Canonical Bases in Stable Theories

by

Ruizhang JIN
Department of Pure Mathematics
University Of Waterloo

Waterloo, Ontario, Canada, 2013

Table of Contents

1 Saturation **2**

2 Codes, Many-sorted Structures, and Imaginaries **4**

3 Codes for Type-definable Sets? **10**

4 Forking **11**

5 Canonical Bases **15**

APPENDIX **21**

A DLO Eliminates Imaginaries **21**

References **23**

Introduction

In model theory, a definable set is a set which is definable by a formula the parameters of which comes from the whole universe. For example, let $(\mathbb{R}, 0, 1, +, \times, <)$ be our structure, then the set $\{x|x > 0\}$ is definable by formulas like $x > 0$ and $x + \pi > \pi$. Notice that $x + \pi > \pi$ uses a parameter π, while $x > 0$ doesn't (because 0 is already in our language), so in this example the parameter π is actually redundant. It is then a natural question to ask how much a certain definable set actually depends on chosen parameters.

A canonical parameter (or a "code") for a definable set is a set of parameters that is in some sense minimal or irredundant. This will be made precise in Chapter 3, where we will show that a definable set is perfectly characterized by its code in the sense that any automorphism fixes the definable set iff it fixes the code. Codes do not always exist, but for any model M there is an easy way to construct a new model denoted M^{eq} which preserves all the information of the model M and also has codes for all definable sets. In a very loose sense, definable sets are nothing more than finite tuples in M^{eq}.

Since definable sets have codes in M^{eq}, a natural question we will ask is whether type-definable sets also have codes. It turns out that type-definable sets do not always have codes even in theories with very nice properties. However, in stable theories, if we loosen the restriction on what "fixing the type" means, we always have "canonical bases" in M^{eq} for type-definable sets. Canonical bases are a (weakened) parallel of codes for definable sets: an automorphism fixes a type *generically* iff it fixes a canonical base of the type.

The notion of canonical bases is a classical part of stability theory. The purpose of this essay is to give a gentle and thorough exposition of these well-known ideas.

This essay is organised as follows. In Chapter 1, we recall the definition of a sufficiently saturated model, in which we will often be working in the following chapters. In Chapter 2, we give the formal definition of "codes", and then introduce the many-sorted model M^{eq} which has codes for every definable set. We investigate the possibility of codes for types in Chapter 3, and this motivates discussing "forking" in a stable theory in Chapter

4. We give the definition of canonical bases for types in Chapter 5, which is a parallel of codes for definable sets, prove several facts about them. Finally, and work out an example: canonical bases of types in ACF_0.

In this essay, L will denote a first-order language, T will denote a complete L-theory with only infinite models, and M, N will denote models of T. We usually use A, B, C for parameter sets, and X, Y, Z for definable sets. We use a, b, c for both elements and tuples. When an automorphism α acts on a formula ϕ, we mean that α is applied on all the parameters in ϕ, and the result is denoted ϕ^α. Similarly for types: $p^\alpha = \{\phi\left(x, \alpha(a)\right) : \phi(x, y)$ is an L-formula, $\phi(x, a) \in p\}$. We use $\phi(M)$ to denote the set of realisations of the formula ϕ in a model M. Note that $\phi(M) \subseteq M^n$ where n is the arity of the tuple x. Similarly, for a partial type $p(x)$, $p(M)$ denotes the set of realisations of p. We use $S(A)$ to denote the set of all complete n-types over A, for all $n \geq q$. We often write $a \models p(x)$ to say that a is a realisation of $p(x)$ in some implicitly given model.

Chapter 1

Saturation

In this and the following two chapters, we are going to give facts that are necessary for later discussions. We are not going to give proofs of these facts, but rather refer the reader to [3] and Chapter 1 of [5] for further details.

Suppose T is a complete theory. In stability theory one usually works a *sufficiently saturated model* $\mathbf{C}$ of the theory T. By this we mean that for a sufficiently large cardinal κ, our model $\mathbf{C}$ is

1. *κ-saturated*, i.e., every type over parameters of cardinality $< \kappa$ is realized in $\mathbf{C}$ itself; and

2. *strongly κ-homogeneous*, i.e., if A and B are subsets of $\mathbf{C}$ of cardinality $< \kappa$, and f

is a bijection between A and B which is an elementary map, then f extends to an automorphism of $\mathbf{C}$ itself.

The following useful properties also hold for a sufficiently saturated model. They are consequences of 1 and 2 above.

3. $\mathbf{C}$ is κ^+-*universal*, i.e., for every model M of T of cardinality $\leq \kappa$, M is isomorphic to an elementary substructure of $\mathbf{C}$;

Let $\mathrm{Aut}_A(\mathbf{C})$ denote the set of automorphisms of $\mathbf{C}$ that fix A pointwise. Let $\mathrm{dcl}(A)$ denote the definable closure of A, i.e., $a \in \mathrm{dcl}(A)$ iff there exists a formula ϕ such that $\phi(\mathbf{C}) = \{a\}$. Let $\mathrm{acl}(A)$ denote the algebraic closure of A, i.e., $a \in \mathrm{acl}(A)$ iff there exists a formula ϕ such that $a \in \phi(\mathbf{C})$ and $|\phi(\mathbf{C})|$ is finite.

4. For any subset $A \subset \mathbf{C}$ of cardinality $< \kappa$, $a \in \mathrm{dcl}(A)$ iff for any $f \in \mathrm{Aut}_A(\mathbf{C})$, $f(a) = a$;

5. For any subset $A \subset \mathbf{C}$ of cardinality $< \kappa$, $a \in \mathrm{acl}(A)$ iff $\{f(a) : f \in \mathrm{Aut}_A(\mathbf{C})\}$ is finite.

6. For any subset $A \subset \mathbf{C}$ of cardinality $< \kappa$ and any definable set X, X is A-definable (i.e., definable by an L_A-formula) iff for any $f \in \mathrm{Aut}_A(\mathbf{C})$, $f(X) = X$.

Remark 1.1. For any theory T, there exist sufficiently saturated models. For a proof, we suggest that the reader refer to Chapter 4 of [3].

We will often be working in a fixed sufficiently saturated model $\mathbf{C}$ with an associated cardinal κ. In that case, unless otherwise specified, the conventions are as follows: a small set is a set of size $< \kappa$; parameter sets are always assumed to be small; models are also assumed to be small elementary submodels of $\mathbf{C}$; types are assumed to be over small sets or models. Sometimes we will need to break some of these conventions. In particular we will consider types over $\mathbf{C}$. To avoid confusions we will call types over $\mathbf{C}$ *global types* and denote them with boldface letters (e.g., $\mathbf{p}$).

Chapter 2

Codes, Many-sorted Structures, and Imaginaries

The readers may refer to Chapter 4 of [1] for details and proofs of the facts mentioned in this section.

The following captures what we might mean by "minimal" or "irredundant" parameters for a definable set.

Definition 2.1. Given a structure M, suppose $X \subset M^n$ is a definable set. A tuple a is called a *code* for X or a *canonical parameter* for X if there is an L-formula $\phi(x, y)$, such that $X = \phi(M, a)$ and if a' satisfies $X = \phi(M, a')$, then $a = a'$.

We say that a is a code for $\psi(x)$ if a is a code for $\psi(M)$.

We now give another characterisation of codes.

Proposition 2.2. *Let M be a $|T|^+$-saturated model of T, and X a definable set in M. Then a is a code for X iff for each automorphism α of M, α fixes X as a set iff $\alpha(a) = a$.*

Proof. The following proof is from the unpublished note *Some Elementary Facts About M^{eq}* by Rahim Moosa.

