case of digraphs, which have the property that for any two nodes v, u there exists a single edge connecting them, that is, G is a tournament if it is a model of the sentence:

$$\forall v \forall u((E(u, v) \vee E(v, u)) \wedge (E(u, v) \to \neg E(v, u))).$$

In [14], Lachlan proved a deep result, which is, that up to isomorphism, there exist three infinite countable homogeneous tournaments. These are precisely:

1. The *Universal Tournament*, which we will denote as T^∞.
2. The *Dense Linear Order* with no endpoints, which we will denote as $(\mathbb{Q}, <)$.
3. The *Dense Local Order* which we will denote as $S(2)$.

We will not prove Lachlan's result here. Instead, we will show that all three structures above are indeed homogeneous and we will describe $S(2)$ in three not obviously equivalent ways.

We start with T^∞. Let $\mathcal{T}$ be the class of all finite tournaments. Clearly $\mathcal{T}$ is an amalgamation class (it has strong amalgamation, but not free amalgamation). By definition $T^\infty = \mathrm{Flim}(\mathcal{T})$. Since we have already discussed $(\mathbb{Q}, <)$ we will now work with $S(2)$.

The characterisation is taken from [7], but the final part of the proof we give (Lemma 2.1.7) is new.

By definition, $S(2)$ is a countable, dense set of points on the unit circle, without antipodal points, where two points a, b are linked by an arc if, and only if the directed line from a to b passes the origin on the right side.

Define a structure Γ as a partition of $\mathbb{Q}$ into two dense subsets Q_1 and Q_2 where $a \to b$ if, and only if a and b lie in the same partition and $a < b$ or if $b < a$ and a, b belong to different partitions.

Lemma 2.1.5. *In the notation above, Γ and $S(2)$ are isomorphic.*

Proof. Let $a = \frac{u}{v} \in \mathbb{Q}$ such that $u \in \mathbb{Z}$ and $v \in \mathbb{N}_{>0}$ are in lowest terms, 0 being $\frac{0}{1}$. If $a \in Q_1$ then we define the point $(u, v) \in \mathbb{R}^2$ and if $a \in Q_2$ we define the point $(u, -v) \in \mathbb{R}^2$. We then proceed to scaling (u, v) as a vector so that it has unit length and this gives us a dense subset of the unit circle. Of course, we cannot have antipodal points, since Q_1 and Q_2 are disjoint.

By definition, there is an an edge from (u_1, v_1) to (u_2, v_2) if, and only if:

- $a_1 = \frac{u_1}{v_1} < a_2 = \frac{u_2}{v_2}$ and $v_1, v_2 > 0$ (i.e. a_1, a_2 come from the same set), or
- $a_1 = \frac{u_1}{v_1} > a_2 = \frac{u_2}{v_2}$ and $v_1, v_2 < 0$.

In any case we have that $u_1 v_2 - u_2 v_1 > 0$, which is the same as saying that the directed line we are considering has the origin to the right side.

The argument for the other direction is identical, that is, given any point (u, v) if it lies on the upper unit circle we send it to the rational number $\frac{u}{v}$ in Q_1. Otherwise we send it to the rational number $-\frac{u}{v}$ in Q_2. □

To fix notation, we write C_n for the cycle on n nodes and I_n for a graph on n nodes with no edges. For digraphs G_1, G_2 we write $[G_1, G_2]$ for the *directed disjoint sum* of G_1 and G_2, that is, for the digraph given by the disjoint union of G_1 and G_2 extended by $a \to b$ for all $a \in G_1$, $B \in G_2$. In this notation, let $\mathcal{N} = \{I_2, C_1, C_2, [I_1, C_3], [C_3, I_1]\}$. To get a clearer idea, besides unconnected points, self-loops and 2-cycles, the digraphs we forbid are the following:

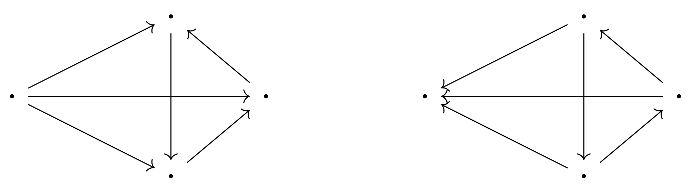


Figure 2.4: The graphs $[I_1, C_3]$ and $[C_3, I_1]$

Let $\mathcal{C} = \text{Forb}(\mathcal{N})$.

Lemma 2.1.6. *In the notation above $Age(\Gamma) = Forb(\mathcal{N})$.*

Proof. It should be clear that any induced finite induced subgraph of Γ does not contain any of the two structures in $\mathcal{N}$. To see this, we will prove that no subgraph G of Γ on four vertices can be isomorphic to any structure in $\mathcal{N}$. If G has no 3-cycle then we are done. Suppose that G has a 3-cycle $v_1 \to v_2 \to v_3 \to v_1$. In this case we must have that two of the nodes in the cycle belong to the same Q_i and the third node belongs to the other set Q_j. This imposes an order in the elements of the cycle. If we consider now the fourth node in G it must be in either Q_1 or Q_2. Hence depending on the labels we have chosen it cannot have an outgoing edge to more than two nodes and it cannot have more than two incoming edges.

Now we prove that we can embed any tournament $D \in \text{Forb}(\mathcal{N})$ into Γ. Let $v \in D$ be an arbitrary vertex and define a partition of D into sets $D_1(v) = \{u \mid u \to v\}$ and $D_2(v) = \{u \mid v \to u\} \cup \{v\}$. We will show that both $D_1(v)$ and $D_2(v)$ are linearly ordered by the relation $\to$.

Note that $I_2 \in \text{Forb}(\mathcal{N})$, so two isolated points do not constitute an induced substructure of D, that is, given any two vertices there exists at least one edge connecting them. Moreover, since $C_2 \in \text{Forb}(\mathcal{N})$, there are no 2-cycles in D, so between every pair of vertices there must be exactly one edge. Finally, since $C_1 \in \text{Forb}(\mathcal{N})$ it follows that there are no self-loops in D. All that says that D is a finite simple tournament. But, in terms of the edge relation being an order it says that it is irreflexive (no self-loops), antisymmetric (no 2-cycles) and connex (tournament). Therefore, we only need to worry about transitivity.

Suppose that we have $x, y, z \in D_1(v)$, such that $x \to y \to z$. We need to show that $x \to z$. Suppose not. Then $z \to x$, so the three points for a subgraph of D isomorphic to C_3. But we also have that $x, y, z \to v$. This is isomorphic to $[C_3, I_1]$, a contradiction. Therefore we have that $x \to z$, as required.

Suppose now that we have $x, y, z \in D_2(v)$, such that $x \to y \to z$. Again, we need to show that $x \to z$. Note that we can suppose that $x \neq v$, since in this case we would be done. Moreover, we must have that $y, z \neq v$, since we have that $v \to y$ and $v \to z$ and the $\to$ relation is antisymmetric. The argument, now, is similar. Suppose not. Then $z \to x$ and again the three form a 3-cycle. Since $v \to x, y, z$ we have an induced subgraph isomorphic to $[I_1, C_3]$, again a contradiction.

Now define a new binary relation $<$ on the domain of D such that $a < b$ if, and only if:

- $a \to b$ and $a, b \in D_i(v)$.
- $b \to a$ and a, b are not in the same set $D_i(v)$.

We claim that this is a linear order. Note that since we have not changed the relations between elements of $D_i(v)$ the subsets remain linearly ordered under the new relation. Moreover, everything is still connected and there are no loops, so we again have to properly check transitivity. Let $x, y, z \in D$, such that $x < y < z$. We need to show that $x < z$. There are eight cases to consider:

1. $x, y, z \in D_1(v)$,
2. $x, y, z \in D_2(v)$,
3. $x, y \in D_1(v)$ and $z \in D_2(v)$,
4. $x, z \in D_1(v)$ and $y \in D_2(v)$,
5. $y, z \in D_1(v)$ and $x \in D_2(v)$,
6. the "duals" of cases $3 - 5$.

As mentioned, cases 1 and 2 follow, because we have not changed anything in $D_i(v)$.

Suppose that we are in case 3. We have that $x \to y$ and since $y < z$ we must have had that $z \to y$. We need to show that $x < z$, or equivalently that $z \to x$, if $z = v$ then we are done, so without loss we may assume that $z \neq v$. Now, suppose for a contradiction that we have $z < x$, then we had $x \to z$ in D. In this case we have that $x \to y$, $x \to z$ and $x \to v$. Moreover, we know that $z \to y$, $y \to v$ and $v \to z$, form a 3-cycle and x points to each vertex in the cycle, i.e. these four nodes form an induced subgraph of D isomorphic to $[I_1, C_3]$, a contradiction. Hence we must have had $z \to x$ in D and thus we get $x < z$, as required.

Now for case 4, we have that $x < y$ and $y < z$, hence $y \to x$ and $z \to y$. Moreover, as always, we may safely assume that $y \neq v$, since if this were the case then we would have $y \to z$ and hence $z < y$. Now, we need to show that $x \to z$ in D. Suppose not. Then $z \to x$ and we have a cycle $y \to x \to v \to y$ and $y \to x$, $z \to x$ and $v \to x$, i.e. a subgraph of D isomorphic to $[I_1, C_3]$ and another contradiction.

For case 5 the argument is similar and the duals do follow in the same way, corresponding to cases we have already examined, but with the contradiction derivable from the reverse disjoint sum each time.

We can thus interpret D as a linear order. So, we can embed D in either Q_1 or Q_2 and hence we are done. □

Moreover:

Lemma 2.1.7. *$S(2)$ is homogeneous.*

Proof. To show this, we will use what is essentially a slightly geometric back and forth argument to extend a partial isomorphism of $S(2)$ to an isomorphism. The argument is by induction. Let $X, Y \subseteq S(2)$ be finite subsets and suppose that we have an isomorphism $f : X \to Y$. Let $u \in S(2) \setminus X$. We will show that there exists an isomorphism $g : X \cup \{u\} \to Y \cup \{v\}$ for some $v \in S(2)$.

Note that u partitions X into two disjoint subsets X_1 and X_2, where $X_1 = \{a \in X \mid a \to u\}$ and $X_2 = \{b \in X \mid u \to b\}$. We note that these two subsets must be linearly ordered by $\to$. We show this for X_1. We need to show that if $a, b, c \in X_1$ and $a \to b$, $b \to c$ then $a \to c$. Indeed, suppose that $c \to a$. Then this gives us a 3-cycle and we know that each of a, b, c points to u, i.e. we have a $[C_3, I_1]$, which is impossible. In particular, $Y_1 = f(X_1)$ and $Y_2 = f(X_2)$ form a partition of Y in distinct linear orders. It suffices to show that there exists some $v \in S(2)$ such that $a \to v$ for all $a \in Y_1$ and $v \to b$ for all $b \in Y_2$.

Let $a_1 \in X_1$ be the largest element in X_1 (i.e. the element that all other elements in X_1 point to) and let $a_2 \in X_1$ be the smallest element of X_1 (i.e. the element in X_1 pointing to all other elements in X_1). Similarly let $b_1 \in X_2$ the largest element in X_2 and $b_2 \in X_2$ the smallest one. Without loss of generality we may assume that the line from b_1 to a_2 has the origin on the right hand side (otherwise we can switch X_1 and X_2, and carry the argument out symmetrically).

Having fixed this assumption, the argument becomes more geometric. We consider antipodal points, on a complete unit circle. In particular, let α_2 and β_1 be the antipodal points of a_2 and b_1, respectively. We know that any point that lies between the antipodal point that lies between α_2 and a_1 is pointed to by all points in X_1 (we denote this set by $(\alpha_2, a_1]$) and that any point that lies between β_1 and b_2 (similarly denoted by $(\beta_1, b_2]$) points to all points in X_2. In particular, u is such a point. The picture (without the edges between X_1 to X_2) is:

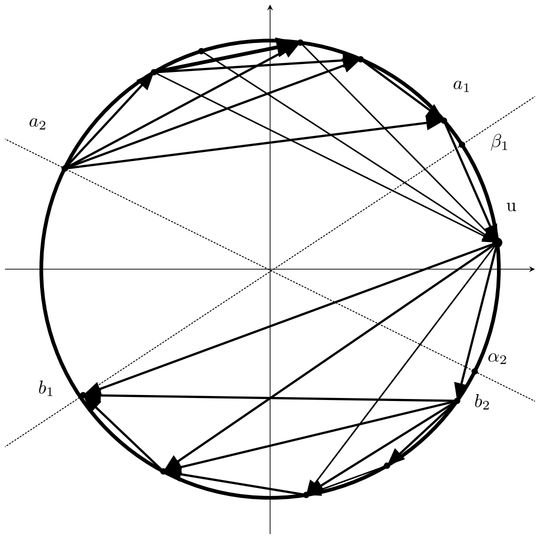


Figure 2.5: Partial Orders in $S(2)$.

It is clear from the picture that the line from b_1 to a_2 has the origin on the right hand side. If the points in X were such that this line had the origin on the left hand side, then the argument below would work, symmetrically.

Now, consider the corresponding sets $(f(\alpha_2), f(a_1)]$ and $(f(\beta_1), f(b_2)]$, where $f(\alpha_2)$ is the antipodal point of $f(a_2)$ and $f(\beta_1)$ the antipodal point of b_2. It suffices to show that their intersection is non-empty. Note that $f(a_1)$ and $f(b_2)$ are distinct points, and hence the set $[f(a_1), f(b_2)]$ is nonempty. So, suppose that the intersection of $(f(\alpha_2), f(a_1)]$ and $(f(\beta_1), f(b_2)]$ is empty. This implies that the directed line from $f(\alpha_2)$ to $f(\beta_1)$ (on the full unit circle) has the origin on the left hand side, since $f(\beta_1)$ occurs on the circle closer to $f(b_2)$ than $f(\alpha_2)$ does. Then, the directed line from $f(a_2)$ to $f(b_1)$ has the origin on the right hand side, which is impossible, since f is a homomorphism and we assumed that b_2 points to a_2.

It follows that the intersection is non-empty. Let v be any point on the intersection of the two sets above. Then if $a' \in Y_1$ we must have that $a' \to v$ and similarly if $b' \in Y_2$ we must have that $v \to b'$. Hence we can define $g : X \cup \{u\} \to Y \cup \{v\}$ extending f by sending u to v.

Since $S(2)$ is countable, by induction we can extend any f to an automorphism of $S(2)$, so it is homogeneous. □

Since $\text{Age}(S(2)) = Age(\Gamma) = \text{Forb}(\mathcal{N})$. It follows from Theorem 2.1.9, that $\text{Forb}(\mathcal{N})$ is an amalgamation class and indeed it is not hard to see that $\text{Age}(\Gamma)$ has the strong amalgamation property. We have that $S(2)$ is universal for $\text{Forb}(\mathcal{N})$, that is, $S(2) = \text{Flim}(\mathcal{N})$.

Homogeneous Digraphs We move on to homogeneous digraphs, in general. For our purposes, it suffices to show that one can construct uncountably many distinct homogeneous digraphs whose age has the free amalgamation property.

We start with an arbitrary (possibly infinite) set of tournaments $\mathcal{T}$ such that no $T \in \mathcal{T}$ is contained in any other $T' \in \mathcal{T}$, as an induced subgraph. We can see that for any such set $\text{Forb}(\mathcal{T})$ is a free amalgamation class, since any $T \in \mathcal{T}$ is by definition connected, so given any digraphs $G, G_1, G_2 \in \text{Forb}(\mathcal{T})$ such that $G \subseteq G_1$ and $G \subseteq G_2$ the free amalgam of G_1 and G_2 over G will not contain any elements of $\mathcal{T}$ as substructures.

It follows that for any $\mathcal{T}$ as above, $\text{Flim}(\text{Forb}(\mathcal{T}))$ is a countable homogeneous (and hence ω-categorical) digraph. If $\mathcal{T}_1 \neq \mathcal{T}_2$ we can see that $\text{Flim}(\text{Forb}(\mathcal{T}_1)) \neq \text{Flim}(\text{Forb}(\mathcal{T}_2))$, since if $T \in \mathcal{T}_1$ is not in $\mathcal{T}_2$ then $\text{Flim}(\text{Forb}(\mathcal{T}_2))$ will contain T as induced subgraph (and similarly if $T \in \mathcal{T}_2$ is not in $\mathcal{T}_1$). To show that there are, in fact, uncountably many of them we just need to show that there are at least infinitely many distinct finite tournaments (and then take the powerset). The existence of such a set is a result of Henson [15]. We call the digraphs of the form $\text{Flim}(\text{Forb}(\mathcal{T}))$, where $\mathcal{T}$ is a set of tournaments, the *Henson Digraphs*.

2.1.11 Types

In this section we give a broad overview of *types*, with our goal being to present the countable *Omitting Types Theorem.*

Definition 2.1.21. Let τ be a signature and Σ a τ-theory. An *n-type* of Σ is a set of formulas $\Phi(\bar{x}) = \{\phi(\bar{x})\}$, where $\bar{x} = (x_1, \ldots, x_n)$, such that for some model Γ of Σ and some n-tuple $\bar{a} = (a_1, \ldots, a_n)$ from Γ we have that $\Gamma \models \phi(\bar{a})$ for all $\phi(\bar{x}) \in \Phi(\bar{x})$. In this case, we say that Γ *realises* Φ. If for some τ-structure Γ' no such n-tuple exists then we say that Γ' *omits* $\Phi(\bar{x})$.

We say that a set of τ-formulas Φ is a *type* if Φ is an n-type for some $n \in \mathbb{N}$. So, $\Phi(\bar{x})$ is a type if $\bar{x}$ is finite and for some τ-structure Γ realises Φ. Types are essentially ways of describing sets of elements of a structure.

We will now define a central notion in the theory of types, complete types. Let Γ be a τ-structure, X a countable set of elements of Γ and $\bar{b}$ a tuple of elements of Γ. Let $\bar{a}$ be a sequence listing all elements of X. Then, the *complete type of $\bar{b}$ over X*, with respect to Γ, over variables $\bar{x}$ is defined to be the set of τ-formulas $\psi(\bar{x}, \bar{a})$ such that $\Gamma \models \psi(\bar{b}, \bar{a})$. We write $\mathrm{tp}_\Gamma(\bar{b}/X)$ or $\mathrm{tp}_\Gamma(\bar{b}/\bar{a})$ for the complete type of $\bar{b}$ over X with respect to Γ, where in the second case $\bar{a}$ lists the elements of X.

Note that, since formulas are finite and $\bar{a}$ will in general be infinite, only a finite part of X will be mentioned in each $\psi(\bar{x}, \bar{a}) \in \mathrm{tp}_\Gamma(\bar{b}/\bar{a})$. In this light, the complete n-type of $\bar{b}$ over X, with respect to Γ, over variables $\bar{x}$ essentially is "everything we can say about $\bar{b}$ in terms of X". We call the elements of X the *parameters* of the complete type. We write $\mathrm{tp}_\Gamma(\bar{b})$ for $\mathrm{tp}_\Gamma(\bar{b}/\emptyset)$, the type of $\bar{b}$ over the empty set of parameters.

Definition 2.1.22. Let τ be a signature, Γ a τ-structure and $X \subseteq \Gamma$. Let $p(\bar{x})$ be a set of τ-formulas in variables $\bar{x}$, with parameters from X. We say that $p(\bar{x})$ is a *complete type* over X, with respect to A in variables $\bar{x}$ if it is the complete type of some tuple $\bar{b}$ over X with respect to some elementary extension of Γ.

Intuitively, a complete type over X is everything we can say, in terms of X about some tuple $\bar{b}$ which may or may not exist in A (but it does exist in some elementary extension of A). Note that a type $\Phi(\bar{x})$ is complete if, and only if for each first-order τ-formula $\phi(\bar{x})$ we have that either $\phi(\bar{x}) \in \Phi(\bar{x})$ or $(\neg\phi(\bar{x})) \in \Phi(\bar{x})$, but not both.

In this notation, a *type* over X, with respect to Γ in the variables $\bar{x}$ is a subset of some complete type over X. If every finite subset of a type is realised in a structure we say that it is *finitely realised.*

Theorem 2.1.10. *Let τ be a fixed signature, A a τ-structure, X a set of elements of A and $\Phi(\bar{x})$ a set of τ-formulas with parameters from X, where $\bar{x} = (x_1, \ldots, x_n)$. Then:*
(a) $\Phi(\bar{x})$ is a type over X with respect to A if, and only if $\Phi(\bar{x})$ is finitely realised in A.
(b) $\Phi(\bar{x})$ is a complete type over X with respect to A if, and only if $\Phi(\bar{x})$ is a set of τ-formulas with parameters from X which is maximal with the property tat it is finitely realised in A.
In particular, if $\Phi(\bar{x})$ is finitely realised in A then it can be extended to a complete type over X, with respect to A.

Proof. The proof uses *elementary diagrams.* For this reason we refer the reader to [4] (Theorem 5.2.1) for a complete presentation. □

Now, if $X = \emptyset$ is the empty set of parameters, then it should be clear from the proof that whether a given set of formulas is a type with respect to Γ solely depends on the theory of Γ. It thus makes sense to call types over the empty set, with respect to some τ-structure Γ the *types of Th*(Γ). More generally we can say that for any τ-theory Σ and any set of τ-formulas $\Phi(\bar{x})$ such that for every finite subset $\Psi(\bar{x}) \subseteq \Phi(\bar{x})$ we have that $\Sigma \cup \{\exists\bar{x}\Psi(\bar{x})\}$ is consistent, then $\Phi(\bar{x})$ is a type of Σ.

The theorem also motivates the following definition, which agrees with our previous terminology:

Definition 2.1.23. Let τ be a signature and Σ a τ-theory. Let $\Phi(\bar{x})$ a type of Σ. Then $\Phi(\bar{x})$ is a *complete type of* Σ if it is a maximal type of Σ with respect to inclusion.

The goal of this section is to describe when such a Σ has a model that omits a type $\Phi(\bar{x})$. To this end, suppose that there exists some τ-formula $\theta(\bar{x})$ such that:

(a) $\Sigma \cup \{\exists\bar{x}\theta(\bar{x})\}$ is satisfiable.

(b) For every formula $\phi(\bar{x}) \in \Phi(\bar{x})$ we have that $\Sigma \models \forall\bar{x}(\theta(\bar{x}) \to \phi(\bar{x}))$.

On the one hand, it is clear that if Σ is a complete theory then $\Sigma \vDash \exists\bar{x}\theta(\bar{x})$, by (a). So, (b) then shows that every model of Σ must also be a model of $\Phi(\bar{x})$, i.e. every model of Σ realises $\Phi(\bar{x})$.

If θ is as above, we say that $\theta(\bar{x})$ is a *support* of $\Phi(\bar{x})$ over Σ. Moreover:

- If $\theta(\bar{x}) \in \Phi(\bar{x})$ we say that $\theta(\bar{x})$ *generates* $\Phi(\bar{x})$.
- If $\Phi(\bar{x})$ has a support over Σ then we say that $\Phi(\bar{x})$ is a *supported type* over Σ.
- If $\Phi(\bar{x})$ has a generator then we say that $\Phi(\bar{x})$ is a *principal type* over Σ.

Let Γ be a τ-structure and $p(\bar{x})$ be a complete type over the empty set with respect to Γ, then by definition any τ-formula $\phi(\bar{x})$ that generates $p(\bar{x})$ is a support for $\phi(\bar{x})$. Clearly, the converse is also true, that is, if $\phi(\bar{x})$ is a support of $p(\bar{x})$, then we have that $\Gamma \vDash \exists\bar{x}\phi(\bar{x})$, so, by definition $\phi(\bar{x}) \in p(\bar{x})$, i.e. $\phi(\bar{x})$ generates $p(\bar{x})$. Therefore, a complete type p is principal if, and only if it is supported. We will call an τ-formula $\phi(\bar{x})$ *complete* for some theory Σ if it generates a complete type of Σ.

Theorem 2.1.11 (Countable Omitting Types Theorem, Theorem 6.2.1 in [4])**.** *Let τ be a countable signature and Σ a consistent τ-theory, such that for each $n \in \mathbb{N}$ there is a set of τ-formulas $\Phi_n(\bar{x})$ that has no support over Σ. Then Σ has a model that omits $\Phi_n(\bar{x})$ for all $n \in \mathbb{N}$.*

The importance of this theorem will become clear (if it is not so already) when we prove the characterisations of ω-categorical theories in the next section.

Proof. Omitted. □

2.1.12 Countable Categoricity

In this section we will state what constitutes essentially the fundamental theorem for ω-categorical theories. The exposition follows [4], again.

Theorem 2.1.12 (Ryll-Nardzewski Theorem, Theorem 6.3.1 in [4])**.** *Let τ be a countable relational signature and Σ a complete τ-theory which has at least one infinite model. Then the following are equivalent:*
(a) Σ is ω-categorical.
(b) If Γ is any countable model of Σ then $Aut(\Gamma)$ is oligomorphic.
(c) Any countable model of Σ realises only finitely many complete n-types for each $n \in \mathbb{N}$.
(d) For each $n \in \mathbb{N}$ there are only finitely many τ-formulas $\phi(\bar{x})$, in n free variables that are non-equivalent modulo Σ.
(e) For any countable model Γ of Σ, for each $n \in \mathbb{N}$, every n-type of Σ is principal and realised in Γ.

The proof requires the Countable Omitting Types Theorem and so it is omitted.

Corollary 2.1.8. *Let τ be a finite relational language and Γ a countable homogeneous τ-structure. Then Γ is ω-categorical.*

Proof. For each $n \in \mathbb{N}$ there are only finitely many isomorphism types of substructures of Γ of cardinality n, since the signature is finite and relational (these are given by the finitely many combinations that n elements can be in relations of τ together). Let Δ_1 and Δ_2 be substructures of Γ of the same isomorphism type. By definition, then, there exists an isomorphism $f : \Delta_1 \to \Delta_2$ and since Γ is homogeneous, this isomorphism can be extended to an automorphism of Γ. In particular, the elements of Δ_1 and Δ_2 are in the same orbit of $\mathrm{Aut}(\Gamma)$, i.e. if two substructures have the same isomorphism type then their elements are in the same orbit of $\mathrm{Aut}(\Gamma)$. Conversely, if $\bar{a}$ and $\bar{b}$ are in the same n-orbit of $\mathrm{Aut}(\Gamma)$ then since isomorphism preserve formulas they have the same isomorphism type. Since there are finitely many isomorphism types for each n, we have that $\mathrm{Aut}(\Gamma)$ has finitely many n-orbits, i.e. $\mathrm{Aut}(\Gamma)$ is oligomorphic. Hence, Γ is ω-categorical. □

Note that, as we showed, any Fraïssé Limit is countable and homogeneous and hence if we assume that τ is a finite relational language then we will have an ω-categorical structure. This immediately gives us multiple examples of structures that display the properties in the theorem above. We have the following corollary, which is more or less a concise reformulation of (b) and (d), from above:

Corollary 2.1.9. *Let τ be a countable signature and let Γ be a countable ω-categorical τ-structure. For all $n \in \mathbb{N}$, we have that the n-tuples of elements of A $\bar{a}$, $\bar{b}$ are in the same orbit under $Aut(\Gamma)$ if, and only if they satisfy the same τ-formulas.*

This concludes all the model theoretic background that we will need for the next chapter. Now, a discussion on the complexity of Constraint Satisfaction Problems would not be complete without introducing some fundamental preliminaries from Complexity Theory. We will do so in the next section.

2.2 Basic Complexity Theory

In a sense, despite the fact that we will be discussing the complexity of CSPs, we need very few tools from Complexity Theory for most of the discussion.

What we will be assuming is the classes **P** and **NP**, **P**-time reductions, the notion of "hardness" and the "guess and check" method of determining if a problem in in **NP**. The presentation in this chapter follows [5].

Moreover, we will mention that **NP**-intermediate problems exist and will be using some Descriptive Complexity, assuming only *Fagin's Theorem* (Theorem 2.2.1). In general, the formal discussion of Complexity Theory in Chapter 3 will be kept informal.

Nonetheless, Descriptive Complexity is an interesting intersection point of Finite Model Theory and Complexity Theory and we will present in this section enough material to prove Fagin's Theorem. This part of the section is based on [6] and [16].

2.2.1 Basic Definitions

Any formal discussion of Complexity Theory needs to start by defining some *model of computation.*

Determinism Our model of choice are deterministic Turing Machines.

Definition 2.2.1. A *deterministic Turing Machine M* is a 5-uple $(Q, \Sigma, q_0, (\mathbf{acc}, \mathbf{rej}), \delta)$, where
- Q is the finite set of *states* of M.
- $\Sigma \cup \{\rhd\}$ is the finite *alphabet* of M.
- $q_0 \in Q$ is the *initial state.*
- $\mathbf{acc}, \mathbf{rej} \subset Q$ are the *accepting*, and *rejecting states.*
- $\delta : (Q \times \Sigma) \to Q \times \Sigma \times \{0, \pm 1\}$ is the *transition function* of M.

We assume that $Q \cap \Sigma = \emptyset$, that $\rhd \notin \Sigma$ (we call $\rhd$ the "start" symbol), and that for each fixed alphabet Σ all the states Q of any Turing Machine M with alphabet Σ come from some countable set.

The transition function δ is the "algorithm" of M. To understand how M acts as a "computer" we think of M as having a finite (but infinitely extendable) *tape* consisting of a string of symbols from Σ and a *head* which always points at one position of the tape (starting at the "start") symbol. Intuitively, the head can read the square that it is pointing at and it is only allowed to write in that square. It can then move to the left or right on the tape, or stay at the same square.

We write $\Sigma^\star$ for finite strings of symbols from Σ.

We give a short description of how Turing Machines operate.

- The tape starts with the "start" symbol and a string $w \in \Sigma^\star$. This is the *input* of the Turing Machine; We indicate that the input of M is the string w by $M(w)$. The head initially points to the "start" square; The initial state is q_0.
- A *configuration* of M is a triple (q, l, r), where $q \in Q$ is a state, $l \in \Sigma^\star$ is a string, which contains the contents of the tape of M, under and on the left of the head and $r \in \Sigma^\star$ a string which contains the contents of the tape strictly on the right of the head.

  In this notation, the *initial configuration* of $M(x)$ is (q_0, start, w).
- A computation of $M(x)$ is given by a sequence of configurations $c_0, c_1, \dots$ where for all i we have that if $c_i = (q, l, r)$ and $c_{i+1} = (q', l', r')$, where writing $l = al_0$ and $r = r_0 b$ for $l_0, r_0 \in \Sigma$ and $a, b \in \Sigma^\star$ then:

  - If $\delta(q, l_0) = (q', a', +1)$ then $c_{i+1} = (q', (aa'r_0), b)$ (i.e. the symbol under the head changed from a to a' and the head moved to the right).
  - If $\delta(q, l_0) = (q', a', -1)$ then $c_{i+1} = (q', (aa, a'r_0b)$ (i.e. the symbol under the head changed from a to a' and the head moved to the left).
  - If $\delta(q, l_0) = (q', a', 0)$ then $c_{i+1}(q', (aa'), r)$, (i.e. the symbol under the head changed from a to a' and the head stayed where it was).

  A complete axiomatisation of how the computation is carried out is expressed formally in the proof of Theorem 2.2.1, where we give a logical characterisation of computation.

- If any of the states in $\mathbf{acc} \cup \mathbf{rej}$ is reached on input w, we say that M *halts*. Since we are only interested in decision problems we do not consider Turing Machines to have output, other than **acc** and **rej**.

Let M be a deterministic Turing Machine. To fix notation, if M on input w reaches one of the states in **acc** we write $M(w) \downarrow \mathbf{acc}$ and call $M(w)$ an *accepting computation*, if it reaches one of the states in **rej** we write $M(w) \downarrow \mathbf{rej}$ and say that $M(w)$ was a *rejecting computation*. If, on input w, none of these states is reached, i.e. if M does not halt, we write $M(w) \uparrow$.

We will not consider M to have multiple tapes. Our measure of "time" is the number of steps in the computation of M until a halting state is reached. We note that a bound on time immediately gives us a bound on space, since if the computation of $M(x)$ halts after n steps then M cannot have used more than n squares to carry out this computation, since the head moves one square at a time. In particular, we define:

Definition 2.2.2. Let $f : \mathbb{N} \to \mathbb{N}$. A deterministic Turing Machine M is said to *operate within time* $f(n)$ if for every word $w \in \Sigma^\star$ we have that $M(w)$ halts in at most $f(|w|)$ steps, where $|w|$ is the length of w.

We are, fundamentally, interested in "decision problems". Formally, for a fixed alphabet Σ and a fixed language $\mathcal{L} \subseteq \Sigma^\star$, a decision problem is a problem of the form:

GIVEN	A finite string $w \in \Sigma^\star$.
QUESTION	Is $w \in \mathcal{L}$?

In terms of Turing Machines, we have the following:

Definition 2.2.3. Let $\mathcal{L} \subseteq \Sigma^\star$ be a language and M a deterministic Turing Machine with alphabet Σ. We say that M *decides* $\mathcal{L}$ if for any string w from Σ:
- Whenever $w \in \mathcal{L}$ we have that $M(w) \downarrow \mathbf{acc}$.
- Whenever $w \notin \mathcal{L}$ we have that $M(w) \downarrow \mathbf{rej}$.

Similarly, we say that M *accepts* $\mathcal{L}$ if:
- Whenever $w \in \mathcal{L}$ we have that $M(w) \downarrow \mathbf{acc}$.
- Whenever $w \notin \mathcal{L}$ we have that $M(w) \uparrow$.

If a language $\mathcal{L}$ is decided by some deterministic Turing Machine operating within time $f(n)$ then we say that:

$$\mathcal{L} \in \mathbf{TIME}(f(n)).$$

In this notation, we define:

$$\mathbf{P} = \bigcup_{k=0}^{\infty} \mathbf{TIME}(n^k).$$

We call problems in **P** *tractable*. The *Invariance Thesis*, first proposed in [17], states that all "reasonable" models of computation have the same complexity as deterministic Turing Machines, up to a polynomial factor. This thesis lies at the heart of Complexity Theory, as it allows us to discuss computation informally, and under the assumption that our model of computation is "reasonable", the class **P** is independent of the choice of model that we make.

Nondeterminism We move now from deterministic to nondeterministic Turing Machines. The only difference is that the transition function δ is no longer a function to $Q \times \Sigma \times \{0, \pm 1\}$. Instead, we have:

$$\delta : (Q \times \Sigma) \to \mathcal{P}(Q \times \Sigma \times \{0, \pm 1\}),$$

where $\mathcal{P}$ denotes the powerset operator. The computation of nondeterministic Turing Machine M on input w is, essentially a "tree" of computations, like the ones we described in the previous section.

Definition 2.2.4. Let $\mathcal{L} \subseteq \Sigma^\star$ be a language and let $w \in \mathcal{L}$. We say that a nondeterministic Turing Machine M *accepts* w if $M(x) \downarrow \mathbf{acc}$ on one of the paths on the tree of $M(w)$. We say that M accepts $\mathcal{L}$ if $\mathcal{L} = \{w \in \Sigma^\star \,|\, M \text{ accepts } w\}$.

We want to extend our notion of time bounds to nondeterministic Turing Machines. To this end, with the notation of Definition 2.2.2 we say that a nondeterministic Turing Machine M *operates within time* $f(n)$ on input w if the depth of the computation tree of $M(x)$ is bounded by $f(|w|)$.

Definition 2.2.5. Let M be a nondetermiinistic Turing Machine. We say that M decides a language $\mathcal{L}$ within time $f(n)$ if M operates within time $f(n)$ and M accepts $\mathcal{L}$.

If a language $\mathcal{L}$ is decided by some nondeterministic Turing Machine operating within time $f(n)$ then we say that:

$$\mathcal{L} \in \mathbf{NTIME}(f(n)).$$

In this notation, we define:

$$\mathbf{NP} = \bigcup_{k=0}^{\infty} \mathbf{NTIME}(n^k).$$

It may not be immediately obvious but we can simulate a nondeterministic Turing Machine operating within time $f(n)$ by a deterministic Turing Machine operating within time $c^{f(n)}$ for some constant $c > 1$ (the point is that in the simulation we make a time-space trade off). For a proof, we refer the reader to Theorem 2.6 in [5]. In the notation we have introduced, this says that:

$$\mathbf{NTIME}(f(n)) \subseteq \bigcup_{c>2} \mathbf{TIME}\left(c^{f(n)}\right).$$

Note that if $\mathcal{L} \subseteq \Sigma^\star$ is any language we define the *complementary language* $\bar{L}$ to be $\Sigma \setminus \mathcal{L}$. We define the class co-**NP** to be the class:

$$\text{co-}\mathbf{NP} = \{\bar{\mathcal{L}} \,|\, \mathcal{L} \in \mathbf{NP}\}.$$

Of course, it is not known if $\mathbf{NP} = \text{co-}\mathbf{NP}$.

Note that, on the other hand, it is a well known fact there are countably many languages $\mathcal{L} \subseteq \Sigma^\star$, for a fixed Σ that can be decided by Turing Machines (nondeterministic or not). This is easy to see from the definition, since there are at most countably many possible transition functions δ, as Σ is finite and any Q is a finite subset of a countable set. In fact, we may encode any Turing Machine, for arbitrary Σ, to one using $\Sigma' = \{0, 1\}$ and hence the number of decidable languages, in general, is independent of Σ.

Problem Reduction One of the fundamental ideas of Complexity Theory is that of *problem reduction.*

Let $\mathcal{L}, \mathcal{L}'$ be languages. We say that $\mathcal{L}$ is **P**-*time reducible* to $\mathcal{L}'$, written $\mathcal{L} \leq_{\mathbf{P}} \mathcal{L}'$ if there exists a map f such that $w \in \mathcal{L}$ if, and only if $f(w) \in \mathcal{L}'$ and f can be computed by a deterministic Turing machine operating in **P**-time[2]. If $\mathcal{L} \leq_{\mathbf{P}} \mathcal{L}'$ and $\mathcal{L}' \leq_{\mathbf{P}} \mathcal{L}$ then we say that $\mathcal{L}$ and $\mathcal{L}'$ are *polynomially equivalent* or **P**-*time equivalent*, written $\mathcal{L} \sim_{\mathbf{P}} \mathcal{L}'$.

Let $\mathcal{C}$ be a complexity class. We say that a language $\mathcal{L}$ is $\mathcal{C}$-*hard* if every language in $\mathcal{C}$ is **P**-time reducible to $\mathcal{L}$. We say that $\mathcal{L}$ is $\mathcal{C}$-*complete* if $\mathcal{L}$ is $\mathcal{C}$-hard and $\mathcal{L} \in \mathcal{C}$.

All of our discussion thus far has been in terms of languages, but under the assumption that any instance of a decision problem can be concisely encoded into a language, which will then be decided by a Turing Machine (see Section 2.2.2 for an extended example) we might as well discuss problems directly.

We give here a very simple and intuitive characterisation of the class **NP**, given by the *"guess and check"* method:

Given a problem P if we can guess a polynomially bounded (in size) solution for P which can be verified in **P**-time, in the size of the guess, then P is in **NP**.

We will use this sort of language when discussing the complexity of specific problems in the next chapter. A complete discussion is given in [5]. The intuition is that a nondeterministic Turing Machine M, nondeterministically follows

[2] We have not formally defined what it means for a Turing Machine to compute a function $f : \mathbb{N} \to \mathbb{N}$, but, intuitively, the Turing Machine has a set of halting states **halt** instead of **acc** and it reaches one of these states with the output of the function (e.g. in binary) printed on the tape.

a path for each of the possible solutions and since each path has **P**-many steps, M decides P in **NP**. The guess then is the "succinct certificate" that one of the computations was an accepting one.

In terms of arguing about **NP**-completeness, one first has to show that a single problem is **NP**-complete, by arguing that any nondeterministic **P**-time computation can be reduced to a solution of the specific problem. Cook and Levin did this to show that Boolean Satisfiability (commonly referred to as SAT) is **NP**-complete in [18]. Other famous **NP**-complete problems, which we will discuss in the next chapter include 3-colourability (3COL) and BETWEENNESS.

Another very important result of classical Complexity Theory, which we will assume, without proof is that if $\mathbf{P} \neq \mathbf{NP}$ then there exist problems that are not in **P** and are not **NP**-complete. This is a famous result of Lander [19]. Of course, the assumption in the proof is that $\mathbf{P} \neq \mathbf{NP}$ and we do not have any natural candidates for **NP**-intermediate problems. This is part of the motivation behind the study of dichotomous classes of problems, since proving that a class is not dichotomous may give us some natural candidates for **NP**-intermediate problems.

2.2.2 Descriptive Complexity

At this point, we will start to delve in the realms of *Descriptive Complexity Theory*, the branch of Finite Model Theory that revolves around the fundamental idea that "a problem is as complex to solve as the language needed to express it, and vice versa".

Our aim in this section is to make this seemingly intuitive and yet surprising correspondence precise. We start by defining the appropriate machinery and present a proof of the celebrated theorem of Fagin (originally in [20]), which states that second-order existential logic ($\mathbf{SO}\exists$) precisely captures **NP**. First, we need to quickly introduce *second-order logic.*

Second-Order Logic

Let τ be a signature. A *second-order logic* formula ϕ is a formula:

$$\bar{Q}\bar{X}\phi$$

where $\bar{Q}$ is a sequence of second-order $\forall$ and $\exists$ quantifiers, $\bar{X}$ is a sequence of second-order variables, with associated arities $\bar{n}$, and ϕ is a first-order $(\tau \cup \{\bar{X}\})$-formula. We may write X_n to indicate that X has arity n, but usually this will be clear from the context.

We define truth in a structure in the standard way (we do not follow Henkin semantics). In particular, we keep all the notions from Tarski, for the first-order part and define $\Gamma \vDash \forall X_n \phi$ where ϕ is a second-order formula in $\tau \cup \{X_n\}$ if for all subsets $X' \subseteq \Gamma^n$ we have that $(\Gamma, X_n) \vDash \phi$ and similarly $\Gamma \vDash \exists X_n \phi$ where ϕ is a second-order formula in $\tau \cup \{X_n\}$ if there exists a subset $X' \subseteq \Gamma^n$ we have that $(\Gamma, X_n) \vDash \phi$.

We define existential second-order logic ($\mathbf{SO}\exists$) to be the subset of second-order logic consisting of all second-order logic sentences whose second-order quantifications are existential.

2.2.3 Encoding Structures

Complexity Theory is and always has been interested in the study of computation based on classical models such as Turing Machines. In order to give a grounding to the study of Descriptive Complexity we thus need to choose a computational model (the choice is, in fact, not important, as under the Invariance Thesis they are all equivalent, in terms of the problems that we identify as tractable). We also need to provide a method of encoding structures into inputs for the chosen model. We will choose, again, Turing Machines. The discussion in this section is based on [16].

Turing Machines operate, as mentioned, on inputs that are words over a fixed finite alphabet. Note that we will only be discussing finite structures in finite relational signatures. For any finite structure Γ we can thus identify the domain of Γ with an initial segment of the positive integers. In fact, if Γ contains n objects there are $n!$ ways of doing so, without any assumptions on the order of these objects. Such an order needs to be chosen, for the encoding procedure to be well-defined. In this case we may simply add a new binary relation $<$ on $\mathrm{dom}(\Gamma)$, which is meant to represent a linear order on the elements of $\mathrm{dom}(\Gamma)$.

Given a structure $(\Gamma, <)$, as described above, where $|\text{dom}(\Gamma)| = n$ we may identify $\text{dom}(\Gamma)^k$ with the set $\{0, 1, \ldots, n^k - 1\}$ by associating each k-tuple with its rank in the lexicographical order induced by $<$. We can thus encode ordered structures as binary strings, e.g. by identifying by 1's the tuples that belong to each k-ary relation. There are many choices for encoding Γ as a string in a finite alphabet Σ. Before giving an explicit encoding, that we will assume in the proof of Fagin's Theorem, we need to mention that any encoding we choose must have the following properties:

(i) It must preserve isomorphisms, that is, isomorphic structures must be assigned to the same string.

(ii) It must be polynomially bound in the size of the input structure, that is, there must exist a polynomial $p \in \mathbb{N}[X]$ such that the length of the string produced for any Γ is at most $p(|\text{dom}(\Gamma)|)$.

(iii) It must be first-order definable, that is for all $k \in \mathbb{N}$ and symbols $\sigma \in \Sigma$ there exists a first-order $\tau \cup \{<\}$-formula $\beta_\sigma(x_1, \ldots, x_k)$ such that, for all structures Γ and all $\bar{a} \in \text{dom}(\Gamma)^k$ the following holds

$$\Gamma \vDash \beta_\sigma(\bar{a}) \text{ if, and only if the } \bar{a}\text{-th symbold of the code of } \Gamma \text{ is } \sigma.$$

There is another point made in [16], which says that an encoding must also:

(iv) It must allow us to compute the values of atomic statements "efficiently",

but we have not defined the necessary machinery to make this precise and we will assume that it is always the case.

Now we define a fairly intuitive encoding, for a Turing Machine that has $\Sigma = \{\cdot, 0, 1\}$. We will write a^n, for $a \in \Sigma$ to mean a string given by the concatenation of n consecutive copies of the symbol a.

Let $\tau = \{R_1, \ldots, R_t\}$ be a finite relational signature, let Γ be a finite τ-structure and $<$ be a linear order on $\text{dom}(\Gamma)$, which has cardinality n. Let l be the maximal arity of the relations in τ. For each relation R of arity j we associate a string $\chi(R) = w_0 \ldots w_{n^j - 1} 0^{n^l - n^j} \in \{0, 1\}^{n^l}$, where $w_i = 1$ if, and only if the i-th tuple of A^j belongs to R. Now we set the code of Γ to be $1^n \cdot 1^l \cdot \chi(R_1) \cdot \ldots \cdot \chi(R_t)$. The intuition is that we need to tell M how many elements are in $\text{dom}(\Gamma)$ and what the arity of the largest relation in τ is and then we can give all of the relations as encoded above. We introduce $\cdot$ as a symbol to make the encoding easier to understand, but, as mentioned, we can provide an alternative encoding using only 0's and 1's, with only a linear slow-down.

When we say that a Turing Machine M decides a class $\mathcal{C}$ of τ-structures we mean that M decides the set of encodings of structures of $\mathcal{C}$.

2.2.4 Fagin's Theorem

Fagin's Theorem is one of the fundamental results in Descriptive Complexity theory. It states that **SO**∃ captures **NP** on the class of all finite structures. In fact, this is one of the strongest results in Descriptive Complexity, as it assumes nothing about the structures themselves. Such results for other complexity classes remain open for arbitrary structures. We have similar capturing results, though, for classes of *ordered* finite structures.