Assume that a is a code for X and $\phi(x, a)$ defines X where $\phi(x, y)$ is an L-formula. If an automorphism α of M fixes a, it fixes $\phi(x, a)$ and therefore fixes X. If $\alpha(X) = X$, then $\phi(M, a)^\alpha = \phi(M, a)$, so $\phi(M, \alpha(a)) = \phi(M, a)$. By the definition of codes, $\alpha(a) = a$.

Assume now that for each automorphism α of M, α fixes X as a set iff $\alpha(a) = a$. By Property 4 in Chapter 1, X is definable by a formula $\phi(x, a)$ where $\phi(x, y)$ is an L-formula.

Let $p(y) = \mathrm{tp}(a)$. We claim that $p(y)$ implies $(\neg\forall x(\phi(x,a) \leftrightarrow \phi(x,y))) \vee (a = y)$. If $a' \models p(y)$ but $a' \neq a$, then by saturation there is an automorphism of M that takes a to a', so by our assumption, $\phi(x,a')$ does not define the set X. By compactness there is an L-formula $\psi(y) \in p(y)$ which implies $(\neg\forall x(\phi(x,a) \leftrightarrow \phi(x,y))) \vee (a = y)$. Clearly $\phi(x,a) \wedge \psi(a)$ still defines the set X. If $\phi(x,a') \wedge \psi(a')$ also defines the set X, then $\psi(a')$ holds and $\forall x(\phi(x,a') \leftrightarrow \phi(x,a))$, which yields $a = a'$. Hence a is a code for X, witnessed by $\phi(x,y) \wedge \psi(y)$. □

Remark 2.3. It follows from the above proposition that if a and b are both codes of X, then $a \in \mathrm{dcl}(b)$ and $b \in \mathrm{dcl}(a)$.

We will often be working in a sufficiently saturated model. In these cases, we will always use this automorphism characterisation of codes, rather than the definition.

The following example shows that we do not necessarily have codes for all definable sets in a structure M.

Example 2.4. Let $L = (E)$ be our language where E is a binary relation, and T be the theory that there are infinitely many equivalence classes and each equivalence class has infinitely many elements. Let M be any model of T and $a \in M$. We prove that the definable set $E(M,a)$ does not have a code.

Suppose $b = (b_1, .., b_n)$ is a tuple and $\phi(x,b)$ defines the set $E(M,a)$. Let α be an automorphism of M which fixes every equivalence class as a set but does not fix b_1 (possible because every equivalence class has infinitely many elements). Then $\phi(x,\alpha(b))$ defines the set $E(M,a)^\alpha = E(M,a)$, but $\alpha(b) \neq b$. So b is not a code for $E(M,a)$.

It turns out that 0-definable equivalence relations are the only obstacles to definable sets having codes.

To solve this issue, we will introduce the model M^{eq} and the theory T^{eq}. We first introduce the concept of "many-sorted language".

Definition 2.5. (Many sorted language) A *many sorted language* is a language which contains sorts, relation symbols, function symbols (constants being 0-ary function symbols), and for each sort, variables of that sort. Each relation symbol R will be associated with a tuple $(S_1, ..., S_n)$ of sorts (called the *arity* of R); and each function f will be associated with a tuple of sorts for the domain (the *arity* of f) and also a target sort. Well-formed formulas are built up as usual, except we require that if the arity of a relation R is $(S_1, ..., S_n)$, then for $R(v_1, ..., v_n)$ to be a formula, each term v_i must be of sort S_i, and similarly for functions.

A structure for a many-sorted language M will consist of disjoint domains corresponding to the various sorts of the language. The interpretation of a relation symbol R will then be a subset of $S_1^M \times ... \times S_n^M$ where $(S_1, ..., S_n)$ is its arity, and similarly for the interpretation of function symbols.

Remark 2.6. We cannot construct by compactness an element which is not in any sort, because any variable belongs to a specific sort, and to say that some variable is not in a specific sort is thus not well-formed.

Now, for a 1-sorted language L, an L-theory T, and an L-structure M, we are going to introduce the many-sorted language L^{eq}, the L^{eq}-theory T^{eq}, and the L^{eq}-structure M^{eq} in a canonical fashion.

Let $ER(T)$ be the collection of $E \subset M^n \times M^n$ where E is a 0-definable equivalence relation, i.e., an equivalence relation on M^n definable by an L-formula $\phi_E(x, y)$ (without parameters), x and y being finite tuples of the same arity n. For each equivalence relation $E \in ER(T)$, L^{eq} will contain a sort S_E. In particular there will be a sort $S_=$. For E as above, L^{eq} will also contain a new function symbol f_E, whose arity is $(S_=)^n$ for the appropriate n and whose target sort is S_E. All the relation and function symbols of L will also be in L^{eq} and their arity will be of the form $(S_=)^n$ for some n.

Our structure M^{eq} is as follows. The interpretation of $S_=$ in M is just M itself, and the interpretation of the function and relation symbols in L are then interpreted accordingly. For every $E \in ER(T)$, the interpretation of the sort S_E will be $M^n/E = \{a/E : a \in M^n\}$ which is the set of E-classes of M^n, and the interpretation of f_E will be the function that maps $a \in M$ to a/E.

By considering all variables in a formula as in the sort $S_=$, we can treat any L-formula as an L^{eq}-formula. By induction, we have easily

Fact 2.7. For any L-formula $\phi(x)$, and any tuple a from M, $M \models \phi(a)$ iff $M^{eq} \models \phi(a)$.

Let T^{eq} be the theory claiming that all sentences that are true in T are true, with the additional axioms that for each $E \in ER(T)$, f_E is a surjective map from $(S_=)^n$ onto S_E, and $f(x) = f(y)$ iff $E(x, y)$. Clearly M^{eq} is a model of T^{eq} if M is a model of T.

The following are several facts about T^{eq}.

Fact 2.8.

1. Every model M^* of T^{eq} is of the form M^{eq} where $M := (S_=)^{M^*}$ is a model of T.
2. T^{eq} is complete.

3. If $\phi(x_1, ..., x_k)$ is an L^{eq} formula, and each x_i is of sort $S_=$, then there exists an L-formula $\psi(x_1, ..., x_k)$ which is equivalent to ϕ.

4. If M is κ-saturated and strongly κ-homogeneous, then M^{eq} is also κ-saturated and strongly κ-homogeneous.

5. An automorphism of M can be uniquely extended to M^{eq}. Moreover, every automorphism of M^{eq} is acquired in this way.

By definition, passing to M^{eq} gives us codes for certain equivalence classes. For each 0-definable equivalence relation $E \in ER(T)$ and each E-class X, let $b \in X$, $a = f_E(b)$, and $\phi(x, y)$ be the L-formula $y = f_E(x)$. Then $\phi(x, a)$ defines the set X, and if for some a', $\phi(x, a')$ also defines X, we have $a = a'$. The following proposition tells us more.

Proposition 2.9. *If $M \models T$, then for each definable set X in M, X as a definable set in M^{eq} has a code in M^{eq}.*

Proof. Let X be a definable set in M defined by $\phi(x, a)$ where $\phi(x, y)$ is an L-formula. Then $E(y_1, y_2) := \forall x(\phi(x, y_1) \leftrightarrow \phi(x, y_2))$ is a 0-definable equivalence relation. Let Y be the E-class containing a, and let $b = f_E(a) \in M^{eq}$. It is not hard to see that b is the code for our set X witnessed by the formula $\forall y(b = f_E(y) \to \phi(x, y))$. □

Remark 2.10. We actually proved the following fact in the above proof: if every equivalence class of every 0-definable equivalence relation in M has a code in M, then every definable set in M has a code in M.