Theorem 2.2.1 (Fagin's Theorem)**.** *Let τ be a fixed, non-empty, finite relational signature and let $\mathcal{C}$ be an isomorphism-closed class of finite τ-structures. Then $\mathcal{C}$ is in* **NP** *if, and only if $\mathcal{C}$ is definable by an* **SO**∃ *sentence.*

Proof. We need to show that we can (a) decide a class of finite models of an **SO**∃-sentence (also referred to as a *generalised spectrum*) using a nondeterministic Turing Machine and (b) any nondeterministic Turing Machine operating in **P**-time decides a generalised spectrum.

Let ψ be the **SO**∃ sentence $\exists R_1 \ldots \exists R_n \phi$, where ϕ is a first-order $\tau \cup \{R_1, \ldots, R_n\}$-sentence. We define a nondeterministic Turing Machine M which given a τ-structure Γ together with an arbitrary order $<$ on $\text{dom}(\Gamma)$ decides if $(\Gamma, <) \vDash \psi$.

First, M nondeterministically guesses the relations R_i. We can easily see that the guess is polynomially bound on the size of the input which is the encoding of Γ. Then M checks if $\langle \Gamma, <, R_1, \ldots, R_n \rangle \vDash \phi$. Since ϕ is in **FO** this can be done **P**-time (in fact it can be done in **LOGSPACE**, but we have not defined it here).

Since the computation of M is precisely a polynomially bound guess followed by a polynomial time computation, the generalised spectrum $\mathcal{C}$ of ψ is in **NP**.

The other direction will require us to work much harder. Let $\mathcal{C}$ be an isomorphism-closed class of τ-structures and M a non-deterministic (one-tape) Turing Machine, which given as input the code of a finite τ-structure Γ, together with an arbitrary linear order $<$ on its domain decides in polynomial time if $\Gamma \in \mathcal{C}$. We need to construct an $\mathbf{SO}\exists$ sentence ϕ whose finite models are precisely the structures in $\mathcal{C}$.

To fix notation, let $M = (Q, \Sigma, q_0, (\mathbf{acc}, \mathbf{rej}), \delta)$, where Q is the set of states, Σ the alphabet, $q_0 \in Q$ the initial state, $\mathbf{acc} \subseteq Q$ the set of accepting states and $\delta : (Q \times \Sigma) \to \mathcal{P}(Q \times \Sigma \times \{0, \pm 1\})$ the transition function of M.

We may assume that all computations of M when given the code of a structure Γ of cardinality n terminate after at most $n^k - 1$ for some fixed positive integer k.

We need that ϕ is of the form $\exists < \exists \bar{X} \psi$, where ψ is a first-order $(\tau \cup \{<\} \cup \{\bar{X}\})$-sentence such that $\Gamma \models \psi$ for some $(\{<\} \cup \{\bar{X}\})$-expansion of Γ if, and only if $\bar{X}$ encodes an accepting computation of M of length at most $n^k - 1$ when given the code of Γ with linear order $<$ on the $\mathrm{dom}(\Gamma)$.

Since all computations are bounded by n^k we will represent the n^k squares of each possible computation and the n^k possible steps that the Turing Machine can make by identifying numbers up to n^k with k-tuples from Γ, since each element of Γ will be coded by a number from 1 to n.

In order to be able to encode M we need $\bar{X}$ to contain the following second-order variables:

1. To encode the states, for each state $q \in Q$ we include a k-ary existential relation S_q.
2. To encode the potential contents of the tape, in terms of the alphabet, for each symbol $\sigma \in \Sigma$ we include a $2k$-ary existential relation T_σ.
3. To encode the head we include a $2k$-ary existential relation H.

Intuitively, as we mentioned k-tuples express the numbers from 1 to n^k. In particular, since on each path of the computation a nondeterministic Turing Machines functions just like a deterministic one and a time bound gives a space bound these are all the numbers we will need to code up a computation into a formula. The intuition is that:

1. For each $q \in Q$ the predicate $S_q(\bar{x})$ holds if, and only if the state of the machine at time $\bar{x}$ is q.
2. $T_\sigma(\bar{x}, \bar{y})$ holds if, and only if at time $\bar{x}$ the cell $\bar{y}$ contains the symbol σ.
3. $H(\bar{x}, \bar{y})$ holds if, and only if at time $\bar{x}$ the head is at position $\bar{y}$.

We build ϕ up from these relations and $<$ by including the following conjuncts in ϕ:

1. We need a conjunct to express that $<$ is a linear order. We then extend $<$ to a lexicographic order on k-tuples. This step is easy enough. For k-tuples $\bar{x}$ and $\bar{y}$ we will write $\bar{x} = \bar{y}$ to mean:
$$\bigwedge_{i=1}^{k} (x_i = y_i)$$
and $\bar{x} < \bar{y}$ to mean:
$$\bigvee_{i=1}^{k} \left(\bigwedge_{j=1}^{i-1} (x_j = y_j) \wedge (x_i < y_i) \right).$$
We define $\bar{x} \leq \bar{y}$ in the obvious way.
2. We express the *successor* function on k-tuples. Let $\bar{x}$ and $\bar{y}$ be k-tuples. We define $\textsc{Succ}(\bar{x}, \bar{y})$ to be the formula:
$$(\bar{x} < \bar{y}) \wedge (\forall \bar{z} (\bar{x} < \bar{z} \to ((\bar{z} = \bar{y}) \vee (\bar{z} < \bar{y}))))$$
3. A conjunct to express that the initial state is q_0 and that the head is at the head starts from there:
$$\forall \bar{x} \left((\forall \bar{y} (\bar{x} \leq \bar{y})) \to (S_{q_0}(\bar{x}) \wedge H(\bar{x}, \bar{x})) \right)$$
4. A conjunct to say that the head is never in two positions of the tape at the same time:
$$\forall \bar{x} \forall \bar{y} \left(H(\bar{x}, \bar{y}) \to (\forall \bar{z} (\neg(\bar{y} = \bar{z}) \to (\neg H(\bar{x}, \bar{z})))) \right)$$
5. A conjunct to express that M is never in two states at once:
$$\forall \bar{x} \bigwedge_{q \in Q} \left(S_q(\bar{x}) \to \bigwedge_{q' \neq q} (\neg S_{q'}(\bar{x})) \right)$$

6. A conjunct to express that each cell of the tape contains one symbol:

$$\forall\bar{x}\forall\bar{y}\bigwedge_{\sigma\in\Sigma}\left(T_\sigma(\bar{x},\bar{y})\rightarrow\bigwedge_{\sigma'\neq\sigma}(\neg T_{\sigma'}(\bar{x},\bar{y}))\right)$$

7. A conjunct to express that the initial contents of the state are precisely the code of Γ with the linear order given from $<$.

$$\forall\bar{x}\forall\bar{y}\left((\forall\bar{z}(\bar{x}\leq\bar{z}))\rightarrow\bigwedge_{\sigma\in\Sigma}(\beta_\sigma(\bar{y})\rightarrow T_\sigma(x,\bar{y}))\right),$$

   where β_σ is as in the definition of an encoding in the previous section.

8. A conjunct to express that the only possible place where the tape changes is under the head.

$$\forall\bar{w}\forall\bar{x}\forall\bar{y}\forall\bar{z}\left(((\neg(\bar{x}=\bar{y}))\wedge\textsc{Succ}(\bar{x},\bar{w}))\rightarrow\left(\bigwedge_{\sigma\in\Sigma}(H(\bar{x},\bar{y})\wedge(T_\sigma(\bar{x},\bar{z})\rightarrow T_\sigma(\bar{w},\bar{z}))\right)\right)$$

9. A conjunct to express the transition function δ.

$$\forall\bar{w}\forall\bar{x}\forall\bar{y}\bigwedge_{\sigma\in\Sigma}\left(\bigwedge_{q\in Q}\left((H(\bar{x},\bar{y})\wedge S_q(\bar{x})\wedge T_\sigma(\bar{x},\bar{y})\wedge\textsc{Succ}(\bar{x},\bar{w}))\rightarrow\left(\bigvee_{\Delta}(H(\bar{w},\bar{y}')\wedge S_{q'}(\bar{w})\wedge T_{\sigma'}(\bar{w},\bar{y}))\right)\right)\right),$$

   where Δ is the set of next values that $\delta(q,\sigma)$ takes and q',σ' come from this set and we set $\bar{y}'$ to be either $\textsc{Succ}(\bar{y}',\bar{y})$ if the head moves to the left or $\textsc{Succ}(\bar{y},\bar{y}')$ if the head moves to the right, or just $\bar{y}$ if the head does not move, as in the previous discussion on the computation of Turing Machines.

At this point, we are almost done. We are just missing the most important bit, which is that some computation is accepting, i.e. a clause:

$$\exists\bar{x}S_{\mathbf{acc}}(\bar{x}).$$

Then, putting all of these together, by construction gives us a formula that expresses precisely the computation of M. □

In this section we have focused too much on technical details and we will not do so in the next chapter, but that is for good reason. The ideas in the proof of Fagin's Theorem are in general very important, giving us precise syntactic characterisations for semantic notions. Moreover, the formulaic characterisation we have given is fairly natural (despite its technicality) and precisely describes how nondeterministic Turing Machines operate, so it gives us a nice intuition about the class **NP**, making the notions in the previous section more concrete.

Indeed, for instance this tells us immediately that co-**NP** is precisely SO∀ and it tells us precisely why the polynomial hierarchy **PH** (see [5] for details) is the same class as SO.

Moreover, a large part of the next chapter will be devoted to descriptive complexity, and although this precision will not be necessary there (we will, as is typical in Complexity Theory literature, not be using Turing Machines explicitly) it is important to understand precisely the questions being asked and the inherent reasons fixed logics $\mathcal{L}$ correspond to complexity classes.

Chapter 3

Constraint Satisfaction Problems

In this chapter, we will start discussing Constraint Satisfaction Problems (CSPs) in a systematic way. As a motivation for the study of specific classes of CSPs we note that it is known, as we have previously mentioned, that if $\mathbf{P} \neq \mathbf{NP}$ then **NP**-intermediate problems exist, that is, problems that are neither in **P** nor are **NP**-complete. Complexity Theory is interested in finding large classes of **NP** that do not contain **NP**-intermediate problems, that is, classes in which every problem is either in **P** or is **NP**-complete.

Any class of problems that contains both **P** and **NP**-complete problems but does not contain **NP**-intermediate problems is said to display a *dichotomy*, in terms of the complexity of its problems. We note that such classes are interesting, since at the time of writing there are no known "natural" **NP**-intermediate problems, and proving that a class does not display a complexity dichotomy would potentially lead to a natural candidate.

It has been shown that some classes of CSPs do not indeed contain any **NP**-intermediate problems. A famous conjecture by Feder and Vardi stated that the class of all Constraint Satisfaction Problems on finite templates also displays this dichotomy. This conjecture was finally proved in 2017 in [2] and [3]. We introduce the necessary terminology to define a strictly larger class of CSPs which is also believed to display the same dichotomy and offer a general discussion on the subject.

3.1 Introduction

Informally, a Constraint Satisfaction Problem (CSP) is a problem of the following form:

GIVEN:
- A finite set of **variables**.
- For each variable a (not necessarily finite) **domain** in which this variable lives.
- A finite set of **constraints** on these variables.

QUESTION:
- Is there some **assignment** of values to these variables such that all the constraints are satisfied.

Since the values that the variables are allowed to take and the constraints on these values are essentially arbitrary this setting is clearly very broad, and it encompasses problems originating from all areas of theoretical computer science. We will take a more model theoretic and logical approach, though, generally ignoring specific problems and focusing more on classes of problems. This setting makes it very natural to discuss graph problems and to ask satisfiability questions.

At this point, we have introduced a lot of model theoretic machinery, and we will use it to study the complexity of CSPs systematically. The goal of this chapter, apart from providing a clearer exposition of CSPs, is two-fold. We want to give some general intuition about the basic techniques used in CSPs and we want to justify the reasons why some of these techniques were applied in the first place. In doing so, we will fully define and prove (referring the reader to the literature for some very technical proofs) the following classification:

$$\texttt{CSP} \sim_{\mathbf{P}} MM\mathbf{SNP} \subsetneq \texttt{CSP}^{\star} \subsetneq M\mathbf{SNP} \subsetneq \mathbf{SNP} \sim_{\mathbf{P}} \mathbf{SO}\exists = \mathbf{NP},$$

which will serve as a pool of examples that clarify the abstract definitions. In general, the main results in this chapter have been published by Bodirsky in [7, 8], and other papers that [8] surveys. We attempt to make some of the arguments more clear and we fix some minor errors that we identified in the literature.

3.2 Formal Definitions

We have discussed the general idea behind CSPs, but in order to be able to study them more systematically, using the tools that we have developed thus far we need to define them formally. In this section we will give four, seemingly different, but fundamentally equivalent, approaches to Constraint Satisfaction.

3.2.1 The Homomorphism Approach

We will now give a first formal definition for Constraint Satisfaction Problems. Let τ be a finite relational signature and Γ a fixed τ-structure. Then $\texttt{CSP}(\Gamma)$ is the following computational problem:

Given	A finite τ-structure Δ.
Question	Is there a homomorphism $f : \Delta \to \Gamma$?

We shall call Γ the *template*, Δ an *instance* of $\texttt{CSP}(\Gamma)$ and the homomorphism $f : \Delta \to \Gamma$ the *solution* of the instance. The computational problem then is how difficult it is to find a solution for a given instance. The following example illustrates this point nicely:

Example 3.2.1. Let τ be a signature for digraphs. Let $\Gamma = K_3$. Then $\texttt{CSP}(\Gamma)$ is the problem of deciding whether a given finite graph G we can find a map $f : G \to K_3$ such that for any vertices $v_1, v_2 \in G$ we have that $E(v_1, v_2)$ implies $E(f(v_1), f(v_2))$.

Suppose that G can be coloured with 3 colours, so that no two vertices $v_1, v_2 \in G$ that are connected by an edge (we call these neighbouring vertices) share a colour. Then mapping all the vertices with the same colour to the same vertex of K_3 gives a well defined homomorphism, as any two vertices in G that are connected will map to distinct connected vertices of K_3. On the other hand, if a map G can homomorphically map to K_3 then we can find a proper three colouring of G.

It is, hence, not hard to see that computationally the two problems are equivalent, hence $\texttt{CSP}(K_3)$ is precisely 3Col. This is a known **NP**-complete problem.

Note that for an instance Δ of $\texttt{CSP}(\Gamma)$ even though we need that both τ and Δ are finite, for the computational problem to make sense, we do not need that Γ is. In fact, sometimes it is not possible to express certain problems using finite templates.

Example 3.2.2. Let $\Gamma = (\mathbb{Q}, <)$. Then $\texttt{CSP}(\Gamma)$ is the problem of deciding whether a finite graph has no cycles. This may not be immediately obvious, but we note that $<$ is a perfectly well-defined "edge" relation on $\mathbb{Q}$ and then, any finite digraph with no cycles can be mapped to $\mathbb{Q}$ and of course, any graph for which such a mapping exists is acyclic. The problem Acyclic is a known **P** problem.

Suppose that Γ were a finite template and $\texttt{CSP}(\Gamma) = \texttt{CSP}((\mathbb{Q}, <))$. Let $|\Gamma| = n$. but then the linear order on m elements must be contained in Γ, for all $m \in \mathbb{N}$, what this means is that Γ contains a disjoint directed sum of m independent vertices (in the notation of Section 2.1.10 $[I_1, \dots, I_1]$), for each m, and hence Γ has to be infinite.

It is sometimes more useful to think of $\texttt{CSP}(\Gamma)$ as a class structures, instead. This leads us to the following definition:

Definition 3.2.1. Let τ be a finite signature and Γ a fixed τ-structure. Then $\texttt{CSP}(\Gamma)$ is the class of all finite τ-structures Δ that homomorphically map to Γ.

In the previous chapter we defined Forb($\mathcal{N}$) for some class of τ-structures $\mathcal{N}$ to be the class of all finite structures which do not contain any structure $\Delta \in \mathcal{N}$ as an *induced* substructure (Definition 2.1.6). In the same spirit we define:

Definition 3.2.2. Let τ be a finite relational signature and let $\mathcal{N}$ be a class of τ-structures. We say that a τ-structure Γ is $\mathcal{N}$*-free* if for all structures $\Delta \in \mathcal{N}$ there exist no homomorphisms $f : \Delta \to \Gamma$. We denote the class of all $\mathcal{N}$-free structures by wForb($\mathcal{N}$).

Note that it is easy to see that for any class of structures $\mathcal{N}$ we have that Forb($\mathcal{N}$) $\subseteq$ wForb($\mathcal{N}$). Of course the other direction is obviously not true in general. One notes that if Γ is any structure and Δ is a structure on a non-empty domain with no relations on its elements then $\Delta \in$ wForb(Γ) for any Γ, but if Γ is connected (in the sense that all elements of Γ belong to some relation of arity at least 1) then $\Delta \notin$ Forb(Γ).

In fact, now we can easily say a lot more about CSPs. The following proposition, gives us a concise and convenient way of characterising which classes of problems can be viewed as CSPs for some Γ.

Proposition 3.2.1. *Let τ be a finite relational signature and let $\mathcal{C}$ be a class of finite τ-structures. Then, the following are equivalent:*
1. $\mathcal{C} =$ CSP(Γ) for some (possibly infinite) τ-structure Γ.
2. $\mathcal{C} =$ wForb($\mathcal{N}$) for some class of finite connected τ-structures $\mathcal{N}$.
3. $\mathcal{C}$ is closed under disjoint unions and inverse homomorphisms.

Proof. Suppose, first that $\mathcal{C} =$ CSP(Γ) for some τ-structure Γ. Then we can define $\mathcal{N}$ to be the class of finite connected τ-structures that do not homomorphically map to Γ. Then, just by the definition, we have that $\mathcal{C} =$ wForb($\mathcal{N}$).

Suppose now that $\mathcal{C} =$ wForb($\mathcal{N}$), for some class of finite connected τ-structures. First, we will show that $\mathcal{C}$ is closed under disjoint unions. Let $\Delta_1, \Delta_2 \in \mathcal{C}$ and suppose that for some $\Delta \in \mathcal{N}$, we have a homomorphism $f : \Delta \to \Delta_1 \sqcup \Delta_2$. Note that since Δ is connected there must exist a homomorphism $f' : \Delta \to \Delta_1$ or $f' : \Delta \to \Delta_2$. This is, by assumption, impossible, and hence $\Delta_1 \sqcup \Delta_2 \in \mathcal{C}$. Now, suppose that $f : \Delta' \to \Delta$ is a homomorphism and $\Delta \in \mathcal{C}$. Suppose that Δ' contains a structure from $\mathcal{N}$. Then that substructure would homomorphically map to a substructure in Δ, which is impossible.

Finally, suppose that $\mathcal{C}$ is closed under disjoint unions and inverse homomorphisms. Define Γ to be the structure made up of disjoint unions of all isomorphism classes in $\mathcal{C}$. Then, very structure in $\mathcal{C}$ homomorphically maps into Γ, by construction. Now, suppose that $\Delta \in$ CSP(Γ), so there exists a homomorphism $f : \Delta \to \Gamma$. Then, the image of Δ is contained in the disjoint union of structures from $\mathcal{C}$. Since $\mathcal{C}$ is closed under disjoint unions and inverse homomorphisms it follows that $\Delta \in \mathcal{C}$. ☐

This gives us an easy way of deciding if a computational problem can be expressed as CSP(Γ) for some structure Γ.

Example 3.2.3. Consider the well known problem TRIANGLE-FREE, for undirected graphs. This is exactly the problem CSP(Γ) where $\Gamma =$ wForb(K_3) (forgetting the direction of the edges of K_3). Note that, as we have seen, Forb(K_3) has the free amalgamation property. This implies that Forb(K_3) $=$ wForb(K_3) (by Proposition 3.4.4).

3.2.2 The Sentence Evaluation Approach

We will shift points of view now to give a more logical way of defining CSPs. The point is that we will now define instances of CSPs in terms of logical formulas instead of defining them in terms of structures. As we will see, since we are only interested in finite structures, the two approaches are fundamentally equivalent.

Definition 3.2.3. Let τ be a finite relational signature. A *primitive positive* τ-formula $\phi(\bar{x})$ is first order formula of the form:

$$\exists \bar{y} \left(\bigwedge_{i=1}^{n} \psi_i(\bar{x}, \bar{y}) \right),$$

where the ψ_i are atomic τ-formulas. We say that ϕ is a *primitive positive τ-sentence* if it has no free variables.

Let τ be a finite relational signature and Γ a τ-structure. Then, CSP(Γ) is the following computational problem:

GIVEN	A primitive positive τ-sentence ϕ.
QUESTION	Is ϕ true in Γ?

From this point of view, we call the relations in Γ the *constraints*, ϕ the *instance* of the CSP and the assignment of elements of Γ to the existentially quantified variables of ϕ the *solution* to the CSP. The computational problem is, of course, how hard it is to find such an assignment.

Even though, as we have claimed, the two approaches are equivalent this is not immediately obvious and to show this we need to define two new concepts.

Let τ be a fixed finite relational signature and write $\tau = \{R_1, \dots, R_n\}$, where R_i has arity n_i. If Δ is a finite τ-structure with elements $x_1, \dots, x_l$ and $R_i^\Delta(x_{i_1}, \dots, x_{i_{n_i}})$ we can define the *canonical query* of Δ to be the formula

$$\exists x_1 \dots x_l \left(\bigwedge_{i=1}^{n} R_i(x_{i_1}, \dots, x_{i_{n_i}}) \right)$$

We will denote the canonical query of Δ by $Q(\Delta)$.

Conversely, we can completely reverse this construction and given a primitive positive τ-sentence ϕ, where

$$\exists x_1 \dots x_m \left(\bigwedge_{i=1}^{n} R_i(x_{i_1}, \dots, x_{i_{n_i}}) \right)$$

we define the *canonical database* of ϕ to be the structure with domain $x_1, \dots, x_m$ and $R_i^\Delta(x_{i_1}, \dots, x_{i_{n_i}})$. We will denote the canonical database of ϕ by $D(\phi)$.

If Δ is non-empty and has no relations then we may define the canonical query of Δ to be just $\top$, and if Δ is empty we may define it to be just $\bot$. Conversely we may define the canonical database of $\top$ to be any structure with no relations and the canonical database of $\bot$ to be the empty structure. This is actually sensible, and it does not cause too much trouble. The point is that if Δ is any structure with no relations then $\Delta \in \texttt{CSP}(\Gamma)$ for any Γ, and conversely, any non-empty structure Γ has $\Gamma \vDash \top$ ($\top$ is just shorthand for $\exists x(x = x)$).

The following proposition is immediate from the definitions:

Proposition 3.2.2. *Let τ be a finite relational signature and let Γ be a τ-structure. Then:*
1. *If Δ is a finite τ-structure then there is a homomorphism from Δ to Γ if, and only if $Q(\Delta)$ is true in Γ.*
2. *Let ϕ be a primitive positive τ-sentence (other that $\bot$). Then ϕ is true in Γ if, and only if there is a homomorphism from $D(\phi)$ to Γ*

So, instances of $\texttt{CSP}(\Gamma)$ can be viewed as either finite structures in the same signature or primitive positive sentences.