Proposition 2.11. *Let M be a $|T|^+$-saturated model of T. Every definable set in M^{eq} has a code in M^{eq}.*

Proof. Let $E(x, x')$ be a 0-definable equivalence relation in M^{eq} where x and x' are of arity $(S_1, ..., S_n)$, let X be an E-class and $d \in X$, and let $f_1, ..., f_n$ be the functions of L^{eq} mapping from $S_=$ to S_i. Let $\psi(y_1, ..., y_n) = E\left((f_1(y_1), ..., f_n(y_n)), d\right)$ where $y_1, ..., y_n$ are of sort $S_=$, and let Y be the set defined by $\psi(y_1, ..., y_n)$. So Y is the pull-back of X to the home sort M. By the construction of M^{eq},we know that an automorphism of M^{eq} fixes X iff it fixes Y. Also, since $y_1, ..., y_n$ are of sort $S_=$, Y actually lives in M, and thus has a code in M^{eq}, say a. It follows from Proposition 2.2 that a is also a code for X.

By Remark 2.10, since every equivalence class in M^{eq} has a code in M^{eq}, every definable set has a code in it. □

We now continue to define elimination of imaginaries.

Definition 2.12. A theory T (possibly in a many-sorted language) has *elimination of imaginaries*, or *eliminates imaginaries*, if for every model M, every 0-definable equivalence relation E, and every E-class X, X has a code in M.

Remark 2.13.

1. If T is complete, one need only check the definition in some (rather than any) model.
2. Proposition 2.11 implies that for any complete theory T, T^{eq} has elimination of imaginaries.
3. Remark 2.10 shows that if T has elimination of imaginaries and $M \models T$, then every definable set in M has a code in M.

Passing to T^{eq} gives us codes for all definable sets. It is therefore never necessary to pass to $(T^{eq})^{eq}$; moreover, if T has elimination of imaginaries, it is not necessary for us to pass to T^{eq}.

Now consider the theory ACF_0, the theory of algebraically closed field of characteristic 0. We prove that ACF_0 eliminates imaginaries. First we give the following fact (proof can be found on p. 62 of [2]).

Fact 2.14. Let K be a field, I be an ideal of $K[x_1, ..., x_n]$. There exists a *minimal field of definition* of I. This means that there exists a field $k_0 \subseteq K$, such that

1. I is generated by polynomials whose coefficients are in k_0; and
2. If I is generated by polynomials whose coefficients are in $k \subseteq K$, then $k \supseteq k_0$.

Furthermore, if α is an automorphism of K, then $I^\alpha = I$ iff $\alpha \restriction_{k_0} = \mathrm{id}$, where $I^\alpha = \left\{ \sum_i \left(\alpha(a_i) \prod_j x_j^{n_{ij}} \right) : \sum_i \left(a_i \prod_j x_j^{n_{ij}} \right) \in I \right\}$.

Proposition 2.15. *ACF_0 eliminates imaginaries.*

Proof. We work in a sufficiently saturated model $K \models \mathrm{ACF}_0$. We assume some familiarity with this theory. In particular, we know that ACF_0 admits quantifier elimination, so every definable set is a finite boolean combination of *Zariski closed sets*, i.e., zero sets of polynomial equations.

Let $X \subset K^n$ be Zariski-closed. Let $I(X) = \{f \in K[x_1, ..., x_n] : f(b) = 0 \text{ for all } b \in X\}$ be the ideal of X. By Noetherianity $I(x)$ is finitely generated, so its minimal field of definition is a finitely generated filed extension of $\mathbb{Q}$. Suppose the minimal field of definition of $I(X)$ is $\mathbb{Q}(a_1, ..., a_l)$. Let $a = (a_1, ..., a_l)$.

If $\alpha \in \mathrm{Aut}(K)$ fixes a, then it fixes the field $\mathbb{Q}(a_1, ..., a_l)$ pointwise. Since $\mathbb{Q}(a_1, ..., a_l)$ contains the coefficients of the generators of the ideal $I(X)$, $I(X) = I(X)^\alpha$. Since X is Zariski-closed, $V(I(X)) := \{b : f(b) = 0 \text{ for all } f \in I(X)\}$ equals X, and since α fixes $I(X)$, α fixes $V(I(X)) = X$.

If $\alpha \in \mathrm{Aut}(K)$ fixes X, then it fixes $I(X)$. By Fact 2.14, α fixes the minimal field of definition of $I(X)$ pointwise, so it fixes $a = (a_1, ..., a_l)$.

By Proposition 2.2, a is a code for X. We proved that every Zariski-closed set has a code.

Now let X be an irreducible Zariski closed set with code a. Suppose that $Y \subsetneq X$ be a definable set with code b, and that the Zariski closure of $X \backslash Y$ is X. Every automorphism fixing a and b fixes the set $X \backslash Y$. Every automorphism α fixing $X \backslash Y$ preserves the collection of Zariski closed sets that contains $X \backslash Y$, thus fixing their intersection, which is X, the Zariski closure of $X \backslash Y$. Since α fixes X and $X \backslash Y$, it fixes Y, so it fixes a and b. So (a, b) is a code for $X \backslash Y$.

Now let X be an arbitrary definable set. Let $\bar{X}$ be the Zariski closure of X. Let $\bar{X} = X_1 \cup ... \cup X_n$ be the irreducible decomposition of X with a_i a code for for X_i. If $X \cap X_i = X_i$, then $c_i := a_i$ is a code for $X \cap X_i$. Otherwise, let $Y = X_i \backslash (X \cap X_i)$. Note that $Y \subset X_i$ is a definable set, and the Zariski closure of $X_i \backslash Y$ is X_i, so we need to find a code for Y. This can be done by using the same way we are using to find the code for X, and this recursive process will stop at some point because X is a finite boolean combination of Zariski closed sets. Now suppose b_i is a code for Y. By argument from the last paragraph, $c_i := (a_i, b_i)$ is a code for $X \cap X_i = X_i \backslash Y$. We now have $X = (X \cap X_1) \cup ... \cup (X \cap X_n)$ and each $X \cap X_i$ has a code c_i. Now, an automorphism fixes X iff it fixes $\{X_i \cap X : i = 1, ..., n\}$ as a set, which means that it might permute the subscripts (permuting the subscripts is the only possibility by the uniqueness of the irreducible decomposition). Suppose $X \cap X_i = \phi(x, c_i)$. If there exists an automorphism α such that $\alpha(X_i \cap X) = X_j \cap X$ for some j, then we may assume that $X \cap X_j = \phi(x, c_j)$, i.e., the codes c_i and c_j are based on the same formula. Let $C_i = \{c_j : \text{there exists an automorphism taking } X \cap X_j \text{ to } X \cap X_i\}$. Since elements in C_i are finite tuples and $|C_i|$ is finite, C_i is Zariski closed and has a code, say d_i. It is not hard to see that every automorphism fixes X iff it fixes all C_i's as sets, iff it fixes all d_i's. So $(d_1, ..., d_n)$ is a code for X.

Thus ACF_0 eliminates imaginaries. $\square$

From now on, unless otherwise specified, we will work implicitly inside T^{eq} instead of just T. For example, by $a \in \mathrm{acl}(A)$ we actually mean that $a \in \mathrm{acl}^{eq}(A)$.

Chapter 3

Codes for Type-definable Sets?

We work in a sufficiently saturated model $\mathbf{C}$, or rather $\mathbf{C}^{eq}$, of a complete theory T.

Since definable sets in $M^e q$ have codes, a natural question to ask is whether type-definable sets, i.e., sets of realisations of types, also have codes. A natural definition of codes for types is: a (possibly infinite) set B is a code for a type $p(x)$ if for every automorphism α of our ambient saturated model $\mathbf{C}$, $p^{\alpha}(\mathbf{C}) = p(\mathbf{C})$ iff $\alpha \restriction_B = \mathrm{id}$. In this chapter, we point out that unlike definable sets, even in $\mathbf{C}^{eq}$, type-definable sets do not necessarily have codes.

Let ACF_0 be our theory, K be our sufficiently saturated ambient model (containing the complex number set as a field), and $p(x)$ be the type saying that x is transcendental over the field $\mathbb{Q}(\pi)$. We claim that p does not have a code.

The set of realizations of $p(x)$ is the set of elements which are transcendental over $\mathbb{Q}(\pi)$, which is $K \backslash \mathbb{Q}(\pi)^{\mathrm{alg}}$. An automorphism α of K fixes $p(K)$ as a set iff it fixes $Q(\pi)^{\mathrm{alg}}$ as a set.

Suppose $a \in K \backslash \mathbb{Q}(\pi)^{\mathrm{alg}}$. Let $a' \in K \backslash \mathbb{Q}(\pi)^{\mathrm{alg}}$ be an element not equal to a. As both are transcendental, by saturation there exists an automorphism α of K that fixes $\mathbb{Q}(\pi)^{\mathrm{alg}}$ pointwise but maps a to a'. So $\alpha(p(K)) = p^{\alpha}(K) = p(K)$. Since α does not fix a, a code for p cannot contain a.

Suppose $a \in \mathbb{Q}(\pi)^{\mathrm{alg}} \backslash \mathbb{Q}^{\mathrm{alg}}$. Let $a' \in \mathbb{Q}(\pi)^{\mathrm{alg}} \backslash \mathbb{Q}^{\mathrm{alg}}$ be an element not equal to a. There exists an automorphism of $\mathbb{Q}(\pi)^{\mathrm{alg}}$ (which can be extended to an automorphism of K) that fixes $\mathbb{Q}^{\mathrm{alg}}$ pointwise but maps a to a'. Since α fixes $\mathbb{Q}(\pi)^{\mathrm{alg}}$ as a set, α fixes $K \backslash \mathbb{Q}(\pi)^{\mathrm{alg}} = p(K)$. Since α does not fix a, a code for p cannot contain a.

From the argument above, a code for our type p, if it exists, must be a subset of $\mathbb{Q}^{\mathrm{alg}}$. Let $a \in K \backslash \mathbb{Q}(\pi)^{\mathrm{alg}}$, and let $\sigma \in \mathrm{Aut}_{\mathbb{Q}^{\mathrm{alg}}}(K)$ take π to a. Clearly σ does not fix $\mathbb{Q}(\pi)^{\mathrm{alg}}$ as a set, so it does not fix $p(K)$. This means that a code for our type cannot be a subset of $\mathbb{Q}^{\mathrm{alg}}$. Therefore, p does not have a code.

Note that in our example, the automorphism σ that mapped π to the other transcendental number $a \notin \mathbb{Q}(\pi)^{\mathrm{alg}}$ did not fix the set $p(K)$, but it did not move it much. Indeed, p and p^σ have the same "generic" context, namely transcendentality. One way to resolve the problem of type-definable sets not having codes is to weaken the notion of code to ask automorphisms to only fix the type "generically". Of course we need a robust notion of "generic" for this.

In the following chapter, we will introduce the concept of non-forking extension, which is a precise definition of what "generic" might mean.

Chapter 4

Forking

Our goal here is to quickly review the basic notions around forking. The reader may refer to pp. 14-28 of [5] for detailed proofs of unproven facts in this section.

We fix a sufficiently saturated model $\mathbf{C}$ and work implicitly in $\mathbf{C}^{eq}$ in this section.

Definition 4.1. Let M be a model of T, $p(x) \in S(M)$, and $\delta(x, y)$ be an L-formula. An L_M-formula $\phi(y)$ is a *δ-definition* of p, if for all $b \in M$, $\phi(b)$ holds iff $\delta(x, b) \in p(x)$. A type $p(x) \in S(M)$ is *definable over $A \subset M$* if for each L-formula $\delta(x, y)$, there exists a δ-definition of p which is an L_A-formula. A type is *definable* if it is M-definable.

While δ-definitions need not be unique, every δ-definition of p defines the same set in the model M, namely $\{b : \phi(x, b) \in p(x)\}$. Hence they are equivalent modulo T. We therefore abuse notation by referring to *the* δ-definition of p, and denoting it by $d_p\delta(y)$.

A type p being A-definable means that for each L-formula $\delta(x, y)$, there is an L_A-formula $d_p\delta(y)$ which tells us which $\delta(x, b)$ are in our type. In particular note that if p is A-definable and $\alpha \in \mathrm{Aut}_A(M)$, then $p^\alpha = p$.

The problem, of course, is that types need not be definable. This is where stability theory enters the picture.

Definition 4.2. An L-formula $\delta(x, y)$ is *stable* if there do not exist two sequences of tuples $(a_1, ..., a_n, ...)$ and $(b_1, ..., b_n, ...)$ such that $\models \delta(a_i, b_j)$ iff $i \leq j$. A theory T is *stable* if every L-formula in T is stable.

We will not use this combinational definition directly, instead we will use the following key property: types over models in stable theories are definable.

Fact 4.3. Suppose $\delta(x, y)$ is stable and $p(x) \in S(M)$. Then $p(x)$ has a δ-definition over M.

From now on we will assume that our theory T is stable.

Let $p_1 \in S(A)$ and $p_2 \in S(B)$, where $A \subset B$ are two small sets. We say that p_2 is an extension of p_1 if every formula that appears in p_1 also appears in p_2. Consider, in the language of fields, the complete type $p(x)$ over $\mathbb{Q}$ that says the element is transcendental over $\mathbb{Q}$. Among the extensions of $p(x)$ to $\mathbb{C}$, there is a distinguished one, say $q(x)$, which says it is transcendental over $\mathbb{C}$. This extension, compared to the others, is somehow a "free" extension of $p(x)$,

This notion of "free extension" is made rigorous in stable theories by the following definition.

Definition 4.4. Let $M \subset N$ be models of T and $q(x) \in S(N)$. We say that q *does not fork* over M if q is definable over M.

It follows easily that for each L-formula $\delta(x, y)$, if $q \in S(N)$ is an extension of $p \in S(M)$ that does not fork over M, then there is a δ-definition of q that is a δ-definition of p. Since δ-definitions are unique modulo T-equivalence, it follows that p and q have the same δ-definitions.

Actually, it is a fact that for any models $M \subset N$ and any $p(x) \in S(M)$, there always exists an extension $q(x) \in S(N)$ which does not fork over M.Indeed, the δ-definitions of p tell us how to build $q(x)$: for each $a \in N$, we put $\delta(x, a)$ into q if $N \models d_p\delta(a)$. Of course one has to check that this produces a consistent complete type. This q will be the unique non-forking extension of p to N. We denote this q by $p|N$. In fact we have

Fact 4.5. Let M be a model, $A \subset M$ be algebraically closed and $p(x) \in S(A)$. There is a unique extension of p to M which is definable over A. We denote this extension by $p|M$.

Remark 4.6. We say that a set X is *algebraically closed* if $X = \mathrm{acl}(X)$. Similarly, X is definably closed if $X = \mathrm{dcl}(X)$.

Now we can extend Definition 4.4 to algebraically closed sets: Let $A \subset B$ be two algebraically closed parameter sets and $q \in S(B)$. We say that q *does not fork over* A if for some, equivalently any, model $M \supset A$, $q|M$ is definable over A.

By definition and Fact 4.5, $p \in S(A)$ does not fork over A.

And here is the final generalized definition of non-forking where the constraint of being algebraically closed is dropped.

Definition 4.7. Let $A \subset B$ be two arbitrary parameter set and $q \in S(B)$. We say that q *does not fork over* A, if there is an extension $q' \in S(\mathrm{acl}(B))$ of q, such that q' does not fork over $\mathrm{acl}(A)$. If $q \in S(B)$ extends $p \in S(A)$ and q does not fork over A, then we also say that q is a *non-forking extension of* p *to* B.