Recall from Section 2.1, that a τ' structure Γ' is definable from a τ-structure Γ, if $\mathrm{dom}(\Gamma) = \mathrm{dom}(\Gamma')$ and for each n-ary relation R of τ' we have that:

$$R^{\Gamma'} = \{\bar{a} \in \Gamma^n \mid \Gamma \vDash \phi(\bar{a})\},$$

for some τ-formula $\phi(\bar{x})$. We say that Γ' is *primitive positive* definable from Γ if the τ-formulas above are primitive positive. The advantage that this new approach offers us is that we can easily characterise how the CSP of a primitive positive expansion of Γ differs from the CSP of Γ. We have:

Lemma 3.2.3. *Let Γ be a τ-structure and R a primitive positive definable relation in Γ. Then $\texttt{CSP}(\Gamma)$ and $\texttt{CSP}((\Gamma, R))$ are linear time equivalent.*

Proof. Let ϕ be an instance of $\texttt{CSP}(\Gamma)$. Then by definition a solution for this problem reduces for a solution for $\texttt{CSP}((\Gamma, R))$. The converse is not as obvious. Let ϕ be an instance of $\texttt{CSP}((\Gamma, R))$ and suppose that $\psi(x_1, \dots, x_n)$ is the primitive positive definition of $R(x_1, \dots, x_n)$. Let ϕ' be ϕ, where we have replaced each occurrence of R in ϕ with ψ. Since ϕ was already primitive positive we may move all the new existential quantifiers to the front, so that ϕ' remains primitive positive. This construction is linear in the size of ϕ and it is clear that ϕ is aYes instance of $\texttt{CSP}((\Gamma, R))$ if, and only if ϕ' is a Yes instance of $\texttt{CSP}(\Gamma)$. □

What this means is that when we discuss the computational complexity of $\texttt{CSP}(\Gamma)$ we can do so up to primitive positive definable relations in Γ.

3.2.3 The Satisfiability Approach

If we view the shift from the Homomorphism approach to the Sentence Evaluation approach as a shift from translating instances of the CSP from structures into logical formulas it is natural to consider translating the template of the CSP into logic as well. This is the core idea behind this approach.

Let τ be a finite relational signature and Σ a first-order τ-theory. Then $\mathtt{CSP}(\Sigma)$ is the following computational problem:

GIVEN A primitive positive τ-sentence ϕ.
QUESTION Is $\Sigma \cup \{\phi\}$ satisfiable?

Combining this with the notion of a canonical query, from the previous statement, we may restate $\mathtt{CSP}(\Sigma)$ as the following problem:

GIVEN A finite τ-structure Δ.
QUESTION Is $\Sigma \cup \{Q(\Delta)\}$, where $Q(\Delta)$ is the canonical query of Δ, satisfiable?

With this in mind, we have:

Definition 3.2.4. Let Σ be a finite relational signature and T a τ-theory. We define $\mathtt{CSP}(T)$ as the set of all finite τ-structures Δ such that $\Sigma \cup \{Q(\Delta)\}$, where $Q(\Delta)$ is the canonical query of Δ, is satisfiable.

This definition is justified by the following proposition:

Proposition 3.2.4. *Let τ be a finite relational signature, Σ a satisfiable τ-theory and Γ a τ-structure. Then, the following are equivalent:*
1. $\mathtt{CSP}(\Gamma) = \mathtt{CSP}(\Sigma)$.
2. For every primitive positive τ-sentence ϕ we have that $\Sigma \models \neg\phi$ if, and only if $\Gamma \models \neg\phi$.

Proof. Suppose that $\mathtt{CSP}(\Gamma) = \mathtt{CSP}(\Sigma)$ and let ϕ be a primitive positive formula, such that $\Sigma \models \neg\phi$. Then, $\Sigma \cup \{\phi\}$ is not satisfiable, so $\phi \notin \mathtt{CSP}(\Sigma)$. In particular, we have that $\phi \notin \mathtt{CSP}(\Gamma)$ and so $\Gamma \not\models \phi$. It then follows that $\Gamma \models \neg\phi$, as required. Suppose, conversely, that for every primitive positive formula ϕ we have that $\Sigma \models \neg\phi$ if, and only if $\Gamma \models \neg\phi$. Suppose that $\phi \notin \mathtt{CSP}(\Sigma)$, then $\Sigma \cup \{\phi\}$ is not satisfiable and hence $\Sigma \models \neg\phi$. Thus, $\Gamma \models \neg\phi$ and hence $\phi \notin \mathtt{CSP}(\Gamma)$. The other inclusion is similar. □

Proposition 3.2.5. *Let Σ and Σ' be two satisfiable first-order τ-theories. Then the following are equivalent:*
1. $\mathtt{CSP}(\Sigma)$ and $\mathtt{CSP}(\Sigma')$ are the same computational problem.
2. For every model Δ of Σ there exists a model Δ' of Σ' and a homomorphism $f : \Delta \to \Delta'$.
3. For every existential positive sentence ϕ we have that $\Sigma \models \neg\phi$ if, and only if $\Sigma' \models \neg\phi$, i.e. the two theories entail the same negations of existential positive sentences.

Proof. Note that the equivalence of 1 and 3 is more or less immediate from the definition, as if $\mathtt{CSP}(\Sigma) = \mathtt{CSP}(\Sigma')$ then given primitive positive ϕ such that $\Sigma \models \neg\phi$ we have that $D(\phi) \notin \mathtt{CSP}(\Sigma)$, where $D(\phi)$ is the canonical database of ϕ. Hence $D(\phi) \notin \mathtt{CSP}(\Sigma')$ and thus $\Sigma' \models \neg\phi$. Conversely, it is enough to note that the negations of just the primitive positive formulas that a theory entails completely characterise the canonical databases of the formulas they entail, that is, they completely characterise their CSP.

We will now prove that 2 and 3 are also equivalent. Suppose that for Σ and Σ' we have that 2 holds. Let Δ be a model of Σ and ϕ be a primitive positive formula such that $\Delta \models \neg\phi$. Since homomorphisms preserve existential positive sentences and there is a homomorphism $f : \Delta \to \Delta'$, where Δ' is a model of Σ' it follows that $\Delta' \models \neg\phi$ and hence $\Sigma' \models \neg\phi$. Conversely, suppose that Σ and Σ' entail the same negations of primitive positive sentences. Let Δ be a model of Σ and let T be the first-order existential positive theory of Δ. Let $T' = \{\phi_1, \ldots, \phi_n\} \subseteq T$ be any finite subset of T and consider $\Sigma' \cup T'$. We claim that this is satisfiable. If not, then we would have that $\Sigma' \models \neg(\phi_1 \wedge \cdots \wedge \phi_n)$, but $(\phi_1 \wedge \cdots \wedge \phi_n)$ is an existential positive sentence, and hence $\Sigma \models \neg(\phi_1 \wedge \cdots \wedge \phi_n)$, which is impossible. By compactness, it follows that $\Sigma' \cup T$ is satisfiable, and hence it has a model, Δ'. It remains to show that there exists a homomorphism $f : \Delta \to \Delta'$. To construct this map enumerate each element in Δ. For each element in Δ there exists an element in Δ' that has the same positive existential type as that element, since Δ' is a model of T. In particular, we can assign each element of Δ to a corresponding element of the same existential positive type in Δ'. It is easy to see that the resulting map preserves atomic formulas, and is hence a homomorphism. □

We will now show that this unlike the previous two approaches, which are exactly equivalent, allowing us to "canonically" switch between the two, this is not the case anymore.

CSPs are essentially homoomrphism problems, and to this end we define the "homomorphism" version of JEP. A class of τ-structures $\mathcal{C}$ has the *Joint Homomorphism Property* (cf. JEP, from Definition 2.1.14) if for any $\Delta, \Delta' \in \mathcal{C}$ there exists a $\Gamma \in \mathcal{C}$ such that both Δ and Δ' homomorphically map to Γ.

Proposition 3.2.6. *Let Σ be a satisfiable τ-theory. Then the following are equivalent:*
1. *There exists some structure Γ such that CSP(Γ) and CSP(Σ) are the same.*
2. *For all primitive positive sentences ϕ_1 and ϕ_2 such that $\Sigma \cup \{\phi_1\}$ and $\Sigma \cup \{\phi_2\}$ are both satisfiable, $\Sigma \cup \{\phi_1, \phi_2\}$ is also satisfiable.*
3. *The class of models of Σ has the Joint Homomorphism Property.*

Proof. We will first show that 1 and 2 are equivalent. Suppose that for some structure Γ we have that $\texttt{CSP}(\Sigma)$ and $\texttt{CSP}(\Gamma)$ are the same problem. In particular, suppose that for some structure Γ we have that Σ and Γ entail the same negations of primitive positive formulas. Let ϕ_1 and ϕ_2 be primitive positive sentences and suppose that both $\Sigma \cup \{\phi_1\}$ and $\Sigma \cup \{\phi_2\}$ are satisfiable. Since Γ and Σ entail the same negations of primitive positive formulas and $\Sigma \not\models \neg\phi_i$, for $i = 1, 2$ we have that $\Gamma \not\models \neg\phi_i$ for $i = 1, 2$. In particular, we have that Γ satisfies both ϕ_1 and ϕ_2 and hence 2 follows.

Now, assume 2. We need to show that Σ has a model Γ such that Γ satisfies $\Sigma \cup \{\phi\}$ for any primitive positive formula $\phi \in \texttt{CSP}(\Sigma)$. By assumption, Σ is satisfiable and by 2 we note that every finite subset of $\Sigma \cup \texttt{CSP}(\Sigma)$ is satisfiable, since the conjunction of primitive positive formulas is a primitive positive formula. To make this more precise, let $\{\phi_1, \ldots, \phi_n\} \in \texttt{CSP}(\Sigma)$ we know that $\Sigma \cup \{\phi_1\}$ is satisfiable and that $\Sigma \cup \{\phi_2\}$ is satisfiable. Then, by 2, $\Sigma \cup \{\phi_1, \phi_2\}$ is satisfiable and in particular, $\Sigma \cup \{\phi_1 \wedge \phi_2\}$ is satisfiable. Repeating this n times gives us the required result. Then, by compactness we have that $\Sigma \cup \texttt{CSP}(\Sigma)$ has a model Γ. Then, clearly, Γ and Σ entail the negations of the same primitive positive formulas, so $\texttt{CSP}(\Sigma) = \texttt{CSP}(\Gamma)$.

We will now show that 2 and 3 are equivalent. The implication from 3 to 2 is immediate, since if models of Σ have the joint homomorphism property and Δ_1 is a model of $\Sigma \cup \{\phi_1\}$, Δ_2 a model of $\Sigma \cup \{\phi_2\}$ there exists a model Δ of Σ in which both Δ_1 and Δ_2 homomorphically map. Since homomorphisms preserve existential positive sentences we have that Δ satisfies both ϕ_1 and ϕ_2 and hence $\Sigma \cup \{\phi_1, \phi_2\}$ is satisfiable.

Finally, we need to show that 2 implies 3. This will be the hardest part of the proof. Let Δ_1 and Δ_2 be two models of Σ. Let τ' be an expansion of τ which is obtained by adding distinct constant symbols for each element of both Δ_1 and Δ_2 and let Γ_1 and Γ_2 be the τ'-expansions of Δ_1 and Δ_2, respectively (interpreting the constants for symbols from the other structure arbitrarily). We claim that the theory Σ' consisting of Σ together with the sets of primitive positive τ' sentences $\Sigma_1 = \{\phi \mid \Gamma_1 \models \phi\}$ and $\Sigma_2 = \{\phi \mid \Gamma_2 \models \phi\}$ is satisfiable. This follows from compactness and 2, since if T is any finite subset of Σ', then both $T_1 = T \cap \Sigma_1$ and $T_2 = T \cap \Sigma_2$ are logically equivalent to sentences ϕ_1 and ϕ_2, respectively. Note that both T_1 and T_2 only contain existential positive sentences, since these are all the τ'-sentences in Σ_i (for $i = 1, 2$) which contain the new constants that we have added. Then $\Gamma_1 \models \phi_1$ and $\Gamma_2 \models \phi_2$. It follows that $\Sigma \cup \{\phi_1, \phi_2\}$ is satisfiable, by assumption and by compactness we get that Σ' is satisfiable. Then, if we let Γ be a model of Σ' we see that there are homomorphisms from Γ_1 and Γ_2 to Γ (immediately by the diagram lemma, again see [4]) and if we let Δ be the τ-reduct of Γ, the result follows. □

A fairly easy example, which shows the proposition above in action is the following:

Example 3.2.4. Let τ be a signature with two binary relations E and N and let Σ be the theory:

$$\{(\forall x \forall y E(x, y)) \vee (\forall x \forall y N(x, y)), \forall x \forall y \neg(E(x, y) \wedge N(x, y))\}.$$

Let ϕ_1 be the sentence $\exists x E(x, x)$ and ϕ_2 the sentence $\exists x N(x, x)$. Note that then both $\Sigma \cup \{\phi_1\}$ and $\Sigma \cup \{\phi_2\}$ are satisfiable, but $\Sigma \cup \{\phi_1, \phi_2\}$ is not.

Before moving on to the Descriptive approach we give the following, negative example, which in general is neglected in the literature:

Example 3.2.5. Consider the constraint problem:

Given	A finite graph G.
Question	Is G a tournament?

Let $\mathcal{C}$ be the class of finite graphs that are tournaments and suppose that for some Γ we have that $\mathcal{C} = \texttt{CSP}(\Gamma)$. But, for any Γ, we have that the graph on two unconnected vertices homomorphically maps to Γ, so it is in $\texttt{CSP}(\Gamma)$. This is a contradiction, hence we cannot have that $\mathcal{C} = \texttt{CSP}(\Gamma)$ for any Γ. An alternative way of seeing this is that $\mathcal{C}$ is not closed under disjoint unions, and hence it is not possible to describe it at $\texttt{CSP}(\Gamma)$ for any Γ. The same argument shows that the following constraint problem is also not expressible as $\texttt{CSP}(\Gamma)$ for any Γ:

Given	A finite graph G.
Question	Is G connected?

The intuition behind this weakness in the expressibility the CSP formalism is that formalising constraints via homomorphisms does not allow us to impose the existence of relations in the instances, but just to characterise the relations that occur.

3.2.4 The Descriptive Approach

Finally, as we have seen we may express both the template and the instances of a CSP in terms of logic. In particular, one may ask how this connects with the theory of descriptive complexity.

Let us, first, define some important subsets of second order existential logic.

Definition 3.2.5. Let τ be a finite relational signature.

(i) We define **SNP** (for *Strict* **NP**) to be the class of all second order existential formulas where the first order part is universally quantified. That is, for a fixed signature τ a formula ϕ is an **SNP** formula if it of the form:

$$\exists \bar{X} \forall \bar{x} \left(\bigwedge_{i=1}^{n} \psi_i \right),$$

where $\psi_i(\bar{x})$ are quantifier-free first-order formulas with relation symbols from $\tau \cup \{\bar{X}\}$.

(ii) Let $\mathcal{C}$ be a class of finite τ-structures. Then we say that $\mathcal{C}$ is in **SNP** if there exists an **SNP** formula ϕ such that for any finite τ-structure Δ we have that $\Delta \in \mathcal{C}$ if, and only if $\Delta \vDash \phi$.

We also have the following syntactic restriction of **SNP**:

Definition 3.2.6. Again, let τ be a finite relational signature and let ϕ be an **SNP** sentence and write

$$\exists \bar{X} \forall \bar{x} \left(\bigwedge_{i=1}^{n} \psi_i \right)$$

where ψ_i are formulas with relation symbols from $\tau \cup \{\bar{X}\}$. Then ϕ is called *monotone* if for all i, each literal in the ψ_i with a relation symbol R, from τ is of the form $\neg R(\bar{x})$. We say that a monotone sentence is *without inequality* if we do not allow either equality or inequality relations in the first order part of ϕ.

The point of this definition lies on the distinction between the relation symbols from τ, which we call the *input relations* and the existentially quantified relations, which we refer to as the *existential relations*.

In particular, a formula is monotone if all of its conjuncts with input relations have the same *polarity*, that is, if they are all positive or all negative. Since we are discussing constraints we assume that the polarity is negative. The name monotone comes from the fact that if a structure Δ does not satisfy a monotone ϕ, then a structure Δ' which is the same as Δ, but with added elements in the input relations, will not satisfy ϕ, either.

We give an example, which will, as a matter of fact become important later:

Example 3.2.6. Let τ be a signature with a single binary relation symbol E and consider the τ-sentence:

$$\begin{aligned} \exists E' \forall x \forall y \forall z (& (\neg E(x,y) \vee E'(x,y)) \\ & \wedge (\neg E(x,y) \vee E'(y,x)) \\ & \wedge (\neg E'(x,y) \wedge \neg E'(x,z) \wedge \neg E'(y,z)) \\ & \wedge \neg (E'(x,y) \wedge E'(x,z) \wedge E'(y,z))). \end{aligned}$$

This is clearly in **SNP**. Moreover, it is not hard to see that this is monotone, since the input relations only appear negated. This sentence says that given a directed graph G neither G nor the complement of G contains an undirected triangle.

First, let us note that **SNP** is clearly a subset of **NP** (by Fagin's Theorem) but it has been shown, in [1] that every problem in **NP** is actually equivalent to a problem in **SNP**, under polynomial time reductions, i.e. **SNP**, as a class is as strong as all of **NP**. In particular, assuming that **P** $\neq$ **NP** and hence that **NP**-intermediate problems exist, **SNP** contains **NP**-intermediate problems.

We give another syntactic restriction of **NP**:

Definition 3.2.7. A subset of second order logic is called *monadic* if the second order quantification is over unary relations only.

Note that the sentence from example 3.2.6 is not monadic. Consider the following example:

Example 3.2.7. Let τ be a signature with a single binary relation E. Consider the sentence:

$$\exists C_1 C_2 C_3 \forall x \forall y \left(\bigvee_{i=1}^{3} C_i(x) \right) \wedge \left(\bigwedge_{1 \leq i < j \leq 3}^{3} \neg (C_i(x) \wedge C_j(x) \right) \\ \wedge \left(\bigwedge_{i=1}^{3} \neg E(x,y) \wedge \neg C_i(x) \vee \neg C_i(y) \right).$$

This sentence is very clearly monadic and G is a graph that satisfies it if, and only if it is 3-colourable. This example is in some sense canonical, since monadic second-order logic essentially allows us to "colour" the elements of a structure.

In fact, we provided above three potential syntactic restrictions of **NP**, namely monotonicity, monadicity and the possibility of allowing inequality in formulas. It is known that no two of these are essentially restrictions of **NP**, that is, we have the following, originally from [1] (Theorems 1, 2 and 3 in [1]) :

- Every problem in **NP** is polynomial time equivalent to a problem in monotone monadic **SNP** with inequality.
- Every problem in **NP** is polynomial time equivalent to a problem in monadic **SNP** without inequality.
- Every problem in **NP** is polynomial time equivalent to a problem in monotone **SNP** without inequality.

It is interesting then to examine the cases where all three of these restrictions are imposed at the same time. We state the following theorem, again from [1], without proof:

Theorem 3.2.1 (Theorem 6 in [1])**.** *Every problem in monotone monadic* **SNP** *without inequality is polynomial time equivalent to* CSP(Γ) *for some finite structure* Γ.

We stated the previous fact without proof, since it essentially constitutes a starting point for our exploration on CSPs. It should be, nevertheless, stressed that the proof requires non-trivial tools from Complexity Theory. In particular, the original proof, by Feder and Vardi used randomised Turing reductions and was eventually derandomised in [21].

We add another possible syntactic restriction of **SNP**. The point of this restrictions is that it gives us a syntactic characterisation of connected structures:

Definition 3.2.8. Let ϕ be a first-order τ-formula, written in prenex CNF, that is, ϕ is of the form:

$$\bar{Q}\bar{x} \left(\bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} \chi_{ij}(\bar{y}_{ij}) \right) \right),$$

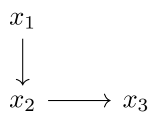
where where $\bar{Q}$ is a sequence of quantifiers, each χ_{ij} is a literal and $\bar{y}_{ij}$ are variables from $\bar{x}$. We define the *canonical database* of a clause ψ, of the form $\bigvee_{i=1}^{m} \chi_i(\bar{y_i})$ to be the structure $D(\psi)$ with domain $\bar{y}_i$ and relations $R^{D(\psi)}(x_1, \ldots, x_n)$ for each negative literal $\chi_i(x_1 \ldots, x_n)$ that appears in the clause. We say that ϕ is *connected* if the canonical database of each clause in ϕ is connected.

The following example illustrates this point:

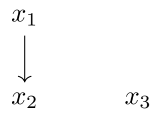
Example 3.2.8. Let ϕ be the following formula

$$\forall x_1, x_2, x_3(((E(x_1,x_2) \wedge E(x_2,x_3)) \to E(x_3,x_1)) \wedge ((E(x_1,x_2) \to E(x_2,x_3))),$$

then, the canonical database of the first clause above is:



which is connected. The canonical database of the second clause is:



which is not connected, so ϕ is not a connected formula.

Note that, the definition of the canonical database of a clause in a first-order formula does not seem to agree with the definition of the canonical database of a primitive positive formula. This is because the two notions are meant to encode inherently different ideas. The canonical database of a primitive positive formula is precisely the structure that this formula is trying to describe while the canonical database of a clause is meant to encode which variables appear inside the same relation symbol, in the formula.

When we discuss the canonical database of a formula, unless we explicitly state that the formula is primitive positive we will be working with the second definition. In the proof of Theorem 3.6.2 we will show how canonical databases defined as in Definition 3.2.8 agree with canonical databases of primitive positive formulas, when discussing potential forbidden structures.

Definition 3.2.9. Let ϕ be an **SNP** formula, whose first-order, quantifier part is written in CNF, that is, ϕ is:

$$\exists \bar{X} \forall \bar{x} \left(\bigwedge_{i=1}^{n} \psi_i \right),$$

where each $\psi_i(\bar{x})$ is a clause, i.e. it is of the form $\bigvee_{j=1}^{m_i} \chi_{ij}(\bar{x})$ for literals $\chi_{ij}(\bar{x})$. We say that ϕ is *connected* if its first-order part is connected.

Feder and Vardi also proved, in [1], that within **SNP** the classes of finite structures with the semantic property of being closed under inverse homomorphism are the same as these with the syntactic property of being captured by a monotone formula. That is:

Lemma 3.2.7. *Let ϕ be an* **SNP** *sentence. Then the class of structures that satisfy ϕ is closed under inverse homomorphisms if, and only if ϕ is equivalent to a monotone* **SNP** *sentence.*

Note that the proof of this is also extremely long and in it the distinction between input and existential relations that we mentioned previously becomes important.

In addition, we have that for classes of structures in **SNP** the semantic property of being closed under disjoint unions corresponds precisely to being captured by a connected formula. That is:

Lemma 3.2.8. *Let ϕ be an* **SNP** *sentence. The class of structures that satisfy ϕ is closed under disjoint unions if, and only if ϕ is equivalent to a connected* **SNP** *sentence.*

Proof. Fix a finite relational signature τ. Let ϕ be an **SNP** sentence and write:

$$\exists \bar{X} \forall \bar{x} \left(\bigwedge_{i=1}^{n} \psi_i \right),$$

where, as per the definition, each ψ_i is quantifier free in $\tau \cup \{\bar{X}\}$.