In fact, it turns out that if some extension of q to $\mathrm{acl}(B)$ does not fork over $\mathrm{acl}(A)$, then all extensions of q to $\mathrm{acl}(B)$ do not fork over $\mathrm{acl}(A)$. This follows from the fact that the group of elementary permutations of $\mathrm{acl}(B)$ which fixes B pointwise acts transitively on the set of extensions of q to $\mathrm{acl}(B)$ (see p. 20 of [5]).

Note that every extension of p to $\mathrm{acl}(A)$ is a non-forking extension.

Definition 4.8. A type $p \in S(A)$ is *stationary* if it has a unique non-forking extension to any set B containing A. In this case the non-forking extension of p to B is denoted by $p|B$.

Proposition 4.9. *Types over algebraically closed sets are stationary.*

Proof. This follows from Fact 4.5 and the definition.

Let $p(x) \in S(A)$ where A is algebraically closed. Suppose $B \supseteq A$. Let $M \supseteq \mathrm{acl}(B)$ be a model. By Fact 4.5, there exists a type $q \in S(M)$ which is definable over A. It is easy to see that $q \upharpoonright \mathrm{acl}(B)$ is also definable over A, and by definition, $q|B$ is a non-forking extension of p to B.

Suppose q_1, q_2 are two non-forking extensions of p to $B \supseteq A$, say $q_1, q_2 \in S(B)$. Let $q_1', q_2' \in S(\mathrm{acl}(B))$ be extensions of q_1, q_2, respectively. By definition, q_1, q_2 are non-forking extensions of p. Let $M \supset B$ be a model and consider $q_1'|M$ and $q_2'|M$. Since $q_1'|M$ and $q_2'|M$ are non-forking extensions of a type p over an algebraically closed set to the model M, by Fact 4.5 $q_1'|M = q_2'|M$. Hence $q_1 = q_2$ as desired. □

Thus, in order to show that a type over an arbitrary set A is stationary, it suffices to check that it has a unique extension to $\mathrm{acl}(A)$.

We can now extend the notion of δ-definability to stationary types over arbitrary sets A.

Definition 4.10. Let A be a parameter set and let $p \in S(A)$ be stationary. For an L-formula $\delta(x, y)$, the *δ-definition of p* is the δ-definition of $p|M$ for some, equivalently any, model $M \supseteq A$. We say that p is *definable over B* if it has δ-definitions over B for all δ.

Two types $p \in S(A)$ and $q \in S(B)$ are *parallel* if they have the same non-forking extension to some set containing $A \cup B$. If p and q are stationary, they then have the same δ-definitions for any L-formula $\delta(x, y)$. In this way, we consider parallel stationary types as "having the same generic content", thus formalizing what was hinted at in Chapter 3.

Proposition 4.11. *Let $p \in S(A)$ be stationary. Then p is definable over A.*

Proof. Fix a model $M \supset A$. Since p is stationary, p has a unique extension $p' = p|M$. It follows that for all $\alpha \in \mathrm{Aut}_A(\mathbf{C})$, $p'^{\alpha} = p'$. Hence α fixes $d_{p'}\delta(\mathbf{C})$ for all L-formulas $\delta(x, y)$. It follows from sufficient saturation of our ambient model that $d_{p'}\delta$ is equivalent to an L_A-formula. As $d_{p'}\delta$ is the δ-definition of p, we have that p is definable over A. □

Example 4.12. Let ACF_0 be our theory. Since ACF_0 admits elimination of imaginaries, we do not need and will not pass to ACF_0^{eq}.

Let $p(x) \in S(\mathbb{Q})$ be the type which says x is transcendental over $\mathbb{Q}$. Given any field F of characteristic 0 (so $\mathbb{Q}$ is a subfield of F), we claim that p has a unique non-forking extension to F and that that extension is the one saying that x is transcendental over F. We may assume that F is algebraically closed.

First note that $p(x)$ has a unique extension $p'(x) \in S(\mathbb{Q}^{\mathrm{alg}})$ which says that x is transcendental over $\mathbb{Q}^{\mathrm{alg}}$. As $\mathbb{Q}^{\mathrm{alg}} = \mathrm{acl}(\mathbb{Q})$, p' is a non-forking extension of p.

Now suppose $q(x) \in S(F)$ is a non-forking extension of $p(x)$. Then by definition it must be a non-forking extension of $p'(x)$. Let $n \geq 0$ and consider $\phi\left(x, (y_0, ..., , y_n)\right)$ the L-formula $(y_0 + y_1 x + ... + y_n x^n = 0) \wedge (y_n \neq 0)$. If $q(x)$ contains the formula $\phi\left(x, (a_0, ..., , a_n)\right)$ for some $(a_0, ..., a_n) \in F^{n+1}$, then $d_q\phi(F)$ is not the empty set, so $F \models \exists(y = (y_0, ..., y_n))d_q\phi(y)$.

Notice that $\mathbb{Q}^{\text{alg}} \preccurlyeq F$, and $\exists y d_q\phi(y)$ is an $L_{\mathbb{Q}^{\text{alg}}}$-formula (since q does not fork over $\mathbb{Q}^{\text{alg}}$, we can take $d_q\phi(y)$ to be an $L_{\mathbb{Q}^{\text{alg}}}$-formula). So $\mathbb{Q}^{\text{alg}} \models \exists y d_q\phi(y)$. Also notice that $d_q\phi(y)$ is a ϕ-definition of $p'(x)$ because q is a non-forking extension of p'. So there exists a tuple $(b_0, ..., b_n)$ from $\mathbb{Q}^{\text{alg}}$, such that p' contains the formula $(b_0 + b_1x + ... + b_nx^n = 0) \wedge (b_n \neq 0)$. But p' says that x is transcendental over $\mathbb{Q}^{\text{alg}}$, a contradiction. Thus $q(x)$ does not contain $\phi\,(x, (a_0, ..., , a_n))$ for any tuple $(a_0, ..., a_n)$ from F. This means that $q(x)$ is the type saying that x is transcendental over F.

This argument also shows that the type $p(x)$ is stationary.

Chapter 5

Canonical Bases

We have seen in Chapter 2 that for any theory, when we are working in $\mathbf{C}^{eq}$, every definable set X has a code, i.e., a tuple such that an automorphism fixes X setwise iff it fixes the code. In Chapter 3, we asked if there is something like a code for type-definable sets, and showed that the naive generalisation does not work. In this chapter we show that for stationary types in stable theories, there is an appropriate weakening of code that works, namely the canonical base.

Recall that our convention is that types are over small sets. However we will also consider types over the entire saturated model $\mathbf{C}$, and we call these *global types*, denoting them with boldface $\mathbf{p}$, $\mathbf{q}$, etc. If α is an automorphism, then $\mathbf{p}^\alpha$ is defined just like an automorphism acting on ordinary types: $\mathbf{p}^\alpha = \{\phi\,(x, \alpha(a)) : \phi(x, y) \text{ is an } L\text{-formula}, \phi(x, a) \in \mathbf{p}\}$.

We first define "codes" for global types in the naive way.

Definition 5.1. A set A is called the *canonical base* of a global type $\mathbf{p} \in S(\mathbf{C})$, if for each automorphism α of $\mathbf{C}$, $\mathbf{p}^\alpha = \mathbf{p}$ iff α fixes the set A pointwise.

The following example shows that not all global types have canonical bases.

Example 5.2. Let $\mathbf{C}$ be an $\aleph_1$-saturated model of DLO, the theory of dense linear ordering without endpoints. We assume some familiarity with this theory, in particular that DLO admits quantifier elimination and elimination of imaginaries. We give a proof of DLO eliminating imaginaries in the appendix.