Suppose that ϕ is connected and let Δ_1 and Δ_2 be finite structures that satisfy ϕ. We want to show that the disjoint union $\Delta_1 \sqcup \Delta_2$ satisfies ϕ. We know that there exist first-order expansions Γ_1 and Γ_2 of Δ_1 and Δ_2 respectively that satisfy $\forall \bar{x} \left(\bigwedge \psi_i \right)$, the first-order part of ϕ. These expansions are given by introducing new relation symbols for the existential variables and defining the relations $\{X^{\Gamma_i}\}$ as in the satisfying second-order assignment. Suppose that the disjoint union of these expansions does not satisfy $\forall \bar{x} \left(\bigwedge \psi_i \right)$. By definition, there exists a clause ψ_i and elements $a_1, \ldots, a_n$ in $\Gamma_1 \sqcup \Gamma_2$ that do not satisfy ψ_i, for some i. This means that $\psi_i(a_1, \ldots, a_n)$ is false in $\Gamma_1 \sqcup \Gamma_2$. We know that both Γ_1 and Γ_2 make ψ_i true, so this means that the a_i cannot all come from the same component of the disjoint union. But then the canonical database for ψ_i cannot be connected since, if we write $\psi_i(a_1, \ldots, a_n)$ as:

$$\left(\bigwedge_{i=1}^{m} R_i(a_1, \ldots, a_{k_i}) \right) \rightarrow \left(\bigwedge_{i=1}^{m'} R'_i(a_1, \ldots, a_{j_i}) \right),$$

we have that some literal in the antecedent must only contain a_i's from Γ_1 and some only from Γ_2. This means, in particular, that ψ_i is disconnected, which is a contradiction.

Conversely, let $\mathcal{C}$ be the class of finite τ-structures that satisfy ϕ and suppose that $\mathcal{C}$ is closed under disjoint unions. We want to define a connected **SNP** formula χ that is equivalent to ϕ. Suppose, without loss, that $\bar{x} = (x_1, \ldots, x_b)$, where b is at least 3. If $b \leq 2$ then we may add up to three "dummy" variables to the original ϕ to ensure that this condition is met. We construct the sentence χ, which is of the form $\exists \bar{X} \exists E \forall \bar{x} \theta$ as follows:

(a) For each n-ary relation symbol $R \in \tau$ and each $i < j \leq n$ add the clause

$$R(x_1, \dots, x_n) \to E(x_i, x_j)$$

as a conjunct in θ.

(b) Add the clause $(E(x_1, x_2) \wedge E(x_2, x_3)) \to E(x_1, x_3)$ as a conjunct in θ.

(c) For each clause ψ_i of the original ϕ with variables $y_1, \dots, y_c$ from the x_i add the clause:

$$\left(\bigwedge_{i<j\leq c} E(y_i, y_j) \right) \to \psi_i$$

as a conjunct in θ.

We claim that θ is connected and that ϕ and χ are logically equivalent.

For the first claim, note that by (a) we have constructed θ is such a way so that whenever a pair of variables appears in the same relation then there is an "edge" from one to the other in the canonical database of θ, when we have relations of arity more than 2. With the second and third conditions we have ensured that there are edges between x_1, x_2 and x_3. This takes care of the "edges" connecting any two variables that appear in the same relation, when our signature has only unary and binary relations. In this case we can still define a connected formula, but all the connections are added artificially, "after the fact". Moreover, (b) and (c) ensure that the relation is symmetric and transitive (recall that θ is universally quantified).

The proof of the second claim is more intricate. We need to show that any τ-structure satisfies ϕ if, and only if it satisfies χ. On the one hand if Δ is a finite τ-structure that satisfies ϕ then define E to be $\text{dom}(\Delta) \times \text{dom}(\Delta)$. In particular, then, it is easy to see that all conjuncts in θ constructed as in $(a) - (c)$ are satisfied immediately. Moreover, any conjunct constructed as in (d) will be satisfied, since $\Delta \models \phi$ and hence an expansion of Δ satisfies each ψ_i.

On the other hand, if Δ is a finite τ-structure that satisfies χ, then by definition there is a $(\tau \cup \{\bar{X}, E\})$-expansion Γ of Δ that satisfies $\forall \bar{x} \theta$. We can write Γ as a disjoint union $\Gamma_1 \sqcup \cdots \sqcup \Gamma_d$ where each Γ_j is connected, that is, any pair of distinct elements of Γ_j appears in the same relation R or E. Note that it is necessary that $d \leq b$, since "at worst" there are no new edges E added except from the ones from (b) and (c) (which is the case when all relations are unary). We claim that $E = \text{dom}(\Gamma_j) \times \text{dom}(\Gamma_j) \setminus \{(a,a) \,|\, a \in \text{dom}(\Gamma_j)\}$ for each j. Of course, by (c) if $E(a,b)$ holds then $E(b,a)$ holds in Γ_j (for $a, b \in \Gamma_j$). Moreover, if two elements are in the same disjoint component then either they appear in the same relation from τ and hence by (a) there is an edge E connecting them or there was an edge connecting them in the first place. But, by construction, E is transitive and hence within connected structures Γ_j we have that $E = \text{dom}(\Gamma_j) \times \text{dom}(\Gamma_j) \setminus \{(a,a) \,|\, a \in \text{dom}(\Gamma_j)\}$, as claimed. Then each Γ_j must satisfy ψ_i for all i, otherwise the conjuncts we constructed in (d) above would not be satisfied.

Since the class of finite structures that satisfy ϕ is closed under disjoint unions, Δ (viewed as the τ-reduct of Γ) must satisfy ϕ and hence ϕ and χ are logically equivalent. ☐

We observe that the connected formula χ we built above is in fact monotone, without inequality. Combining this with the previous lemma and the fact that $\mathcal{C} = \texttt{CSP}(\Gamma)$, if, and only if $\mathcal{C}$ is closed under disjoint unions and inverse homomorphisms gives us:

Theorem 3.2.2. *An* **SNP** *sentence ϕ describes a problem of the form CSP(Γ) if, and only if ϕ is equivalent to a monotone and connected* **SNP** *sentence without inequality.*

We have a similar result for monadic **SNP** sentences. In particular:

Theorem 3.2.3. *A monadic* **SNP** *sentence ϕ describes a problem of the form CSP(Γ) if, and only if it is equivalent to a monotone connected monadic* **SNP** *sentence without inequality.*

Proof. The structure of the proof is identical. We have a criterion for inverse homomorphisms from a result similar to Lemma 3.2.7, for monadic sentences, which we again quote, without proof from []. For disjoint unions, one direction is identical to the proof of Lemma 3.2.8, but the other direction still needs some work (in the proof of Lemma 3.2.8 we defined a binary relation E, which would break monadicity).

We want to prove that if ϕ is a monadic **SNP** sentence that describes a class $\mathcal{C}$ of finite τ structures that is closed under disjoint unions then ϕ is equivalent to a monadic connected **SNP** sentence (without adding conjuncts with positive input literals, i.e. without breaking monotonicity or introducing inequalities). Without loss, suppose that ϕ is minimal, in terms of the number of literals it contains. That is, if we remove any literal from ϕ then we end up with a formula ϕ' that no longer describes $\mathcal{C}$.

Assume, for a contradiction that ϕ is not connected and let ψ be a first order clause of ϕ that is not connected, that is, we can write ψ as $\psi_1 \vee \psi_2$ where the variables appearing in ψ_1 and ψ_2 are distinct. Then ϕ_1 and ϕ_2, obtained by replacing ψ_i by ψ_1 and ψ_2, respectively. In particular, for $i = 1, 2$ we have that ϕ_i has fewer literals than ϕ and hence, by minimality they are no longer equivalent. Therefore, there exist finite structures Δ_1 and Δ_2 which do not satisfy ϕ_1 and ϕ_2, respectively but both Δ_1 and Δ_2 are models for ϕ. By assumption, then $\Delta = \Delta_1 \sqcup \Delta_2$ satisfies ϕ. Let Γ be the expansion of Δ that satisfies the first-order part of ϕ. In particular, Γ and Δ only differ by unary relations. Let Γ_1 and Γ_2 be the substructures induced by $\mathrm{dom}(\Delta_1)$ and $\mathrm{dom}(\Delta_2)$ in Γ, respectively. Since Γ and Δ only differ by unary relations, if Γ_1 satisfied ψ_1 then Δ_1 would satisfy ϕ_1 and similarly for Γ_2. Clearly then, it is not possible that $\Gamma_1 \sqcup \Gamma_2$ satisfies ϕ, which is a contradiction. □

We should note that the argument above would not have worked in the non-monadic case. The point is that if we had introduced existential relations of arity more than 1 then Γ would not be necessarily equal to the disjoint union of Γ_1 and Γ_2 (as the relations we would have added may have had elements from both of them).

In particular, this result is a subtle restatement of our previous theorem, which stated that every problem in monotone monadic **SNP** without inequality is polynomially equivalent to `CSP`(Γ) for some finite structure Γ. What we have managed to give is a precise characterisation of essentially the criteria needed for a converse direction. Indeed, we have proved that given a problem in monadic **SNP** that problem is a CSP precisely if there exists a formula ϕ with some well-defined syntactic restrictions that describes it.

3.3 The Classification of CSPs

At this point, we have essentially defined enough machinery to give some interesting results about Constraint Satisfaction Problems.

We start with a couple of very core results on this direction.

Theorem 3.3.1. *If Γ is finite, then `CSP`(Γ) is contained in* **NP**.

Proof. This is an easy "guess and check" argument. Indeed, if Γ is finite then given any finite Δ in the same signature, then we can guess a mapping $f : \mathrm{dom}(\Delta) \to \mathrm{dom}(\Gamma)$ and check that it is a homomorphism. It is clear that the size of the guess is bounded by the domain of Δ (which would be the case regardless of the size of Γ) but the point is that if Γ is finite we can check whether or not f is a homomorphism in polynomial time, in the size of Δ.

Indeed, the check works as follows:

- For each n-ary relation R in the signature of Γ, if we have $R^{\Delta}(a_1, \ldots, a_n)$ we need to check if $R^{\Gamma}(f(a_1), \ldots, f(a_n))$.
- Since Γ has finitely many elements, R^{Γ} has at most $|\Gamma|^n$ elements, and the check is linear in the size of Δ, for each such relation.

In the end, we are polynomially bounded in the size of Δ (as a structure). So `CSP`(Γ) is in **NP**. □

In fact, we can say something even stronger, but before we can state the theorem we need to introduce one more term, the terminology is from [8]:

Definition 3.3.1. Let τ be a finite relational signature and let $\mathcal{C}$ be a class of finite τ-structures. We say that $\mathcal{C}$ is *finitely constrained* or *finitely bounded* if $\mathcal{C} = \mathrm{Forb}(\mathcal{N})$, for some finite set of τ-structures $\mathcal{N}$. We say that a τ-structure Γ is *finitely constrained* or *finitely bounded* if $\mathcal{C} = \mathrm{Age}(\Gamma)$ is finitely bounded.

As a matter of fact, just by definition we know that $\text{Age}(\Gamma) \subseteq \texttt{CSP}(\Gamma)$ (since the age of a structure contains all its finite *induced* substructures and hence if $\Delta \in \text{Age}(\Gamma)$ then the inclusion map $f : \Delta \to \Gamma$ is an embedding and, in particular, a homomorphism). This gives us the following result:

Theorem 3.3.2. *If Γ is finitely bounded then $\texttt{CSP}(\Gamma)$ is in* **NP**.

Proof. We may argue about this indirectly. Since Γ is finitely bounded, we have that $\text{Age}(\Gamma) = \text{Forb}(\mathcal{N})$, where $\mathcal{N} = \{\Delta_1, \ldots, \Delta_n\}$. For each Δ_i let ψ_i be the canonical query of Δ. Then, since each ψ_i is primitive positive, $\neg\psi_i$ is a universal first-order sentence that is monotone (all of the conjuncts are negated), by definition.

Define ϕ to be the (monotone) **SNP** sentence:

$$\exists R'_1, \ldots R'_m \forall \bar{x} \left(\left(\bigwedge_{i=1}^{m} R_i(\bar{x}) \to R'_i(\bar{x}) \right) \wedge \left(\bigwedge_{i=1}^{n} \psi'_i \right) \right)$$

where $\tau = \{R_1, \ldots, R_m\}$ and ψ'_i is the quantifier-free part of ψ_i, with all relation symbols from τ replaced by their corresponding prime versions.

Then, a structure Δ satisfies ϕ if, and only if there exist relations R'_i extending the relations on Δ such that, with the expanded relations we have that none of the forbidden substructures occur in Δ, that is, the $(\tau \cup \{R'_i\})$-expansion of Δ that satisfies the first-order part of ϕ is a substructure of Γ. We therefore have that $\Delta \in \texttt{CSP}(\Gamma)$ if, and only if $\Delta \vDash \phi$ and hence $\texttt{CSP}(\Gamma)$ is in **SNP**. The result is then immediate. □

In [7] the definition of a finitely constrained structure is slightly different, in particular, a τ-structure Γ is said to be finitely constrained if there exists a τ'-expansion Γ' of Γ such that $\text{Age}(\Gamma) = \text{Forb}(\mathcal{N})$ for a finite set of finite τ' structures. The proof of Theorem 3.3.2 justifies why the two definitions can be used interchangeably. The reason why we mention this is that taking the second definition allows us to give a more "classical complexity" type of proof of Theorem 3.3.2, using a "guess and check" argument:

Second proof of 3.3.2. In an expanded signature τ' we have that $\text{Age}(\Gamma') = \text{Forb}(\mathcal{N})$, for some finite set of finite structures $\mathcal{N}$. Then given an instance Δ of $\texttt{CSP}(\Gamma)$ we guess a τ'-expansion of Δ and a map $f : \Delta' \to \Gamma'$. We then check if this is a homomorphism. This time, Γ' is allowed to be infinite, but for each element in the image of f we only need to check if it breaks any of the constraints imposed by $\mathcal{N}$, which is a linear problem. Thus $\texttt{CSP}(\Gamma)$ is in **NP**. □

Even better, we can extend the "guess and check" argument above to say that if Γ is *finitely axiomatisable* then $\texttt{CSP}(\Gamma)$ is in **NP** (we just need to check if the finitely many axioms are satisfied the images in the map that we guessed).

The obvious question, then, is, what can we say about $\texttt{CSP}(\Gamma)$ for an arbitrary infinite structure Γ?

The answer is: Not much! In general it is not the case that $\texttt{CSP}(\Gamma)$ is always in **NP**. The classical example is $\texttt{CSP}((\mathbb{N}, 0, 1, +, \times))$ (which is equivalent to non-linear constraints). This reduces to Hilbert's 10-th problem which famously is undecidable.

As we mentioned previously, the basis of the discussion of CSPs, from a complexity theory point of view, is that large classes of them are conjectured to display a dichotomy. The first known example is the following:

Theorem 3.3.3. *Let Γ be a template with $|\Gamma| = 2$. Then $\texttt{CSP}(\Gamma)$ is either in* **P** *or* **NP***-complete.*

The idea is that given that the domain has just two elements we can identify them with 0 and 1 and then $\texttt{CSP}(\Gamma)$ becomes equivalent to either 2-Sat or 3-Sat. Of course, this is a really deep result, despite its apparent simplicity and the proof of it is out of the scope of this report.

Finally, just two years ago the big "Dichotomy Conjecture" of Feder and Vardi was finally proved, and hence it is indeed a fact that that:

Theorem 3.3.4. *If Γ is a finite τ-structure then $\texttt{CSP}(\Gamma)$ is either in* **P** *or it is* **NP***-complete.*

Given that, it makes sense to try to systematically discuss CSP(Γ) when Γ is infinite. As we saw, this is indeed a proper superclass of CSP(Γ) for finite structures Γ and it contains interesting problems. Since we have already shown that if Γ is arbitrary then there is not much we can say about its CSP, we will be focusing our attention on a specific class of infinite structures Γ. This will be the focus of the next section.

3.4 ω-categorical Templates

Recall that a structure is called ω-categorical if all countable models of its first-order theory are isomorphic.

In the previous section we discussed classes of CSPs that are in **NP**. One would hope that if Γ is an infinite ω-categorical structure then CSP(Γ) is in **NP**. Some motivation behind this is given by the following proposition:

Proposition 3.4.1. *Let Γ be a an ω-categorical template. Then for any countable relational structure Γ', we have that CSP(Γ') $\subseteq$ CSP(Γ) if, and only if there exists a homomorphism $f : \Gamma' \to \Gamma$.*

Proof. Note that one direction of this is obvious, since if $f : \Gamma' \to \Gamma$ is a homomorphism then any structure that homomorphically maps to Γ will homomorphically map to Γ', and hence CSP(Γ') $\subseteq$ CSP(Γ).

The converse direction is significantly less trivial. Indeed, it makes use of both Ryll-Nardzewski and König's Lemma (which states that any finitely branching infinite tree contains an infinite path). Suppose that CSP(Γ') $\subseteq$ CSP(Γ). This means that given a finite τ-structure Δ for which that there exists $g : \Delta \to \Gamma'$ we can find a homomorphism $f : \Delta \to \Gamma$. In particular, for any finite weak substructure of Γ' we can find a homomorphism from that substructure to Γ. We need to find a homomorphism from Γ' to Γ.

We will use Ryll-Nardzewski to construct a tree that will allow us to apply König's Lemma. We proceed as follows. Let $a_1, a_2, \dots$ be an enumeration of Γ'. At the n-th level of the tree the nodes correspond to the equivalence classes of homomorphisms from $a_1, \dots, a_n$ to Γ, where two homomorphisms are equivalent if, and only if they differ by an automorphism of Γ. We define adjacency between nodes by restriction, that is, the child node of a node on the n-th level given by the map f_n is a node on the $(n+1)$-st level which corresponds to an equivalence class of homomorphisms f_{n+1} that when restricted to the first n elements are equal to f_n. By Ryll-Nardzewski there are only finitely many nodes at each level. But then the tree indeed satisfies the assumptions of König's Lemma and hence must contain an infinite path. This path is clearly the desired homomrphism. □

This property which is enjoyed when Γ and Γ' are both finite remains true in the ω-categorical setting and is not true for arbitrary infinite Γ. We also have the following corollary of Theorem 2.1.12:

Corollary 3.4.2. *Let τ, τ' be countable relational signatures, Γ an ω-categorical τ-structure and Δ τ'-structure with dom(Δ) = dom(Γ). Suppose that Δ is first-order definable from Γ. Then Δ is ω-categorical.*

Proof. Let $\Sigma = \mathrm{Th}(\Gamma)$ and $T = \mathrm{Th}(\Delta)$. Note that if $\phi(\bar{x})$ is a τ'-formula then by rewriting each τ'-relation as a primitive positive τ-formula we can construct a τ-formula $\psi(\bar{x})$ such that for any n-tuple $\bar{a}$ from Γ we have that $\Sigma \vDash \psi(\bar{a})$ if, and only if $T \vDash \phi(\bar{a})$. In particular, since $\mathrm{Th}(\Gamma)$ is ω-categorical, for each $n \in \mathbb{N}$ there exist finitely many τ-formulas in n variables that are pairwise non-equivalent modulo Σ, and hence there must, for each exist $n \in \mathbb{N}$ finitely many τ'-formulas in n variables that are pairwise non-equivalent modulo T. It follows, by Theorem 2.1.12 that Δ is ω-categorical. □

A very similar, but slightly more technical argument gives the same result for structures that are first-order interpretable in an ω-categorical structure (as defined in Section 2.1). Similarly, we can argue that if $\tau \subseteq \tau'$ are finite relational signatures and Γ' is an ω-categorical τ'-structure, then Γ, the τ-reduct of Γ' is ω-categorical, also.

However, despite our best hopes, the class of CSPs of the form CSP(Γ) where Γ is any structure is certainly not a dichotomous subclass of **NP** and that is because it is not a subclass of **NP**. In fact, we have more or less developed enough tools to show this.

As we mentioned in the previous chapter there are uncountably many classes of homogeneous tournaments which have the free amalgamation property. These classes all have a Fraïssé Limit, which is a countable homogeneous digraph, and hence it is ω-categorical. Note that we have the following results:

Proposition 3.4.3. *Let τ be a finite relational signature and Γ a τ-structure such that Age(Γ) has the strong amalgamation property. Define $\mathcal{C}$ to be the following class of finite τ-structures:*

$$\mathcal{C} = \{\Delta \mid dom(\Delta) \subseteq dom(\Gamma) \text{ and } \iota : \Delta \to \Gamma \text{ is an injective homomorphism}\},$$

where ι is the inclusion map. Then $\mathcal{C} = \texttt{CSP}(\Gamma)$.

Proof. It is clear that $\mathcal{C} \subseteq \texttt{CSP}(\Gamma)$, since if $\Delta \in \mathcal{C}$ the inclusion map is a homomorphism, so $\Delta \in \texttt{CSP}(\Gamma)$.

For the converse, let $\Delta \in \texttt{CSP}(\Gamma)$ and $f : \Delta \to \Gamma$ be a homomorphism. It suffices to show that we can find an injective homomorphism $g : \Delta \to \Gamma$, since then we will be able to identify each $v \in \text{dom}(\Delta)$ with an element of $\text{dom}(\Gamma)$ and since g is also a homomorphism, we have that $\Delta \in \mathcal{C}$. Suppose that there exist $u, v \in \text{dom}(\Delta)$ such that $f(v) = f(u) = p \in \Gamma$. We want to find an element $p' \in \Gamma \setminus \{p\}$, which has the same type as p over $f(\Delta)$.

Let $A = f(\Delta) \setminus \{p\}$, $A_1 = A \cup \{p\} = f(\Delta)$ and let $A_2 \in \mathcal{C}$ be an isomorphic copy of A_1. We can find such a structure since $\mathcal{C}$ is closed under taking isomorphic copies. Since Age(Γ) has the strong amalgamation property we take the amalgam of A_1 and A_2, over A and no new identifications are made. This is a substructure of Γ that contains p and another point $\{p'\} = A_2 \setminus A$ (by the strong amalgamation property p and p' are distinct, since no elements outside A are identified with elements of either A_1 and A_2). Then, since A_1 and A_2 were isomorphic p' will have the same type over A as p and setting $f(v) = p'$ does not change any of the relations in $f(\Delta)$. Since Δ is finite we can repeat this for all elements in Δ that f sends to the same place, and in the end we obtain an injective homomorphism, as required. □

The next step is to examine what happens to $\texttt{CSP}(\Gamma)$ when Γ is a digraph and the age of Γ has the free amalgamation property. We have that:

Proposition 3.4.4. *Let τ be a signature with a single binary relation symbol E and let Γ be a τ-structure. If the age of Γ has the free amalgamation property then Age(Γ) = $\texttt{CSP}(\Gamma)$.*

Proof. Note that as we have seen, free amalgamation implies strong amalgamation and hence $\texttt{CSP}(\Gamma)$ is equal to the class $\mathcal{C}$ as defined in Proposition 3.4.3.

We know, by definition, that Age(Γ) $\subseteq \mathcal{C}$ so we need to show that $\mathcal{C} \subseteq$ Age(Γ). We will do this by induction on the number of vertices of $G \in \mathcal{C}$. Clearly, if G has a single vertex then it is in Age(Γ). Suppose that the result holds for all structures with $n-1$ vertices. We will show that any structure on n vertices that homomorphically maps into Γ is in fact in $\mathcal{C}$.

Let G be a structure on n vertices that injects into some substructure of Γ and let $G' \in$ Age(Γ) be the substructure in Γ that G injects to, of minimal size (i.e. G' has n vertices). We will show that thanks to free amalgamation and the fact that τ has a single binary relation, all graphs on n nodes with that embed into Γ are actually in Age(Γ), i.e. that we can remove any edge of G' and that the remaining structure will be in Age(Γ). Let $E(v_1, v_2)$ be any edge in G' and define H to be the structure obtained by removing $E(v_1, v_2)$ from G'. We note that by assumption both G with v_1 removed and G with v_2 removed are in the age of Γ and since we can freely amalgamate these two structures to get H we have that $H \in$ Age(Γ). We can repeat this argument to show that we can remove any number of edges from G' to get a structure in on n nodes in Age(Γ) and hence the result follows. □

Combining the above with Henson's result, the following is almost immediate:

Theorem 3.4.1. *There exists a homogeneous digraph Γ such that $\texttt{CSP}(\Gamma)$ is undecidable.*

Proof. The countability argument in play here is classical. There are uncountably many digraphs Γ which have distinct ages and free amalgamation, as we saw in Section 2.1.10 It follows, by the previous proposition that there are uncountably many distinct computational problems of the form $\texttt{CSP}(\Gamma)$, when Γ is a homogeneous digraph, but since there are only countably many Turing Machines and an undecidable CSP must occur, amongst them. □

As a matter of fact, there exist uncountably many homogeneous digraphs Γ such that $\texttt{CSP}(\Gamma)$ is undecidable. We note, though that when Γ is finitely bounded (so when $\Gamma = \text{Forb}(\mathcal{T})$, and $\mathcal{T}$ is finite) then $\texttt{CSP}(\Gamma)$ is always in **NP** and hence we will focus our attention to $\texttt{CSP}(\Gamma)$ where Γ is a finitely bounded ω-categorical structure. When we say $\texttt{CSP}(\Gamma)$ for an ω-categorical Γ, we will, unless otherwise stated, mean that Γ is also finitely bounded.