Let $a_0 < a_1 < ... < a_n < ...$ be a sequence of increasing elements in $\mathbf{C}$. Let $\mathbf{p}(x)$ be the global type saying that $x > b$ iff $b > a_i$ for all $i = 0, 1, 2, ...$ (although $\mathbf{p}(x)$ is a partial global type, it actually uniquely determines a complete global type which we still denote by $\mathbf{p}(x)$). We claim that $\mathbf{p}(x)$ does not have a canonical base.

Since DLO eliminates imaginaries, if $\mathbf{p}$ has a canonical base, it will be in the home sort $\mathbf{C}$. We now prove that $\mathbf{p}$ does not have a canonical base. Suppose $b < a_k$ for some a_k. Since $b < a_k < a_{k+1} < ...$, there is an elementary map that takes the sequence $(b, a_k, a_{k+1}, ...)$ to $(a_k, a_{k+1}, a_{k+2}...)$ which can be extended to an automorphism α of $\mathbf{C}$. It is easy to see that $\mathbf{p}^\alpha = \mathbf{p}$, but $\alpha(b) = a_k \neq b$, so b is not in the canonical base of $\mathbf{p}$. Suppose now that $b > a_k$ for all $k = 0, 1, 2, ...$ Let $b' > b$. Since $a_0 < a_1 < a_2 < ... < b < b'$, there is an elementary map that takes the sequence $(b, a_k, a_{k+1}, ...)$ to $(b', a_{k+1}, a_{k+2}...)$ which can be extended to an automorphism α of $\mathbf{C}$. It is again easy to see that $\mathbf{p}^\alpha = \mathbf{p}$, but $\alpha(b) = a_k \neq b$, so b is not in the canonical base of $\mathbf{p}$. So if $\mathbf{p}$ has a canonical base, it has to be the empty set. Let $b_0 < b_1 < b_2 < ...$ be a sequence such that $b_i > a > j$ for all $i, j \geq 0$. Let $\mathbf{q}(x)$ be the global type saying that $x > b$ iff $b > a_i$ for all $i = 0, 1, 2, ...$ (similar to $\mathbf{p}(x)$, $\mathbf{q}(x)$ is also a complete type). The global types $\mathbf{p}$ and $\mathbf{q}$ are different because $x < b_0$ is in $\mathbf{p}$ but not in $\mathbf{q}$. There exists a partial elementary map that takes $(a_i)_{i=0}^n$ to $(b_i)_{i=0}^n$ which can be extended to an automorphism α. Since α (trivially) fixes the empty set but takes the global type $\mathbf{p}$ to a different global type $\mathbf{q}$, the empty set is not a canonical base of $\mathbf{p}$. Therefore, $\mathbf{p}$ does not have a code.

However, in stable theories, canonical bases always exist.

Proposition 5.3. *Assume that T is stable. Every global type $\mathbf{p} \in S(\mathbf{C})$ has a canonical of cardinality $\leq |T|$.*

Proof. We construct a canonical base of $\mathbf{p}$.

By Fact 4.3, for each L-formula $\delta(x, y)$, $\mathbf{p}$ has a δ-definition. Let $d_{\mathbf{p}}\delta(y)$ be a δ-definition of $\mathbf{p}$ and a_δ a code for $d_{\mathbf{p}}\delta(y)$. Let $A = \{a_\delta : \delta(x, y) \text{ is an } L\text{-formula}\}$. Note that A is a small set because it is of size at most the number of L-formulas.

We claim that A is a canonical base of $\mathbf{p}$.

If α is an automorphism of $\mathbf{C}$ and $\mathbf{p} = \mathbf{p}^\alpha$, then for each L-formula $\delta(x, y)$,

$$\begin{aligned} \mathbf{C} \models d_\mathbf{p}\delta(c) &\Leftrightarrow \delta(x, c) \in \mathbf{p} \\ &\Leftrightarrow \delta(x, \alpha(c)) \in \mathbf{p}^\alpha = \mathbf{p} \\ &\Leftrightarrow \mathbf{C} \models d_\mathbf{p}\delta(\alpha(c)). \end{aligned}$$

That is, α fixes $d_\mathbf{p}\delta(\mathbf{C})$. As a_δ is its code, it follows that $\alpha(a_\delta) = a_\delta$. So α fixes A pointwise.

Conversely, suppose α fixes A pointwise, i.e., for each L-formula $\delta(x, y)$, $\alpha(a_\delta) = a_\delta$. Then

$$\begin{aligned} \delta(x, \alpha(c)) \in p &\Leftrightarrow \mathbf{C} \models d_\mathbf{p}\delta(\alpha(c)) \\ &\Leftrightarrow \mathbf{C} \models d_\mathbf{p}\delta(c) && \text{Since } \alpha \text{ fixes the code for } d_\mathbf{p}\delta(y) \\ &\Leftrightarrow \delta(x, c) \in \mathbf{p}. \end{aligned}$$

Thus $\mathbf{p}^\alpha = \mathbf{p}$. □

In the rest of this chapter, we assume that our theory T is stable.

Proposition 5.4. *Let* $\mathbf{p}$ *be a global type. If* A *is a canonical base of* $\mathbf{p}$*, then* B *is a canonical base of* $\mathbf{p}$ *iff* $\mathrm{dcl}(A) = \mathrm{dcl}(B)$*. In particular, every canonical base is small.*

Proof. We freely use the automorphism characterisation of dcl that we have in sufficiently saturated models (see (4) in Chapter 1).

If B is a canonical base of $\mathbf{p}$, then every automorphism of $\mathbf{C}$ fixes B pointwise iff it fixes $\mathbf{p}$ iff it fixes A pointwise. So $\mathrm{dcl}(A) = \mathrm{dcl}(B)$.

Conversely, assume $\mathrm{dcl}(A) = \mathrm{dcl}(B)$. Let α be an automorphism of $\mathbf{C}$. If $\mathbf{p}^\alpha = \mathbf{p}$, then $\alpha|_A = \mathrm{id}$, thus $\alpha|_{\mathrm{dcl}(A)} = \mathrm{id}$, and since $B \subset \mathrm{dcl}(A)$, α fixes B pointwise. If α fixes B pointwise, then $\alpha|_{\mathrm{dcl}(B)} = \mathrm{id}$ and thus α fixes A pointwise, so $\mathbf{p}^\alpha = \mathbf{p}$. We have proved that B is also a canonical base of $\mathbf{p}$.

By Proposition 5.3, for each global type there exists one canonical base which is small. Since small is preserved under definable closure, all canonical bases are small. □

This justifies denoting by $\mathrm{cb}(\mathbf{p})$ the set $\mathrm{dcl}(A)$ where A is any canonical base of the global type $\mathbf{p}$. We abusively call $\mathrm{cb}(\mathbf{p})$ *the* canonical base of $\mathbf{p}$.

Using non-forking extensions we can extend the definition of canonical bases to ordinary stationary types.

Definition 5.5. Suppose $p(x)$ is a stationary type (over a small set of parameters). A *canonical base of* p is by definition a canonical base of the global non-forking extension $p|\mathbf{C}$.

The following characterisation shows that canonical bases do encode the generic content of stationary types.

Proposition 5.6. *Suppose $p(x) \in S(A)$ is stationary and B is a parameter set. The following are equivalent:*

(i) B is a canonical base of p.

(ii) For all $\alpha \in \mathrm{Aut}(\mathbf{C})$, $\alpha \restriction B = \mathrm{id}$ iff p^α and p are parallel.