As we stated, in the previous section CSPs on finite domains display a much desired dichotomy. It is conjectured that this remains true for CSPs on ω-categorical templates. Note, first of all, that the class of CSPs on ω-categorical templates is a strictly larger class that the class of CSPs on finite templates. Without the proper terminology, we showed this in Example 3.2.2.

Note that discussing ω-categorical CSPs combines both the homomorphism and the satisfiability approach, since, given $\mathtt{CSP}(\Gamma)$ we are at the same time discussing both the theory of Γ and the maps to Γ. Ryll-Nardzewski, as we saw in the proof of Proposition 3.4.1 serves as a great tool for discussing such CSPs.

The following theorem shows how the first three approaches we described interact when discussing ω-categorical structures:

Theorem 3.4.2. *Let τ be a finite relational signature and Γ a τ-structure. Then, the following are equivalent:*
1. *There exists an ω-categorical Δ such that $\mathit{CSP}(\Gamma) = \mathit{CSP}(\Delta)$.*
2. *There exists a structure Δ' such that $\mathit{CSP}(\Delta') = \mathit{CSP}(\Gamma)$ and for all $n \geq 1$ we have that Δ' has finitely many primitive positive definable relations of arity n.*

Proof. We have more or less developed enough theory to prove this theorem, but the proof is somewhat long and quite technical. For the proof of this we refer the reader to Theorem 3.6.21 and Corollary 3.6.22 in [8]. □

The proof of the theorem above, more or less immediately implies the following concise characterisation of CSPs on ω-categorical templates:

Theorem 3.4.3. *Let Σ be a complete primitive positive τ-theory. Then the following are equivalent:*
1. *Σ has a finite or ω-categrical model.*
2. *Σ has finitely many maximal primitive positive n-types for each $n \in \mathbb{N}$.*

Moreover, we have the following result, from [22] and [23], which we also state, without proof:

Theorem 3.4.4. *Let $\mathcal{N}$ be a finite set of finite connected τ-structures. Then there exists a countable ω-categorical structure Γ such that $\mathit{Forb}(\mathcal{N}) \subseteq \mathit{Age}(\Gamma)$, and in particular, $\mathit{Age}(\Gamma) = \mathit{wForb}(\mathcal{N})$. Moreover Γ can be expanded by finitely many primitive positive relations so that the resulting structure is homogeneous.*

Recall that Definition 2.1.18 said that a τ-structure Γ is universal if for every countable structure Δ such that $\mathrm{Age}(\Delta) \subseteq \mathrm{Age}(\Gamma)$ there exists an embedding $f : \Delta \to \Gamma$. If Δ is any finite $\mathcal{N}$-free structure then it is easy to see that the theorem above guarantees that such an embedding exists, and hence Γ is universal for the class of finite $\mathcal{N}$-free structures. Recall, also, from Proposition 3.2.1 that a class $\mathcal{C}$ is of the form $\mathtt{CSP}(\Gamma)$, for some Γ if, and only if $\mathcal{C} = \mathrm{wForb}(\mathcal{N})$, where $\mathcal{N}$ is a set of finite connected structures. Then, if that class is in addition finite, Theorem 3.4.4 tells us that we can find an ω-categorical structure Γ such that $\mathtt{CSP}(\Gamma) = \mathcal{C}$.

3.5 Examples of CSPs

So far, we have mostly examined CSPs from a purely theoretical point of view, but one should not forget that these problems are computational problems, and in that regard it is always helpful to look at some of the objects we are interested in, in a less abstract way.

3.5.1 Countable Homogeneous Tournaments

We start by recalling, the deep result by Lachlan mentioned in the previous section about the classification of countable homogeneous tournaments. In particular, up to isomorphism, these are

1. The Dense Linear Order with no endpoints $\mathbb{Q}$.
2. The Dense Local order with no endpoints $S(2)$.
3. The universal tournament T^{∞}.

We will use them here as examples as we have already spent quite some time developing their theory and they serve as a nice example of a few of the ideas that we would like to illustrate here.

We already discussed $\mathtt{CSP}((\mathbb{Q}, \leq))$ in Example 3.2.2 and showed that this is equivalent to the problem Acyclic which we know is in **P**.

As for T^∞, which is the Fraïssé Limit of the class of all finite tournaments, any graph G homomorphically maps into T^∞, trivially and hence CSP(T^∞) is also in **P**, in fact it is decided by the Turing Machine that accepts all strings.

What remains is CSP($S(2)$). We will show that this is **NP**-complete, by reducing to it the following problem, called Betweenness:

Given — A finite set V and a finite collection $\mathcal{C}$ of ordered triples $\{(x_i, y_i, z_i)\}_{i\in I} \subseteq V \times V \times V$ of distinct elements.

Question — Is there a set-theoretic bijection $f : V \to |V|$ such that for each $i \in I$, given the triple $(x_i, y_i, z_i) \in \mathcal{C}$ we have that either $f(x_i) < f(y_i) < f(z_i)$ or $f(z_i) < f(y_i) < f(x_i)$.

Intuitively, what this problem asks is that we arrange the triples in a way that the *betweenness* relation is satisfied. This problem is clearly in **NP** and it has been shown (by reducing it to 3Sat) that it is actually **NP**-complete, in [24]. We claim that Betweenness reduces to CSP($S(2)$) in **P**-time and is hence **NP**-complete (it is **NP**-hard, since $S(2)$ is finitely bounded). Given an instance $V, \mathcal{C}$ of Betweenness we need to define an instance G of CSP($S(2)$) such that $(V, \mathcal{C})$ is a Yes instance if, and only if S is a Yes instance of CSP($S(2)$).

We will use $a < b$ and $a \to b$ interchangeably, to indicate that there is an edge from a to b. Consider the construction of G, as follows:

- The vertices of G are the same as the vertices of V together with a new "minimum" vertex u, that is $u \to x$ (i.e. $u < x$) for all $x \in V$.
- For each triple $(x_i, y_i, z_i) \in \mathcal{C}$ we add two new vertices v_i and w_i which we connect to our graph as follows:
  - $x_i \to v_i$, and $w_i \to x_i$.
  - $y_i \to v_i$, and $y_i \to w_i$.
  - $v_i \to z_i$, and $z_i \to w_i$.

To make sense of this mapping will use the characterisation of $S(2)$ as the Fraaïssé Limit of Forb($\{[I_1, C_3], [C_3, I_1]\}$) and as the Dense Linear Order, which is two copies of $\mathbb{Q}$ such that if x, y are in different copies then $x \to y$ if, and only if $y < x$, as we did in Section 2.1.10.

Suppose, first, that there is a solution $f : V \to |V|$ for $(V, \mathcal{C})$. We need to show that this allows us to construct a homomorphism $f : G \to S(2)$. We start off by mapping u, the "minimum" in G to any point of $S(2)$. Then we consider the subset of $S(2)$ given by $X = \{x \in S(2) \mid f(u) < x\}$. This is a linear order, isomorphic to a subset of either Q_1 or Q_2. In particular, our solution for Betweenness, $f : V \to |V|$, gives us a way of homomorphically mapping the elements of $V \subseteq G$ in X, as we have that $u \to x$ for each $x \in G$ that was originally in V. So, ensure that the v_i's and the w_i's we introduced can be homomorphically mapped as well. If $x_i < y_i < z_i$ in the solution f then we can map v_i in the same linear order as x_i, y_i and z_i, sitting between y_i and z_i and we can map w_i in the other copy, sitting between x_i and y_i. If $z_i < y_i < y_i$ then we do the opposite. In the end, we have constructed a well-defined homomorphism from G to $S(2)$, as required.

We covered the "easy" direction. Suppose, now that there is a homomorphism $f : G \to S(2)$. Note that since we have imposed that $u \to x$ for all $x \in V$ it follows that f maps all vertices of G that were in V in a linear order. We need to show that in this linear order, whenever we had a triplet $(x_i, y_i, z_i) \in \mathcal{C}$ then the betweenness relation is preserved. Suppose not. Then there are four possible cases to consider:

- $y_i < x_i$ and $y_i < z_i$. Since the three elements lie on the same linear order there are two cases to consider:
  - $x_i < z_i$. In this case we have that there is a cycle $x_i \to z_i \to w_i \to x_i$ and edges from y_i to all three vertices of the cycle. This is $[I_1, C_3]$, so this does not happen.
  - $z_i < x_i$.In this case we have that there is a cycle $x_i \to v_i \to z_i \to x_i$ and edges from y_i to all three vertices of the cycle. This is, again $[I_1, C_3]$, so this does not happen.
- $x_i < y_i$ and $z_i < y_i$. Again the three elements lie on the same linear order and we have that:
  - $x_i < z_i$. In this case we have that there is a cycle $x_i \to z_i \to w_i \to x_i$ and edges from the vertices of the cycle to y_i. This is $[C_3, I_1]$, so this does not happen.
  - $z_i < x_i$.In this case we have that there is a cycle $x_i \to v_i \to z_i \to x_i$ and again there are edges from the vertices of the cycle to y_i. This is, again $[C_3, I_1]$, so this does not happen.

It then follows that f gives us a way of solving Betweenness, as we pick out the order of the elements of V given by f. Hence CSP($S(2)$) is **NP**-complete, as claimed.

In summary we have seen that when Γ is a countable homogeneous tournament, then $\texttt{CSP}(\Gamma)$ is either in **P** or it is **NP**-complete.

3.5.2 CSPs of Fraïssé Limits

We have already touched on CSPs of the form $\texttt{CSP}(\Gamma)$ where Γ is the Fraïssé Limit of some class of finite structures $\mathcal{C}$, in the sense of Theorem 2.1.8, when we proved that there exist ω-categorical CSPs (that is, CSPs on ω-categorical templates) that are undecidable. In this section we will consider what happens when we try to talk about finitely bounded classes of tournaments.

In particular, let $\mathcal{N}$ be $\{I_1, I_2, C_2, G_1, \dots, G_n\}$, where G_i are arbitrary graphs, other than C_3, $[I_1, C_3]$ and $[C_3, I_1]$ and $n \geq 1$. Then we have that Forb($\mathcal{N}$) does not contain any graphs that have independent nodes and it does not contain any 2-cycles. In particular, all the graphs in Forb($\mathcal{N}$) are tournaments. Suppose that there is some countable Γ such that Age(Γ) = Forb($\mathcal{N}$). We know that then Γ is a Fraïssé Limit, and hence it is countable and homogeneous. In particular, Γ must be one of the tournaments we discussed above. Since this is impossible, it follows that Forb($\mathcal{N}$) is not an amalgamation class. The point is, that in this case $\mathcal{N}$ is contains disconnected graphs, and hence Forb($\mathcal{N}$) is not closed under disjoint unions. Therefore, there is not much we can say about the corresponding computational problem of deciding wether a graph G homomorphically maps to some graph in Forb($\mathcal{N}$, in therms of CSPs (other than the fact that it is in **NP**, by 3.3.2).

Note that Theorem 3.4.4 gives us a very powerful tool in the construction of CSPs, since for any finite set of finite connected τ-structures, $\mathcal{N}$, we can always find a structure Γ which has $\texttt{CSP}(\Gamma)$ = wForb($\mathcal{N}$). Of course, this does not apply in the case discussed above, as $\mathcal{N}$ contains disconnected structures. We mention it here, though, since the construction of Γ with this property in [23] uses Fraïssé limits (in a highly non-trivial manner) to construct the universal structure Γ.

3.5.3 Distance CSPs

The last class of examples of concrete CSPs we will discuss are known as *distance CSPs*. These are CSPs of primitive positive expansions of $(\mathbb{Z}, \texttt{succ})$. These have been extensively studied in [25].

Here we will give a proof that $\texttt{CSP}((\mathbb{Z}, \texttt{succ}))$ is not equivalent to $\texttt{CSP}(\Gamma)$ for any ω-categorical structure Γ. In the proof we will use Theorem 3.4.3.

> **Theorem 3.5.1.** *CSP$((\mathbb{Z}, \texttt{succ}))$ cannot be expressed as CSP(Γ) where Γ is an ω-categorical template that is given by a primitive positive expansion of CSP$(\mathbb{Z}, \texttt{succ})$.*

Proof. First, note that $(\mathbb{Z}, succ)$ is not ω-categorical. This is an immediate consequence of one of the equivalent conditions of Ryll-Nardzewski. In particular, a structure is ω-categorical if, and only if it has finitely many complete n-types for each $n \in \mathbb{N}$. Recall that a complete n-type is a set of formulas $p(x_0, \dots, x_n)$ such that for any formula $\phi(x_0, x_1)$ either $\phi \in p$ or $\neg\phi \in p$.

It is fairly easy to see that $(\mathbb{Z}, \texttt{succ})$ has infinitely many complete 2-types. For $n \in \mathbb{N}$ let $\phi_n(x_1, x_2)$ be the following formula:

$$\exists y_1 \dots \exists y_n \left(\bigwedge_{1 \leq i \leq n-1} \texttt{succ}(y_i, y_{i+1}) \right) \wedge \texttt{succ}(x_1, y_1) \wedge \texttt{succ}(y_n, x_2).$$

Intuitively $\phi_n(x_1, x_2)$ says that the "distance" between x_1 and x_2 is exactly n. Then for each $n \in \mathbb{N}$ there exists a complete 2-type p_n that contains $\phi_n(x_1, x_2)$ and is not equivalent to any 2-type that contains $\phi_m(x_1, x_2)$ for $m \neq n$. Hence $(\mathbb{Z}, \texttt{succ})$ has infinitely many non-equivalent 2-types, hence it is not ω-categorical.

Now, suppose that we have a structure $(\mathbb{Z}, \{R_i\}_{i \in I})$, where $R_i(\bar{x})$ are n-ary relations and I a finite index set, such that $\texttt{succ}$ is primitive positive definable in $(\mathbb{Z}, \{R\}_{i \in I})$. This means that there exists some primitive positive formula $\psi(x_1, x_2)$ in $\{R_i\}_{i \in I}$ such that for any $z_1, z_2 \in \mathbb{Z}$ we have that $(\mathbb{Z}, \texttt{succ}) \vDash \texttt{succ}(z_1, z_2)$ if, and only if $(\mathbb{Z}, \{R_i\}_{i \in I}) \vDash \psi(z_1, z_2)$.

Once again it suffices to show that $(\mathbb{Z}, \{R_i\}_{i\in I})$ has infinitely many complete 2-types. This is easy to see, since the formula $\phi_n(x_1, x_2)$, defined above can be written as

$$\exists y_1 \dots \exists y_n \left(\bigwedge_{1\leq i\leq n-1} \psi(y_i, y_{i+1}) \right) \wedge \Phi(x_1, y_1) \wedge \Phi(y_n, x_2).$$

Since ψ is primitive positive we can rewrite the formula above with the existential quantifiers outside the conjunction and get a formula that expresses what we called "distance n" in $(\mathbb{Z}, \{R_i\}_{i\in I})$. Arguing just as before, this shows that we have infinitely many distinct 2-types again.

Of course, since $\mathtt{CSP}(\Gamma)$ and $\mathtt{CSP}(\Delta)$ are polynomial-time equivalent if and only if Γ and Δ have the same number of "maximal primitive positive n-types" and any structure that is a primitive positive expansion of $(\mathbb{Z}, \mathtt{succ})$ has infinitely many primitive positive maximal 2-types we see that it is impossible to express $\mathtt{CSP}((\mathbb{Z}, \mathtt{succ}))$ as the CSP of some ω-categorical structure, by Theorem 3.4.3. □

3.6 Descriptive Constraint Satisfaction

We will conclude this chapter by reprising the descriptive complexity approach from 3.2.4 and focusing on ω-categorical templates. We first recall the setup. Let $\mathcal{L}$ be a fixed logic and let $\mathcal{C}$ be a class of finite τ-structures. We say that $\mathcal{L}$ *describes* $\mathcal{C}$ if there exists a τ-sentence ϕ in $\mathcal{L}$ such that for all finite τ-structures Γ we have:

$$\Gamma \in \mathcal{C} \text{ if, and only if } \Gamma \models \phi.$$

We are, in particular, interested in the case that $\mathcal{C} = \mathtt{CSP}(\Gamma)$ for some fixed Γ. If there exists a formula ϕ in a logic $\mathcal{L}$ that describes $\mathtt{CSP}(\Gamma)$ we will say that $\mathtt{CSP}(\Gamma)$ is in $\mathcal{L}$. The questions we ask are of two flavours:

1. If $\mathcal{C} = \mathtt{CSP}(\Gamma)$ is in some logic $\mathcal{L}$, what does this tell us about Γ?
2. If $\mathcal{C} = \mathtt{CSP}(\Gamma)$ then what does this tell us about the logics $\mathcal{L}$ that describe $\mathcal{C}$?

We start off by answering the first question in two specific cases, first for first-order logic (denoted **FO**) and **SNP**. We first state, without proof, the following preservation theorem, of [9]:

Theorem 3.6.1. *Let τ be a finite relational signature and ϕ a first-order τ-sentence. Then ϕ is equivalent to an existential positive first-order τ-sentence ψ on all finite τ-structures if, and only if the class $\mathcal{C}$ of all finite τ-structures that satisfy ϕ is closed under homomorphisms.*

Using this, we prove the following result:

Theorem 3.6.2. *If CSP(Γ) is in* **FO** *then there exists an ω-categorical structure Γ' such that CSP(Γ') = CSP(Γ)*

Proof. Let χ be the first-order sentence that captures $\mathtt{CSP}(\Gamma)$ and let $\mathcal{C}$ be the class of all finite τ-structures that do not homorphically map into Γ. Recall that $\mathtt{CSP}(\Gamma)$ is closed under inverse homomorphisms, and hence it follows that $\mathcal{C}$ is closed under homomorphisms. Hence, by Theorem 3.6.1 there exists an existential positive first-order formula ψ that is equivalent to $\neg\chi$, i.e. such that $\Delta \in \mathtt{CSP}(\Gamma)$ if, and only if $\Delta \models \neg\psi$. Note that ψ is positive, which means that the quantifier-free part of ψ does not contain any negations. We may, without loss assume that the quantifier-free part of ψ is in disjunctive normal form. We thus have:

$$\exists \bar{x} \left(\bigvee_{i=1}^{n} \left(\bigwedge_{j_i=1}^{m_{j_i}} \psi_{j_i}(\bar{x}) \right) \right)$$

and in particular,

$$\bigvee_{i=1}^{n} \left(\exists \bar{x} \left(\bigwedge_{j_i=1}^{m_{j_i}} \psi_{j_i}(\bar{x}) \right) \right).$$

Again, without loss, we may assume that ψ is minimal, so removing any literal from ψ results in a non-equivalent formula. Write χ_i for the i-th disjunct of ψ and note that this is a primitive positive formula. Then, we argue that the canonical database of χ_i, $D(\chi_i)$, in the sense of 3.2.8, is connected.

To show this, suppose that for some i we have that $D(\chi_i)$ is not connected. Then, there exist non-empty structures A and B (with some relations on their vertices) such that $D(\chi_i) = A \sqcup B$. Note that, because we assumed that ψ is minimal then if for some $j \neq i$ we have $D(\chi_j)$ contains A or B we could remove the literals that cause the duplication, hence we can assume that there are no other clauses of ψ having either A or B as components of their canonical databases. By assumption, then, we may write χ_i as:

$$\bigwedge_{j=1}^{a} R_j(\bar{x}),$$

where the R_j are relation symbols from the signature τ (since we are in **FO**, all literals in ψ are made up of just relations from τ). Then we must have that this can be written as:

$$\left(\bigwedge_{k=1}^{a} R_k(\bar{y})\right) \wedge \left(\bigwedge_{l=1}^{b} R_l(\bar{z})\right)$$

where $\bar{y}$ and $\bar{z}$ contain distinct variables, from the vector $\bar{x}$. In particular, then, we have that the $\bar{y}$ form the domain of A and the $\bar{z}$ the domain of B. Note that if cannot be the case that A is a model for the canonical query of B, $Q(B)$ (and similarly for B and $Q(A)$). In particular, then, neither A nor B are models of χ_i and by minimality of ψ they are not models of ψ, either.

It then follows that $A \in \texttt{CSP}(\Gamma)$ and $B \in \texttt{CSP}(\Gamma)$, but then, since $\texttt{CSP}(\Gamma)$ is closed under disjoint unions we must have that $A \sqcup B \in \texttt{CSP}(\Gamma)$. This is a contradiction, since $A \sqcup B$ satisfy one of the distjuncts of ψ and hence $A \sqcup B \vDash \psi$.

In particular, the set $\mathcal{N} = \{D(\chi_i) \mid 1 \leq i \leq n\}$ is a finite set of finite, connected structures. By Theorem 3.4.4 we can find an ω-categorical structure Γ' which is universal for the class of $\mathcal{N}$-free structures, i.e. there is some ω-categorical Γ' such that $\text{Age}(\Gamma') = \text{wForb}(\mathcal{N})$.

We claim that $\texttt{CSP}(\Gamma') = \texttt{CSP}(\Gamma)$. Indeed, we note that $\texttt{CSP}(\Gamma)$ is precisely the set of finite structures Δ, such that $\Delta \vDash \neg\psi$. Indeed, then , Δ cannot have any structure from $\mathcal{N}$ as a (weak) substructure, for then it would satisfy a disjunct of ψ. On the other hand, if a structure in $\mathcal{N}$-free then it does not satisfy any disjunct of ψ and hence it does not satisfy ψ. It follows, then that it is in $\texttt{CSP}(\Gamma)$. Hence, we have shown that $\texttt{CSP}(\Gamma) = \texttt{CSP}(\Gamma')$, as required. □

Note that the proof above can be somewhat simplified, in fact it is a very similar argument to the one we carried out for connectedness of monadic **SNP** in Theorem 3.2.3 and we could, in fact have quoted this argument directly, since an **FO** formula is a monadic **SNP** formula with no second-order quantification. Indeed, in that proof we introduced no new quantifiers, so the proof would have sufficed to give the result above. Instead we gave a more complete argument, as we considered it a good opportunity to show how the definition of a canonical database for primitive positive formulas (from Subsection 3.2.2) and the definition of a canonical database as written in Definition 3.2.8 "play well with each other" when we discuss forbidden connected structures.