Proof. Fix $\alpha \in \mathrm{Aut}(\mathbf{C})$. Note that as automorphisms preserve non-forking extensions, $p^\alpha \in S(\alpha(A))$ is also stationary. It follows that $(p|\mathbf{C})^\alpha = p^\alpha|\mathbf{C}$. Hence

$$\begin{aligned} & p \text{ and } p^\alpha \text{ are parallel} \\ \Leftrightarrow\ & p|\mathbf{C} = p^\alpha|\mathbf{C} \\ \Leftrightarrow\ & p|\mathbf{C} = (p|\mathbf{C})^\alpha. \end{aligned}$$

The equivalence of (i) and (ii) is then an immediate consequence of the equivalence displayed above. □

As before, we define $\mathrm{cb}(p) = \mathrm{dcl}(B)$ where B is a canonical base of p. Equivalently, $\mathrm{cb}(p) = \mathrm{cb}(p|\mathbf{C})$. We also say that $\mathrm{cb}(p)$ is *the* canonical base of p.

The following proposition tells us where the canonical base of a type lies.

Proposition 5.7. *For any stationary type $p \in S(A)$, $\mathrm{cb}(p) \subseteq \mathrm{dcl}(A)$.*

Proof. Suppose $p(x) \in S(A)$ is stationary. For any L-formula $\delta(x, y)$, let b_δ be a code for $d_p\delta(y)$. Since $p(x)$ is definable over A, $d_p\delta(\mathbf{C})$ is an A-definable set, so $b_\delta \in \mathrm{dcl}(A)$. Since by definition $d_p\delta(y)$ are also the δ-definitions of $p|\mathbf{C}$, the set $B = \{b_\delta : \delta(x, y) \text{ is an } L\text{-formula}\}$ is a canonical base of $p|C$, and thus a canonical base of p. Since each element of B lies inside $\mathrm{dcl}(A)$, we get that $\mathrm{cb}(p) = \mathrm{dcl}(B) \subseteq \mathrm{dcl}(A)$. □

The following characterisation of canonical base is very useful, and explains in what sense the canonical base is a "minimal" parameter set.

Proposition 5.8. *Let $p \in S(A)$ be a stationary type. Then* $\mathrm{cb}(p)$ *is the smallest definably closed set $B \subseteq \mathrm{dcl}(A)$ such that p does not fork over B and $p \restriction B := \{L_B$ formulas in $p\}$ is stationary.*

Proof. We first prove that p does not fork over $\mathrm{cb}(p)$. To prove this, by Definition 4.7, we need to prove that there is an extension $p' \in S(\mathrm{acl}(A))$ of p that is definable over $\mathrm{acl}(\mathrm{cb}(p))$. Notice that p is stationary, and every extension of p to $\mathrm{acl}(A)$ is non-forking, so p' is actually the unique non-forking extension of p to $\mathrm{acl}(A)$. Let $\delta(x, y)$ be an L-formula, and $d_{p'}\delta(y)$ be a δ-definition of p'. By Definition 4.10, $d_{p'}\delta(y)$ is also a δ-definition of $p'|\mathbf{C} = p|\mathbf{C}$. By the construction of a canonical base of $p|\mathbf{C}$ in the proof of Proposition 5.3, we know that $d_{p'}\delta(\mathbf{C})$ is $\mathrm{cb}(p)$-definable. So p' is definable over $\mathrm{cb}(p)$, thus over $\mathrm{acl}(\mathrm{cb}(p))$.

We now prove that $p \restriction \mathrm{cb}(p)$ is stationary. Let $r := (p|\,\mathrm{acl}(A)) \restriction \mathrm{acl}\,(\mathrm{cb}(p))$. Note that as p does not fork over $\mathrm{cb}(p)$, $p|\,\mathrm{acl}(A)$ is a non-forking extension of the stationary type r. Hence $\mathrm{cb}(p) = \mathrm{cb}(r)$. Now suppose $q \in S(\mathrm{acl}(\mathrm{cb}(p)))$ is an extension of $p \restriction \mathrm{cb}(p)$. As is mentioned in Chapter 4, the group of elementary permutations of $\mathrm{acl}(\mathrm{cb}(p))$ which fixes $\mathrm{cb}(p)$ pointwise acts transitively on the set of extensions of $p \restriction \mathrm{cb}(p)$ to $\mathrm{acl}(\mathrm{cb}(p))$ (see p. 20 of [5]). So there exists an automorphism α of $\mathbf{C}$ such that $\alpha \restriction_{\mathrm{cb}(p)} = \mathrm{id}$ and $q^\alpha = r$. We now get that $\mathrm{cb}(q^\alpha) = \mathrm{cb}(r)$, and by taking α^{-1} on both sides, we get that $\mathrm{cb}(q) = \mathrm{cb}(r)$ (since α fixes $\mathrm{cb}(r) = \mathrm{cb}(p)$ pointwise). As $q, r \in S(\mathrm{acl}(\mathrm{cb}(p)))$, being parallel means that in fact $q = r$. We have shown that $p \restriction \mathrm{cb}(p)$ has a unique extension to $\mathrm{acl}(\mathrm{cb}(p))$, so $p \restriction \mathrm{cb}(p)$ is stationary.

Now suppose that p does not fork over B and $p \restriction B$ is stationary for some definably closed set $B \subseteq \mathrm{dcl}(A)$. Note then that p and $p \restriction B$ have the same δ-definitions. Let b_δ be a code for the δ-definition of p for some L-formula $\delta(x, y)$. Since B is definably closed, we have that $b_\delta \in B$, so $\{b_\delta : \delta(x, y)$ is an L-formula$\} \subseteq B$, and by taking definable closure on both sides, we get that $\mathrm{cb}(p) \subseteq B$. □

We conclude this essay with a description of canonical bases in ACF_0.

Let $K \models \mathrm{ACF}_0$ be our sufficiently saturated model. Since ACF_0 eliminates imaginaries, we do not pass to K^{eq}.

Let F be a (small) field of characteristic 0. Given $p(x) \in S_n(F)$ a complete n-type over F, let $I_p = \{f(x) \in F[x_1, ..., x_n] : f(a) = 0$ for all (equivalently any) $a \models p(x)\}$. Let $V_p = V(I_p) = \{(a_1, ..., a_n) \in K^n : f(a_1, ..., a_n) = 0$ for all $f \in I_p\}$. By noetherianity V_p is an F-definable Zariski closed set, and since $p(x)$ is a complete type over F, V_p is F-irreducible.

Let $V \subseteq K^n$ be an F-irreducible F-definable Zariski closed set. Let $p(x)$ be the complete type containing $x \in V$ and $\{x \notin W : W \subsetneq V, W\ F\text{-definable}\}$ (by quantifier elimination there is a unique type containing this partial type). See Chapter 7 of [4] for a proof.

Proposition 5.9. *Let $F_1 \subset F_2$ be two algebraically closed (small) fields, $p(x) \in S_n(F_1)$, and $q(x) \in S_n(F_2)$. Then $q(x)$ is a non-forking extension of $p(x)$ iff $V_p = V_q$.*

Proof. Let $I_p = \{f(x) \in F_1[x_1, ..., x_n] : f(a) = 0 \text{ for all (equivalently any) } a \models p\}$ be an ideal in $F_1[x_1, ..., x_n]$ generated by $f_1, ..., f_k$. If $I_q = \{f(x) \in F_2[x_1, ..., x_n] : f(a) = 0 \text{ for all (equivalently any) } a \models q\}$ (which is an ideal in $F_2[x_1, ..., x_n]$) is also generated by $f_1, ..., f_k$, then we immediately get that $V_p = V_q$.