Of course, the class **FO** is not a large one in the computational complexity zoo. It is known, in fact, that **FO** lies at the bottom of the **LOGSPACE** hierarchy, that is to say, it is not a very interesting class. We can say more about CSPs of ω-categorical structures. The following is essentially a restatement of 3.3.2:

Theorem 3.6.3. *Let Γ be ω-categorical. Then* CSP(Γ) *is in monotone* **SNP***, without inequalities.*

Proof. We proved in 3.3.2 that if Γ is finitely bounded then $\texttt{CSP}(\Gamma)$ is in **SNP**. Note that $\texttt{CSP}(\Gamma)$ is closed under inverse homomorphisms, hence by lemma 3.3.2 we have that ϕ is equivalent to a monotone **SNP** sentence without inequalities, as claimed. □

Note that we did not use ω-categoricity in the result above, but we will use it in the following:

Theorem 3.6.4. *Let τ be a finite relational sentence and Γ a τ-structure. If* CSP(Γ) *is in monadic* **SNP** *then there exists an ω-categorical τ-structure Γ' such that* CSP(Γ') = CSP(Γ)*.*

Proof. Suppose that ϕ is the monadic **SNP** sentence that captures the class $\texttt{CSP}(\Gamma)$. Recall that if $\texttt{CSP}(\Gamma)$ is in captured by a monadic **SNP** sentence ϕ then there exists a monadic connected and monotone τ-sentence ψ that is equivalent to ϕ that captures $\texttt{CSP}(\Gamma)$, by Theorem 3.2.3. Thus we may assume that ϕ is monotone and connected. We may assume that ϕ is given in negation normal form, that it, it is in conjunctive normal form and each disjunction is negated, i.e. each disjunction is in fact a conjunction of positive and negative literals (but by monotonicity, we know that all literals from the signature are in fact positive). We thus write ϕ as:

$$\exists P_1, \ldots, P_k \forall \bar{x} \left(\bigwedge_{i=1}^{n} \psi_i(\bar{x}) \right),$$

where each $\psi_i(\bar{x})$ is precisely of the form:

$$\bigwedge_{j_i=1}^{a_i} R_{j_i}(\bar{x}) \wedge \bigwedge_{k_i=1}^{b_i} P_{k_i}(x) \wedge \bigwedge_{l_i=1}^{c_i} \neg P_{l_i}(x).$$

We expand the signature τ to τ' by introducing a new unary relation symbol P_i for each existential relation. Now, we repeat this, letting σ be the expansion of τ' with a new unary relation symbol Q_i for each existential relation P_i. Then we replace in all clauses each negated occurrence of P_i with Q_i, for each $i = 1, \ldots, k$. The resulting σ-sentence contains only positive literals. Indeed, it is precisely of the form:

$$\exists P_1, \ldots, P_k \forall \bar{x} \left(\bigwedge_{i=1}^{n} \left(\bigwedge_{j_i=1}^{a_i} R_{j_i}(\bar{x}) \wedge \bigwedge_{k_i=1}^{b_i} P_{k_i}(x) \wedge \bigwedge_{l_i=1}^{c_i} Q_{l_i}(x) \right) \right),$$

We may then rewrite this in negated CNF, as follows:

$$\exists P_1, \ldots, P_k \forall \bar{x} \left(\bigwedge_{i=1}^{n} \neg \left(\bigvee_{j_i=1}^{a_i} \neg R_{j_i}(\bar{x}) \vee \bigvee_{l_i=1}^{c_i} \neg Q_{l_i}(x) \vee \bigvee_{k_i=1}^{b_i} \neg P_{k_i}(x) \right) \right).$$

Call this sentence ϕ' and let ψ'_i be the i-th clause in the sentence above and consider the canonical database of ψ_i, $D(\psi_i)$, as a σ-structure. Then, by assumption, on the original monadic sentence ϕ, each $D(\psi'_i)$ is connected and hence we can define a finite set finite connected σ-structures $\mathcal{N} = \{D(\psi_i) \mid 1 \leq i \leq n\}$.

By Theorem 3.4.4 there exists an ω-categorical σ-structure Δ which is universal for the class of $\mathcal{N}$-free σ-structures. The point is that we can define $\Gamma' \subseteq \Gamma$ to be the σ-structure that only contains those points $v \in \Gamma$ such that, for each i it is not the case that $Q_i(v)$ and $P_i(v)$ both hold (that is, just one of them holds on each point of Γ'), of course, we take the relations on Γ' to be the restrictions of the relations on Δ, on the domain of Γ'. Note that this Γ' is first-order definable from an ω-categorical structure, so it is ω-categorical (by Corollary 3.4.2). Moreover, by forgetting the unary relations we may view Γ' as an ω-categorical τ-structure (reducts of ω-categorical structures are ω-categorical). We claim that a τ-structure satisfies ϕ if, and only if it homomorphically maps to Γ.

Suppose, first that a τ-structure A satisfies ϕ, that is, $A \in \texttt{CSP}(\Gamma)$. Then, by definition there exists a τ'-expansion of A that satisfies the first-order part of ϕ and, in fact, there exists a σ-expansion of A that satisfies the first-order part of ϕ'. It follows that as a σ-structure it is never the case that for a fixed i we have that both P_i and Q_i hold for the same $v \in \text{dom}(A)$. Moreover, since A satisfies ϕ' it does not contain any of the colour patterns forbidden by $\mathcal{N}$ and hence it belongs to $\text{wForb}(\mathcal{N})$. Thus, by universality of Γ', that is, by the fact that $\text{Age}(\Gamma') = \text{wForb}(\mathcal{N})$ the fact that $A \in \texttt{CSP}(\Gamma')$ follows.

Suppose, conversely, that A is a τ-structure that forbids all the structures in $\mathcal{N}$ (that is, none of them homomorphically map to A) and hence $A \in \texttt{CSP}(\Gamma')$. Then we can find a homomorphism $f : A \to \Gamma'$. In particular, for each $a \in \text{dom}(A)$ we may consider $f(a)$ as an element of Δ. Then $f(a)$ satisfies some of the P_i and the Q_i and we may define a σ-expansion of A by imposing these relations. Then, by construction we have a homomorphism from A to Δ and hence A does not contain any structure in $\mathcal{N}$ as a substructure. It follows thus that A satisfies ϕ.

The argument above is quite technical, but it uses some of the ideas we have introduced in interesting ways. To make this clear we summarise it below:

1. We started off with $\texttt{CSP}(\Gamma)$, for an arbitrary Γ and a monadic **SNP**-sentence ϕ capturing $\texttt{CSP}(\Gamma)$.
2. By Theorem 3.2.3 we rewrote ϕ as a connected and monotone monadic **SNP** sentence, in negated normal form.

3. Since monotone sentences define classes of forbidden structures and connected sentences define connected structures, we defined $\mathcal{N}$ to be precisely the class of structures that correspond to negated clauses in ϕ.
4. Monadic sentences impose what what we may think of as "colourings" of the vertices of structures (see the proof of Theorem 3.6.7 for an argument that makes explicit use of this interpretation of monadic predicates).
5. When considering the structures in $\mathcal{N}$ we forced them to keep track of the "colours" that the monadic predicates imposed.
6. By Theorem 3.4.4 we were able to find a structure Δ that is universal for $\mathcal{N}$-free structures.
7. We took Γ' to be the (first-order definable) substructure of Δ that only contained points where the "colourings" were well-defined.
8. We showed that interpreting Γ' as a τ-structure is precisely the structure we needed.

□

In some sense the proofs of Theorem 3.6.2 and theorem 3.6.4 are very similar. In fact the structure of the core arguments is identical:

(a) Get a formula ϕ which captures $\texttt{CSP}(\Gamma)$ and is monotone and connected.

(b) Define $\mathcal{N}$ to be the class of structures that forbid the structures in ϕ.

(c) Invoke Theorem 3.4.4 to get an ω-categorical structure Γ' and show that $\texttt{CSP}(\Gamma) = \texttt{CSP}(\Gamma')$

The difficulty in the proof of 3.6.2 was in step (a), which we got for free in the proof of 3.6.4 (by 3.2.3). On the other hand, the difficulty in the proof of 3.6.4 was in step (c), since we had to make sure that the monadic relations satisfied in models of ϕ are somehow preserved in Γ' (and to do so we actually had to add them in the vocabulary).

Let us fix some notation, in order to organise the classification of CSPs. We introduce the following:

- $\texttt{CSP}$ denotes the class of all CSPs on finite domains.
- $\texttt{CSP}^\star$ denotes the class of CSPs on finitely constrained ω-categorical domains.
- **SO**$\exists$ refers to the class of finite structures definable by existential second-order logic sentences and **SNP** refers to the class of finite structures definable by existential second-order sentences with a universal first-order part. We call these sentences **SNP**-sentences.
- M**SNP** (the M for *Monotone*) refers to the class of finite structures defined by an **SNP**-sentence in negation normal form, without inequalities.
- MM**SNP** (the second M stands for *Monadic*) refers the class of all finite structures definable by a monadic M**SNP**-sentence, i.e. an M**SNP**-sentence where the existential second-order quantifiers are over subsets of the domain.
- M**SO** refers to the class of all finite structures definable by a monadic second-order sentence. Note that structures here are not necessarily in **NP**. In particular, we have examples of structures that are in co-**NP**. To remedy this M**SO** + **SO**$\exists$ refers to the class of all finite structures definable by a monadic second-order sentence that lie in **NP**.
- **MSO**$\exists$ refers to the class of all finite structures definable by a monadic **SO**$\exists$ sentence.

With our newly introduced notation, we can summarise the complexity of CSPs as follows:

$$\texttt{CSP} \sim_{\mathbf{P}} MM\mathbf{SNP} \subsetneq \texttt{CSP}^\star \subsetneq M\mathbf{SNP} \subsetneq \mathbf{SNP} \sim_{\mathbf{P}} \mathbf{SO}\exists = \mathbf{NP}$$

We will now work, again, with some concrete CSPs to get a clearer picture of the hierarchy and show that the inclusions above are in fact strict.

3.6.1 $\texttt{CSP}((\mathbb{Q}, <))$

We have already spent some time discussing $\texttt{CSP}((\mathbb{Q}, <))$ and we have shown that it is in **P** and that it is not equivalent to $\texttt{CSP}(\Gamma)$ for any finite structure Γ. Note that this immediately implies that $\texttt{CSP}((\mathbb{Q}, <))$ cannot be defined by a first-order primitive positive sentence Φ, since in that case the canonical database for Φ would give us a finite template with the same CSP.

This may seem to contradict that $(\mathbb{Q},<) = \text{Flim}(\text{Forb}(\{I_2, C_1, C_2, C_3\}))$, since, of course forbidding a finite set of finite structures can be done using a first order sentence, the conjunction of the canonical query of each of the structures. The point, here, is that $\text{Forb}(\{I_2, C_1, C_2, C_3\}) \subsetneq \texttt{CSP}((\mathbb{Q},<))$, because by definition the left-hand side contains only tournaments, while the right-hand side contains arbitrary acyclic graphs. In fact, if we wanted to describe $\texttt{CSP}((\mathbb{Q},<))$ in this manner we would need to forbid structures that homomorphically map (in contrast to "embed") to C_n for each $n \in \mathbb{N}$.

Theorem 3.6.5. *$\texttt{CSP}((\mathbb{Q},<))$ is not in* **FO**.

Proof. Suppose that we can express the property G is acyclic using a first-order formula ϕ. Then the formula $\neg\phi$ has finite models for all $n \in \mathbb{N}$ and hence by compactness it has an infinite model. That is, there exists a graph G which contains an infinitely long cycle. This is a contradiction, as by assumption cycles can only have finite length, hence we cannot describe acyclicity in first-order logic. □

We also know, immediately from Fagin's Theorem that $\texttt{CSP}((\mathbb{Q},<))$ can be defined using an **SO**$\exists$ sentence. For instance, we could define it as follows:

$$\begin{aligned}\exists P \forall x \forall y \forall z ((E(x,y) \to P(x,y))&\\ \wedge(P(x,y) \wedge E(y,z) \to P(x,z))&\\ \wedge(P(x,y) \to \exists a E(x,a) \wedge P(a,y))&\\ \wedge(\neg E(x,x)) \wedge (\neg P(x,x)))&\end{aligned}$$

since we can define "path" in **SO**$\exists$, but we can do a bit better than that. To proceed, we need to examine $\texttt{CSP}((\mathbb{Q},<))$ as a computational problem, more closely.

In terms of complexity theory, let $G = (V,E)$ be a directed graph. We will give the following, fairly standard algorithm that solves the decision version of Acyclic, in **P**. Note that if a graph is acyclic then it contains at least one node with no outgoing edges.

We have a simple well-known algorithm which decides if a graph is acyclic or not:

1. If V is empty terminate with Yes.
2. Traverse the nodes of V. For each node test if it has no outgoing edges. If no node satisfying this condition is found, then terminate with No.
3. Pick a node with no outgoing edges v from G.
4. Repeat the algorithm with $G' = (V \setminus \{v\}, E \setminus \{E(u,v) \mid u \in V\})$.

Note that this algorithm at worst needs to perform $|V|^2$ comparisons and hence Acyclic $\in$ **P**.

One of the driving questions behind this project was to examine the CSPs that are in **MSO**. To this end, we have the following:

Theorem 3.6.6. *$\texttt{CSP}((\mathbb{Q},<))$ is in monadic co-***NP***, i.e. the complementary problem of $\texttt{CSP}((\mathbb{Q},<))$ is in* **MSO**$\exists$.

Proof. The idea of the proof is to encode the algorithm as an **SO**$\exists$ sentence. Let $\psi(v)$ encode the fact that v has no outgoing edges, that is, let $\psi(v)$ be the formula:

$$\forall u(\neg(E(v,u)).$$

Then, the algorithm tells us that G is acyclic if, and only if, whenever we consider a subgraph G' of G, given by removing some $v \in V$ satisfying $\psi(v)$ we have that G' is either empty or it contains at least one node v' satisfying $\psi(v')$. We can think of the subgraphs G' as the possible subgraphs of G which are closed under "inverse edges", that is, if $x \in G'$ and $E(y,x)$ in G then $y \in G'$. In particular, we have the following sentence ϕ:

$$\forall G'(\forall x \forall y(G'(x) \wedge E(y,x) \to G'(y))) \wedge ((\forall z \neg G'(z)) \vee (\exists z G'(z) \wedge \psi(z))).$$

Clearly, this sentence is monadic, as the second-order quantification is over subsets of G. Since, in the sentence above the second-order quantification is universal it follows that that $\texttt{CSP}((\mathbb{Q},<))$ is in monadic co-**NP**. □

Note that a famous result of Fagin, Stockmeyer and Vardi in [26] shows that monadic **NP** and monadic co-**NP** are distinct classes (cf. it is not known if **NP** and co-**NP** are distinct classes). This shows that we cannot say much more about the complexity of $\texttt{CSP}((\mathbb{Q},<))$, using arguments like the one above.

It should be noted here that Connected is in monadic co-**NP** and we tried to use this to construct a **MSO**$\exists$ sentence for Acyclic. Of course, we later realised that this would not be possible, since, as a matter of fact, Connected is actually not in **MSO**$\exists$, i.e. in monadic **NP**. In fact, this was precisely the example that was used to show that the two classes are distinct.

In fact, thought, there is more that we can say about $\texttt{CSP}((\mathbb{Q},<))$. The following is a corrected version of an argument from [7]:

Theorem 3.6.7. *CSP*$((\mathbb{Q},<))$ *is not in* MM**SNP**.

Proof. Suppose that there were an MM**SNP** sentence ϕ which describes all structures that homomorphically map to $(\mathbb{Q},<)$, that is, a sentence satisfied by all acyclic graphs. Fundamentally, by definition of monotone **SNP** we can view ϕ as a sentence which forbids a finite set of finite structures $\mathcal{N}$ and noting that if ϕ is also monadic then it must do so using only monadic predicates, that is, only by assigning "colours" to the vertices and then forbidding certain combinations of "colours".

By assumption, ϕ is satisfied by all directed paths of length n, for all $n \in \mathbb{N}$. We will show that in this case it must also satisfied by a directed cycle. We write ϕ as:

$$\exists P_1 \dots P_k \forall \bar{x} \left(\bigwedge_{i=1}^{n} \psi_i \right)$$

where $P_1, \dots, P_k$ are monadic predicates. We know that this sentence is satisfied by all directed paths of length $n \in \mathbb{N}$ and we can consider l, the length of the the longest path in terms of the input relation E in one of the clauses ψ_i of ϕ, that is, we may write each ψ_i as:

$$\left(\bigvee_{j=1}^{l} \neg E(x_\alpha, x_\beta) \right) \vee \left(\bigvee_{j=1}^{m} P_{k_j}(x_\gamma) \right) \vee \left(\bigvee_{j=1}^{m'} \neg P_{k'_j}(x_\delta) \right),$$

and consider the maximum l that appears in any ψ_i. That is, l is the longest (not necessarily directed) path that appears in the antecedent of a clause in ϕ. It is easier to intuitively understand what each clause ψ_i is saying, by writing it as:

$$\left(\bigwedge_{j=1}^{l_i} E(x_\alpha, x_\beta) \right) \to \left(\left(\bigvee_{j=1}^{m} P_{k_j}(x_\gamma) \right) \vee \left(\bigvee_{j=1}^{m'} \neg P_{k'_j}(x_\delta) \right) \right)$$

Each clause "dictates" an assignment of the existential relations to all (not necessarily directed) paths of length l_i so that we have the l_i nodes in the path are coloured not coloured in a way that makes the consequent false. Each of these colourings can be viewed as a "forbidden structure" in any structure that satisfies ϕ. Intuitively, this means that any sequence of connected nodes of length up to l with some assignments of the existential relations does not contain any of the structures that ϕ forbids.

In particular, since, as we said, any assignment of the existential monadic variables is nothing more than a colouring of the vertices of the graph and since we can assign each v to any of the k sets P_i we can view this as a colouring of the vertices with 2^k colours. Then, we can view the structures that ϕ forbids as specific combinations of colourings in paths of length up to l.

We know that a directed path of length $l((2^k)^l + 1)$ also satisfies ϕ. This means that we can assign each vertex v of the path to a set of P_i's so that all clauses in ϕ are satisfied. Now, colouring a path of length $l((2^k)^l + 1)$ with 2^k colours means that we will be able to find two distinct and, in fact, disjoint paths, each of length l, both having the same colouring. To see this, note that there are exactly $(2^k)^l$ colourings of a path of length l. If we split the path of length $l((2^k)^l + 1)$ into paths of length l we must have that at least two of them will have the same colouring, by the pigeonhole principle.

Suppose that the first path starts at position p_0 and the second at position p_1 and consider the colouring of the elements from p_0 to p_1. This path has length at least l and without loss we may in fact suppose that it has length l. Of course that path satisfies ϕ and as we can see, adding, after it nodes coloured with exactly the same colours as that path does not "break" ϕ. This means that a directed cycle $p_0 \to \cdots \to p_1 \to p_0$ will still satisfy ϕ, since ϕ has no way of "arguing" about forbidden colourings in paths of length more than l. This is a contradiction, since we have constructed a directed circle that satisfies ϕ. □

To clarify we know that if $\texttt{CSP}((\mathbb{Q}, <))$ were in monadic **SNP** then it would be in MM**SNP**, by Theorem 3.2.3. In particular, we have actually shown, combining with Theorem 3.2.3 that there is no **MSO**∃-sentence ϕ that captures $\texttt{CSP}((\mathbb{Q}, <))$, since if there were one it would have to be monotone, which is impossible.

3.6.2 Concluding the Classifications

At this point we have shown that:

- $\texttt{CSP}$ is a proper subclass of $\texttt{CSP}^\star$ (by Example 3.2.2).
- $\texttt{CSP}^\star$ is a proper subclass of **SO**∃ (even by Example 3.2.6).
- MM**SNP** is a proper subclass of $\texttt{CSP}^\star$ (by Theorems 3.6.4 and 3.6.7).
- $\texttt{CSP}$ is a subclass of MM**SNP** (by Theorem 3.2.1), and in fact, every problem in MM**SNP** is **P**-time equivalent to a problem in $\texttt{CSP}$, and vice versa (we quoted the result from [1]).

Recall Example 3.2.6, where we gave a formula in M**SNP** whose models are the graphs G such that neither G nor its complement has a triangle. We know, (e.g. by Ramsey Theory) that if G has more than 6 vertices then G does not satisfy that formula. Therefore, the class of finite structures that satisfy the formula from Example 3.2.6 is not closed under disjoint unions, so it is not $\texttt{CSP}(\Gamma)$ for any structure Γ. In particular, this shows that M**SNP** contains problems not expressible as CSPs. The obvious question then becomes: Is every CSP in M**SNP** the CSP of some ω-categorical structure?

The answer we give is negative, and in fact, we have already discussed a problem in M**SNP** that does not have an ω-categorical template. Recall that in Section 3.5.3 we showed that $\texttt{CSP}((\mathbb{Z}, \texttt{succ}))$ is not the CSP of any ω-categorical template. We will now give an M**SNP** sentence that describes all structures in $\texttt{CSP}((\mathbb{Z}, \texttt{succ}))$, i.e. all graphs with a well-defined notion of distance. This can be done as follows:

$$\begin{aligned} \exists E \exists T \forall w \forall x \forall y \forall z (&(E \text{ is an equivalence relation}) \\ &\wedge (T \text{ is transitive and irreflexive}) \\ &\wedge (\texttt{succ}(x,y) \to T(x,y)) \\ &\wedge (\texttt{succ}(x,y) \wedge \texttt{succ}(x,z) \to E(y,z)) \\ &\wedge (\texttt{succ}(x,y) \wedge \texttt{succ}(z,y) \to E(x,z)) \\ &\wedge (E(x,z) \wedge \texttt{succ}(x,y) \wedge \texttt{succ}(z,w) \to E(w,y)) \\ &\wedge (\texttt{succ}(x,y) \to \neg E(x,y)). \end{aligned}$$

This sentence (a corrected version of a sentence from [8]) precisely characterises any structure that has a successor relation, in $\mathbb{Z}$ we can view this as $\texttt{succ} = \{(x,y) \in \mathbb{Z}^2 \mid y = x+1\}$. The intuition is that T expresses the notion of a "successor induced" path and E expresses that vertices on two "parallel" paths with a common origin and endpoint give rise to an equivalence relation, which is the case if, and only if the graph we are considering has a well-defined distance. From the example above and Theorem 3.4.3 we get the final interesting inclusion, which is that:

- $\texttt{CSP}^\star$ is a proper subclass of the CSPs in M**SNP**.

In particular, this observation is the last piece missing to fully prove the strictness of all the inclusions given in the beginning of the section.

This concludes the chapter on CSPs and the essentially the mathematical material of the report. In this chapter we have presented some fundamental techniques for studying CSPs. We have shown that these techniques are actually strong enough to give us a fairly clear picture of the classification of CSPs, by applying them only to well-understood structures from Model Theory. In the end, we have managed to give a clear picture of some of the known results on CSPs by working either completely abstractly or only with very small examples, and this shows the power of applying Model Theory to the study of such problems.

Chapter 4

Conclusion

In this chapter, we will give a very quick introduction to some very important techniques in the study of Constraint Satisfaction Problems that we did not have the time to cover in this project. Moreover, we will present some open questions in the area, some known by people who specialise in constraint satisfaction, and some proposed by us. After that we will give a short summary of the work done and some general impressions.

4.1 Uncovered Topics

The main area theoretical aspect of model theoretic techniques applied to CSPs that we did not touch on was what is referred to in the literature as the *"algebraic approach"*. We give here a very short, informal introduction.

Model theory is still relevant, since this approach is based on the notion of a *core*. A finite structure is a core if all of its endomorphisms are embeddings. The core of a structure is a homomorphically equivalent structure over the same signature that is a core and in fact all finite structures have a core. The key idea is that it is easy to discuss the CSPs of cores and that a structure and its core have the same CSP. The definition of cores can be extended to infinite ω-categorical structures in a fairly natural manner and it gives us a brand new set of tools for discussing CSPs.