Let U_m be the set of monomials of degree not greater than m with coefficient 1. Let $\delta(x, y)$ be the L-formula, where x is of arity n and $y = (y_u)_{u \in U_m}$ is of arity $|U_m|$, saying that $\sum_{u \in U_m} y_u u(x) = 0$, i.e., the polynomial whose coefficient is y has x as a root. Let us find explicitly a δ-definition of $p(x)$. Since I_p is generated by $f_1, ..., f_k$, for a tuple $c = (c_u)_{c \in U_m}$ from F_1, $\delta(x, c) \in p(x)$ iff there exist polynomials $q_1, ..., q_k \in F_1[x_1, ..., x_n]$ of degree $\leq m$, such that $\sum_{i=1}^k q_i f_i = \sum_{u \in U_m} c_u u$. By interpreting "there exist polynomials" as "there exist coefficients for polynomials", this can be written as an L_{F_1}-formula, say $\psi(y)$, which is a δ-definition of $p(x)$. Since non-forking extension preserves δ-definitions, $\psi(y)$ is also a δ-definition of $q(x)$. That is, for any tuple $c = (c_u)_{u \in U_m}$ from F_2, $\delta(x, c) \in q(x)$ iff there exist $q_1, ..., q_n \in F_2[x_1, ..., x_n]$ of degree $\leq m$, such that $\sum_{i=1}^k q_i f_i = \sum_{u \in U_m} c_u u$. But $\delta(x, c) \in q(x)$ iff $\sum_{u \in U_m} c_u u \in I_q$. Hence, by enumerating all $m > 0$, we get that I_q is generated by $f_1, ..., f_k$. $\square$

Remark 5.10. In general, it is not hard to prove that if F_1 and F_2 are not necessarily closed, then $q(x)$ is a non-forking extension of $p(x)$ iff V_q is an F_2-irreducible component of V_p.

Proposition 5.11. *Suppose $p \in S_n(F)$ is stationary. Then $\mathrm{cb}(p)$ is the minimal field of definition of V_p.*

Proof. We use the automorphism characterisation of canonical bases: an automorphism $\alpha \in \mathrm{Aut}(K)$ fixes a canonical base of p iff p and p^α are parallel.

For $\alpha \in \mathrm{Aut}(K)$, let G be an algebraically closed group containing F and $\alpha(F)$. By definition, p and p^α are parallel iff $p|G = p^\alpha|G$, and by Proposition 5.9 and the correspondence between types over G and G-irreducible Zariski closed sets, iff $V_p = V_{p^\alpha}$. Thus a code for V_p is a canonical base of p.

Let $H = \mathbb{Q}(a_1, ..., a_l)$ be the minimal field of definition of V_p. We have shown in the proof of Proposition 2.15 that $(a_1, ..., a_l)$ is a code for the Zariski closed set V_p, i.e., $\alpha \in \mathrm{Aut}(K)$ fixes $(a_1, ..., a_l)$ iff it fixes the set V_p. So $(a_1, ..., a_l)$ is a canonical base of p, and $\mathrm{cb}(p) = H = \mathrm{dcl}(a_1, ..., a_l)$, which is the minimal field of definition of V_p. $\square$

APPENDIX

Appendix A

DLO Eliminates Imaginaries

Recall (from Definition 2.12) that a theory T eliminates imaginaries, if for any model M of T, and any definable set X in M, there exists an L-formula $\phi(x, y)$ and a tuple a such that $X = \phi(M, a)$, and if $X = \phi(M, a')$, then $a' = a$. We prove here that DLO, the theory of dense linear ordering without endpoints, eliminates imaginaries.

In the rest of this appendix, the theory will always be DLO. We assume some familiarity of this theory. In particular, we use without proof the fact that DLO has quantifier elimination. We will always be working in a sufficiently saturated model $\mathbf{C}$.

Lemma A.1. *Let X be a definable set in $\mathbf{C}$. If both $\phi(x, a)$ and $\psi(x, b)$ defines X, where $\phi(x, y)$ and $\psi(x, z)$ are two L-formulas, and $a = (a_1, ..., a_m)$ and $b = (b_1, ..., b_n)$ are two tuples, then there is an L-formula $\delta(x, w)$, such that $X = \delta(\mathbf{C}, c)$, where $c = a \cap b$ when viewed as sets, i.e., elements in c are elements that appears in both a and b.*

Proof. We only need to prove the following claim: if b_k does not appear in the tuple a, then there is an $L_{b'}$-formula defining X, where $b' = (b_1, ..., b_{k-1}, b_{k+1}, ..., b_n)$. We can reach the conclusion of the lemma by repeatedly taking away elements in b that are not in a.

Without loss of generality, assume $a_1 < ... < a_m$ and $b_1 < ... < b_n$.

If no a_i falls into the open interval (b_{k-1}, b_{k+1}), then by sufficient saturation, for any $c \in (b_{k-1}, b_{k+1})$, there exists an automorphism α which fixes every element in a and b', but takes b_k to c. Since this automorphism fixes the tuple a, it fixes X. This means that for every $c \in (b_{k-1}, b_{k+1})$, $\psi(x, (b_1, ..., b_{k-1}, c, b_{k+1}, ..., b_n))$ defines X, so actually the $L_{b'}$-formula $\forall z(b_{k-1} < z < b_{k+1} \to \psi(x, (b_1, ..., b_{k-1}, z, b_{k+1}, ..., b_n)))$ defines X.

Now we drop the assumption on a_i and try to prove the same thing. Let $c \in (b_{k-1}, b_{k+1})$. Without loss of generality assume $c < b_k$. Let α be an automorphism of $\mathbf{C}$ which fixes b, fixes every a_i which lies outside (b_{k-1}, b_{k+1}), and moves all the a_i's in (b_{k-1}, b_{k+1}) outside the interval (c, b_k). Since α fixes b, it fixes X, so $\phi(x, \alpha(a))$ also defines X. Now let β be an automorphism that fixes $\alpha(a)$ and b' and takes b_k to c, which exists as $\alpha(a_i)$ is now outside the interval (c, b_k), for any i. Since β fixes $\alpha(a)$, it fixes X. Thus $\beta \circ \alpha$ is an automorphism that fixes X, fixes b', and takes b_k to c. So $\psi(x, (b_1, ..., b_{k-1}, c, b_{k+1}, ..., b_n))$ still defines X, and we get again that the $L_{b'}$-formula $\forall z(b_{k-1} < z < b_{k+1} \to \psi(x, (b_1, ..., b_{k-1}, z, b_{k+1}, ..., b_n)))$ defines X. □

Proposition A.2. *DLO eliminates imaginaries.*

Proof. For any definable set X, let a be a minimal parameter for X in the sense that X is definable over a but not definable over any proper subtuple of a. Let us index $a = (a_1, ..., a_k)$ such that $a_1 < a_2 < ... < a_k$. Suppose $X = \phi(\mathbf{C}, a)$ where $\phi(x, y)$ is an L-formula. We claim that a is a code of X witnessed by the formula $\phi(x, a) \wedge (a_1 < a_2 < ... < a_k)$. Suppose $X = \phi(\mathbf{C}, a') \wedge a'_1 < a'_2 < ... < a'_k$. By Lemma A.1 and minimal choice of a, $\{a'_1, ..., a'_k\} = \{a_1, ..., a_k\}$, and hence $a' = a$ as we have $a_1 < a_2 < ... < a_k$ and $a'_1 < a'_2 < ... < a'_k$. □

Remark A.3. The above corollary shows that any theory satisfying Lemma A.1 has *weak elimination of imaginaries*, i.e., for every definable set X there is a tuple c of parameters and a corresponding formula $\phi(x, y)$ such that $\phi(x, c)$ defines X, and $\phi(x, c')$ defines X iff c' is a permutation of c (hence c is actually more like a *set* of parameters rather than a tuple).

References

[1] Wilfrid Hodges. *Model Theory.* Cambridge University Press, New York, 1993.

[2] Serge Lang. *Introduction to Algebraic Geometry.* Addison-Wesley Publishing Company, Reading, MA, 1996.

[3] David Marker. *Model Theory: An Introduction.* Springer-Verlag, New York, 2002.

[4] Rahim Moosa. *Set Theory and Model Theory.* Unpublished Notes, 2012.

[5] Anand Pillay. *Geometric Stability Theory.* Oxford University Press, New York, 1996.