It is important to note that this treatment of CSPs is very important and is in fact what lead the proof of the big "dichotomy conjecture" for finite domain CSPs.

Another very interesting topic that unfortunately we did not have the time to cover in this project is the proof that a co-**NP**-intermediate problem exists in $\texttt{CSP}^\star$ (assuming $\mathbf{P} \neq \text{co-}\mathbf{NP}$). The proof uses machinery we have touched on, but is highly non-trivial. We refer the reader to Chapter 11 of [8], where a variety of non-dichotomy results are presented.

4.2 Open Questions

In general, it is conjectured that the CSPs that are in **MSO** are expressible as CSPs on ω-categorical templates. We pursued this question, to some extent, but being unable to prove that it is in fact the case that if $\texttt{CSP}(\Gamma) \in \mathbf{MSO}$ then $\texttt{CSP}(\Gamma)$ is equivalent to $\texttt{CSP}(\Gamma')$ for some ω-categorical structure Γ we tried to construct a counter example, looking at different CSPs, on non-ω-categorical structures (for instance $\texttt{CSP}((\mathbb{Z}, \texttt{succ}))$) and tried to find a monadic formula that captures them. This also proved a futile effort, as the set-up necessary even to pose such questions was significant and in the end we were unable to find an answer in time.

In general, the problem seems to be that **MSO** is a very large class, which contains both monadic **NP** and monadic co-**NP** problems (as we saw previously) and hence despite the extensive machinery we developed, we did not have enough tools to give a more informed approach to the problem.

In the process we actually came up with some sensible questions about the logics of CSPs. In particular, we defined in Section 3.2.4 a few complexity subclasses of $\mathbf{SO}\exists$ that did not appear in the classification we proved. These classes have not been widely studied and their interactions with CSPs are not known. In particular, the main question we tried to answer was:

- What is the relation between $\mathbf{MSO} + \mathbf{SO}\exists$ and $\texttt{CSP}^\star$?

The question has two flavours, of course. First, one may ask if it is the case that every CSP in $\mathbf{MSO} + \mathbf{SO}\exists$ is $\mathbf{P}$-time equivalent to a problem in $\texttt{CSP}^\star$ and conversely one may ask if given a problem in $\texttt{CSP}^\star$ there exists a formula in $\mathbf{MSO} + \mathbf{SO}\exists$ which captures it. We were unable to answer these questions, but we conjecture that it is in fact the case that given a problem in $\texttt{CSP}^\star$ one can find a formula in $\mathbf{MSO} + \mathbf{SO}\exists$ which captures it.

4.3 Final Thoughts

In conclusion, the overwhelming majority of the work in this project has been towards clarifying the discussions in the literature, and providing clearer arguments, where we deemed that the typical arguments given where too cryptic.

As noted in the closing remarks of Chapter 3, in doing so, we showed that the model theoretic tools that were presented in Chapter 2 can be applied in new and interesting ways to solve Complexity Theory problems. Indeed, these tools are strong enough to allow us to classify large classes of problems using only abstract results.

In general, it should be noted that the field of CSPs is still very young and at the same time very active and there will, we believe, definitely be more breakthroughs, like the proof of the dichotomy conjecture, in the coming years.

Appendix A

Back and Forth

A.1 Back and Forth Equivalence

In this section we will describe an equivalence relation lying somewhere between isomorphism and elementary equivalence. The discussion here is based on [4] and [6]. Note that we will use Back and Forth arguments in some examples later in the report, but they will not play an important part in the discussion of Constraint Satisfaction Problems, so this section can be safely skipped.

Back and Forth Games In this part of the report, unless otherwise stated, we let τ be an arbitrary signature, we take Γ, Δ to be τ-structures and γ an ordinal. We first define Ehrenfeucht-Fraïssé games in an informal manner.

There are two players $\exists$ and $\forall$. The $\forall$ player wants to show that Γ and Δ are different (in some sense) and the $\exists$ player wants to show that Γ and Δ are the same.

The game is played in γ moves. At each stage:

- $\forall$ picks one of the two structures, Γ or Δ, and an element of that structure.
- $\exists$ picks an element from the other structure.
- Both players have *perfect information*. That is, they both are aware of all previous moves that have been played in the game. Moreover, $\exists$ knows the choice $\forall$ made in that move.

In this context, let $\bar{a} = (a_i \in \Gamma : i \leq \gamma)$ and $\bar{b} = (b_i \in \Delta : i \leq \gamma)$. We define a *play* as a tuple $(\bar{a}, \bar{b})$. We say that $\exists$ wins the play if there exists an isomorphism $f : \langle \bar{a} \rangle_\Gamma \to \langle \bar{b} \rangle_\Delta$ such that $f(\bar{a}) = \bar{b}$. If no such isomorphism exists then $\forall$ wins. We will write $(\Gamma, \bar{a})$ for a τ'-expansion of Γ, where $\bar{a} \subseteq \Gamma$ andτ' contains a new constant symbol for each $a \in \bar{a}$, which we assume is "naming itself".

We write $\Gamma \equiv_0 \Delta$ if for every atomic τ-sentence ϕ we have that $\Gamma \vDash \phi$ if, and only if $\Delta \vDash \phi$. With this notation we have that $\exists$ wins the play $(\bar{a}, \bar{b})$ if, and only if $(\Gamma, \bar{a}) \equiv_0 (\Delta, \bar{b})$.

The "game" we have described is called the *Ehrenfeucht-Fraïssé game of length* γ *on* Γ *and* Δ, written $\mathrm{EF}_\gamma(\Gamma, \Delta)$.

It should be obvious that if $f : \Gamma \to \Delta$ is an isomorphism then $\exists$ has a *winning strategy*, i.e. a fixed set of rules that when followed guarantees that player the win in the game.

If $\exists$ has a winning strategy in $\mathrm{EF}_\gamma(\Gamma, \Delta)$, then we write $\Gamma \sim_\gamma \Delta$. Note that $\sim_\gamma$ is an equivalence relation of τ-structures, for fixed γ.

Example A.1.1. In this example, let τ be the signature for posets and let Γ and Δ be countable dense linear orders without endpoints. We will show that $\exists$ has a winning strategy for the $\mathrm{EF}_\omega(\Gamma, \Delta)$, i.e. that $\Gamma \sim_\omega \Delta$. As the game is played in ω moves, we will define the strategy of $\exists$ by induction. We will formalise this argument in the next section.
In the base case, suppose that $\forall$ picks either structure and some element from that structure. Then $\exists$ can, with no trouble, pick any element from the other structure. This gives us (a_0, b_0).
Now, let $\bar{a}$ be a sequence of i elements from Γ (all distinct) and let $\bar{b}$ be a sequence of i elements from Δ (again

distinct), such that $(\bar{a}, \bar{b})$ is the play of i-moves and that $\exists$ has a winning strategy up to this point. Suppose that $\forall$ picks structure Γ and some $a_{i+1} \in \text{dom}(\Gamma)$ as his move. Then $\exists$ needs to pick an element $b_{i+1} \in \text{dom}(\Delta)$ so that there is an isomorphism $f : \langle \bar{a}, a_{i+1} \rangle_\Gamma \to \langle \bar{b}, b_{i+1} \rangle_\Delta$, such that $f(a_i) = b_i$ for each i. There are 3 cases for $\exists$ to consider:

(i) If $a_{i+1} < a_i$ for all i. In this case $\exists$ can find some $b_{i+1} \in \text{dom}(\Delta)$, such that $b_{i+1} < b_i$ for all i, since Δ is infinite and has no endpoints.

(ii) If $a_i < a_{i+1}$ for all i. In this case $\exists$ can find some $b_{i+1} \in \text{dom}(\Delta)$, such that $b_i < b_{i+1}$ for all i, for the same reason as in case (i).

(iii) If $a_j < a_{i+1} < a_k$ for some $j, k \leq i$. In this case $\exists$ can still find some b_{i+1} lying strictly between b_k and b_k, since Δ is dense.

Note that if $\forall$ picks the other structure for his i-th move then the same exact argument works for $\exists$, by symmetry. In either case a structure preserving isomorphism $f : \langle \bar{a}, a_{i+1} \rangle_\Gamma \to \langle \bar{b}, b_{i+1} \rangle_\Delta$ can be found, and $\exists$, if he had a winning strategy up to play i still has a winning strategy for play $i + 1$. So $\exists$ has a winning strategy for $\text{EF}_\omega(\Gamma, \Delta)$.

Back and Forth Systems Two τ-structures Γ, Δ are called *back and forth equivalent* if $\exists$ has a winning strategy for $\text{EF}_\omega(\Gamma, \Delta)$. Let us now define another notion for describing back and forth equivalence.

A *back and forth system* from Γ to Δ is a set I of pairs $(\bar{a}, \bar{b})$, with $\bar{a}$ from Γ and $\bar{b}$ from Δ, such that:

- I is non-empty.
- If $(\bar{a}, \bar{b}) \in I$ then $\bar{a}$, $\bar{b}$ have the same length and $(\Gamma, \bar{a}) \equiv_0 (\Delta, \bar{b})$.
- For every $(\bar{a}, \bar{b}) \in I$ and every $c \in \Gamma$ there is a $d \in \Delta$ such that $(\Gamma, \bar{a}c) \equiv_0 (B, \bar{b}d)$.
- For every $(\bar{a}, \bar{b}) \in I$ and every $d \in \Delta$ there is a $c \in \Gamma$ such that $(\Gamma, \bar{a}c) \equiv_0 (B, \bar{b}d)$.

Note that we can replace all the conditions above by equivalent conditions where the back and forth system is replaced by a set of isomorphisms from finitely generated substructures of Γ to finitely generated substructures of Δ. In particular, let J be a set of isomorphisms $f : \langle \bar{a} \rangle_\Gamma \to \langle \bar{b} \rangle_\Delta$ such that $f(\bar{a}) = \bar{b}$. Then we may call J a back and forth system from Γ to Δ if it is such that:

- If $f \in J$ then f is an isomorphism from finitely generated substructures of Γ to finitely generated substructures of Δ.
- J is non-empty.
- For every $f \in J$ and every $c \in \Gamma$ there is a $g \supseteq f$ in J such that $c \in \text{dom}(g)$.
- For every $f \in J$ and every $d \in \Delta$ there is a $g \supseteq f$ in J such that $d \in \text{im}(g)$.

The following theorem shows us that all the definitions we have given so far:

Theorem A.1.1. *Let τ be a first-order language and Γ, Δ be τ-structures. Then Γ and Δ are back and forth equivalent if, and only if there is a back and forth system from Γ to Δ.*

A *position* of length n in a play of $\text{EF}_\gamma(\Gamma, \Delta)$ is a pair of n-tuples $(\bar{c}, \bar{d})$ such that $\bar{c}$, $\bar{d}$ are, in order, chosen in the first n moves of $\text{EF}_\gamma(\Gamma, \Delta)$. A position is *winning* if one of the players has a winning strategy from that position. The following should be clear:

- $(\bar{c}, \bar{d})$ is a winning position if, and only if a player has a winning strategy for $\text{EF}_\gamma((\Gamma, \bar{c}), (\Delta, \bar{d}))$.
- The starting position is winning for $\exists$ if, and only if Γ and Δ are back and forth equivalent.

Theorem A.1.2. *Let Γ, Δ be countable τ-structures. Then Γ, Δ are back an forth equivalent if and only if they are isomorphic.*

Proof. One direction is obvious, since, if Γ and Δ are isomorphic, the isomorphism gives a winning strategy for $\exists$.

Conversely, suppose that Γ and Δ are back and forth equivalent, then there is a back and forth system J from Γ to Δ. Enumerate the elements of Γ and Δ and let $f \in J$ be an isomorphism from a finitely generated substructure $\langle \bar{a} \rangle_\Gamma$ of Γ to a finitely generated substructure of $\langle \bar{b} \rangle_\Delta$ of Δ. By assumption there exists an isomorphism h_0 extending f such that $a_0 \in \text{dom}(h_0)$. Moreover, we can find an isomorphism g_0 such that $b_0 \in \text{im}(g_0)$. We proceed inductively, given g_k and taking h_{k+1} to be a map extending g_k in J such that $a_{k+1} \in \text{dom}((h_{k+1}))$ and g_{k+1} the map extending h_{k+1} so that $b_{k+1} \in \text{im}(g_{k+1})$. Then, the map $g = \cup_{n \in \mathbb{N}}(g_n)$ gives us the required isomorphism. □

This means that back and forth equivalence is a stronger notion than elementary equivalence. But consider:

> **Example A.1.2.** Let $\tau = \{\leq\}$ be the signature for linear orders. Take the τ-structures $R = (\mathbb{R}, \leq)$ and $Q = (\mathbb{Q}, \leq)$. Clearly, from the definition of back and forth equivalence, since we only consider countably many finitely generated substructures, the two structures are back and forth equivalent. Of course, though, they are not isomorphic.

It follows that it back and forth equivalence is not a stronger notion than isomorphism, though. But there is something we can do to connect back and forth equivalence with isomorphism.

Games of Elementary Equivalence In this final part we will restrict ourselves to relational signatures (although the results extend if we discuss unnested formulas). We write $\Gamma \approx_k \Delta$ if $\exists$ has a winning strategy for $\text{EF}_k(\Gamma, \Delta)$ for $k < \omega$. In this notation we have that:

$(\Gamma, \bar{a}) \approx_{k+1} (\Delta, \bar{b})$ if, and only if for every $c \in \Gamma$ there is a $d \in \Delta$ such that $(\Gamma, \bar{a}, c) \approx_k (\Delta, \bar{b}, d)$ and similarly for every $d \in \Delta$ there is a $c \in \Gamma$ such that $(\Gamma, \bar{a}, c) \approx_k (\Delta, \bar{b}, d)$.

> **Theorem A.1.3.** *Let τ be a finite signature. Then for any two τ-structures Γ, Δ we have that $\Gamma \equiv \Delta$ if, and only if for all $k < \omega$ we have $\Gamma \approx_k \Delta$.*

A.2 Back and Forth and Expressibility

The rest of this section is devoted to an application of Back and Forth games in Descriptive Complexity. It is taken from [16] and it offers an interesting point of view. Nonetheless, it is absolutely not essential for the content of the report.

A core notion in finite model theory and descriptive complexity is that of a query.

> **Definition A.2.1.** Let τ be a signature, $k \in \mathbb{N}$ and $\mathcal{C}$ a class of τ-structures that is closed under isomorphisms.
> - A *k-ary query* on $\mathcal{C}$ is given by a mapping Q such that:
>   1. For $\mathcal{A} \in \mathcal{C}$, $Q(\mathcal{A})$ is a k-ary relation on $\text{dom}(\mathcal{A})$.
>   2. Q is preserved under isomorphisms, that is, if $\mathcal{A} \cong \mathcal{B}$ and h is an isomorphism, then $h(Q(\mathcal{A})) = Q(\mathcal{B})$.
> - A *Boolean query* on $\mathcal{C}$ is a mapping $Q : \mathcal{C} \to \{0, 1\}$ that is preserved under isomorphisms, i.e. if $\mathcal{A} \cong \mathcal{B}$ then $Q(\mathcal{A}) = Q(\mathcal{B})$.

In particular, we may identify a Boolean query on $\mathcal{C}$ with the subclass $\mathcal{C}' = \{\mathcal{A} \in \mathcal{C} \mid Q(\mathcal{A}) = 1\} \subseteq \mathcal{C}$. A Boolean query expresses a property of a structure and a general k-ary query expresses a property of elements of a structure. In this way, queries allow us to formalise the notion that a property is expressible by a certain logic.

> **Definition A.2.2.** Let $\mathcal{L}$ be a logic and $\mathcal{C}$ a class of τ structures.
> - A k-ary query Q on $\mathcal{C}$ is *$\mathcal{L}$-definable* if there is an $\mathcal{L}$-formula $\phi(x_1, \dots, x_k)$ such that for $\mathcal{A} \in \mathcal{C}$
>
> $$Q(\mathcal{A}) = \{(a_1, \dots, a_k) \in \text{dom}(\mathcal{A})^k \mid \mathcal{A} \vDash \phi(a_1, \dots, a_k)\}.$$
>
> - A Boolean query Q on $\mathcal{C}$ is *$\mathcal{L}$-definable* if there is an $\mathcal{L}$-sentence ψ such that for each $\mathcal{A} \in \mathcal{C}$ we have
>
> $$Q(\mathcal{A}) = 1 \text{ if, and only if } \mathcal{A} \vDash \psi.$$

The first connections between the richness of a language required to express a property and the (computational) complexity of deciding whether a structure has a property are already becoming apparent. Just like a Turing Machine needs to operate correctly on all possible inputs so does a query need to express a property on all possible structures. We note that $\mathcal{C}$ may contain finite or infinite structures. In the context of Finite Model Theory we are preoccupied with finite structures and in particular with the class of all finite τ-structures for a fixed signature τ.

The expressive power of a logic $\mathcal{L}$ on a class $\mathcal{C}$ of finite structures is measured by the collection $\mathcal{L}(\mathcal{C})$ of $\mathcal{L}$-definable queries on $\mathcal{C}$. In a sense, this means that the expressive power of $\mathcal{L}$ is *context dependent.* For example, on the structure $\mathcal{N}$ of the naturals with addition and multiplication, first-order logic has high expressive power, but on the class of finite graphs this is not the case.

This brings us to one of the central questions in Finite Model Theory. Given a logic $\mathcal{L}$ and a class of structures $\mathcal{C}$ how can we determine which queries are $\mathcal{L}$-definable on $\mathcal{C}$ and which are not?

It is easy to see when a query can express a property on a class of structures. For the other direction we need to turn to Back-and-Forth Games and quantifier ranks.

Theorem A.2.1. *Let τ be a signature, $\mathcal{C}$ a class of τ-structures and Q a Boolean query on $\mathcal{C}$. Then Q is first-order definable on $\mathcal{C}$ if, and only if there is a positive integer r such that for $\mathcal{A}, \mathcal{B} \in \mathcal{C}$ if $\exists$ wins the r-move EF game on $\mathcal{A}$ and $\mathcal{B}$ and $Q(\mathcal{A}) = 1$ then $Q(\mathcal{B}) = 1$.*

This gives us a method of determining if a query Q is not first-order definable on $\mathcal{C}$. In particular, To show that Q is not first-order definable on $\mathcal{C}$, it suffices to show that for every positive integer r there are structures $\mathcal{A}_r$ and $\mathcal{B}_r$ in $\mathcal{C}$ such that $Q(\mathcal{A}_r) = 1$ and $Q(\mathcal{B}_r) = 0$ and $\exists$ wins the r-move EF game on $\mathcal{A}_r$ and $\mathcal{B}_r$. This method also has the property that if Q is not first-order definable on $\mathcal{C}$, then for every positive integer r such structures $\mathcal{A}_r$ and $\mathcal{B}_r$ exist.

Bibliography

[1] Feder T, Vardi MY. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study Through Datalog and Group Theory. SIAM J Comput. 1999 Feb;28(1):57–104. Available from: https://doi.org/10.1137/S0097539794266766.

[2] Bulatov AA. A Dichotomy Theorem for Nonuniform CSPs. 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS). 2017 Oct;Available from: http://dx.doi.org/10.1109/FOCS.2017.37.

[3] Zhuk D. The Proof of CSP Dichotomy Conjecture. CoRR. 2017;abs/1704.01914. Available from: http://arxiv.org/abs/1704.01914.

[4] Hodges W. A Shorter Model Theory. New York, NY, USA: Cambridge University Press; 1997.

[5] Papadimitriou CM. Computational complexity. Reading, Massachusetts: Addison-Wesley; 1994.

[6] Immerman N. Descriptive Complexity. Texts in Computer Science. Springer New York; 1998. Available from: https://books.google.co.uk/books?id=kWSZ0OWnupkC.

[7] Bodirsky M. Constraint Satisfaction with Infinite Domains. Humboldt-Universitaet zu Berlin; 2004.

[8] Bodirsky M. Complexity Classification in Infinite-Domain Constraint Satisfaction. CoRR. 2012;abs/1201.0856. Available from: http://arxiv.org/abs/1201.0856.

[9] Rossman B. Homomorphism Preservation Theorems. J ACM. 2008 Aug;55(3):15:1–15:53. Available from: http://doi.acm.org/10.1145/1379759.1379763.

[10] Tait WW. A Counterexample to a Conjecture of Scott and Suppes. J Symbolic Logic. 1959 03;24(1):15–16. Available from: https://projecteuclid.org:443/euclid.jsl/1183733465.

[11] Wang H. The Axiomatization of Arithmetic. The Journal of Symbolic Logic. 1957;22(2):145–158. Available from: http://www.jstor.org/stable/2964176.

[12] Fraïssé R. Sur certaines relations qui géenéralisent l'ordre des nombres rationnels. Comptes Rendus de l'Académie des Sciences;237:540–542.

[13] Lachlan A, Woodrow R. Countable Ultrahomogeneous Undirected Graphs. Transactions of The American Mathematical Society - TRANS AMER MATH SOC. 1980 11;262:51–51.

[14] Lachlan AH. Countable Homogeneous Tournaments. Transactions of the American Mathematical Society. 1984;284(2):431–461. Available from: http://www.jstor.org/stable/1999091.

[15] Henson CW. Countable homogeneous relational structures and $\aleph_0$-categorical theories. Journal of Symbolic Logic. 1972;37(3):494–500.

[16] Grädel E, Kolaitis PG, Libkin L, Marx M, Spencer J, Vardi MY, et al. Finite Model Theory and Its Applications. Texts in Theoretical Computer Science an EATCS Series. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007.

[17] van Emde Boas P. Handbook of Theoretical Computer Science (Vol. A). Cambridge, MA, USA: MIT Press; 1990. p. 1–66. Available from: http://dl.acm.org/citation.cfm?id=114872.114873.

[18] Cook SA. The Complexity of Theorem-proving Procedures. In: Proceedings of the Third Annual ACM Symposium on Theory of Computing. STOC '71. New York, NY, USA: ACM; 1971. p. 151–158. Available from: http://doi.acm.org/10.1145/800157.805047.

[19] Ladner RE. On the Structure of Polynomial Time Reducibility. J ACM. 1975 Jan;22(1):155–171. Available from: http://doi.acm.org/10.1145/321864.321877.

[20] Fagin R. Generalized first-order spectra, and polynomial. time recognizable sets. SIAM-AMS Proc. 1974 01;7.

[21] Kun G. Constraints, MMSNP and expander relational structures. Combinatorica. 2013 Jun;33(3):335–347. Available from: https://doi.org/10.1007/s00493-013-2405-4.

[22] Cherlin G, Shelah S, Shi N. Universal Graphs with Forbidden Subgraphs and Algebraic Closure. Advances in Applied Mathematics. 1999;22(4):454 – 491. Available from: http://www.sciencedirect.com/science/article/pii/S0196885898906418.

[23] Hubička J, Nešetřil J. Homomorphism and embedding universal structures for restricted classes. arXiv e-prints. 2009 Sep;p. arXiv:0909.4939.

[24] Opatrny J. Total Ordering Problem. SIAM Journal on Computing. 1979;8(1):111–114. Available from: https://doi.org/10.1137/0208008.

[25] Bodirsky M, Dalmau V, Martin B, Pinsker M. Distance Constraint Satisfaction Problems. In: Hliněný P, Kučera A, editors. Mathematical Foundations of Computer Science 2010. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. p. 162–173.

[26] Fagin R, Stockmeyer LJ, Vardi MY. On Monadic NP vs Monadic co-NP. Information and Computation. 1995;120(1):78 – 92. Available from: http://www.sciencedirect.com/science/article/pii/S0890540185711005.