EXPLORATIONS IN P-ADIC MODEL THEORY

EXPLORATIONS AND FORMALIZATIONS IN P-ADIC MODEL THEORY

By AARON CRIGHTON, B.Sc.

A Thesis Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree of Doctor of Philosophy

McMaster University DOCTOR OF PHILOSOPHY (2022) Hamilton, Ontario (Mathematics)

TITLE: Padic Model Theory Thesis

AUTHOR: Aaron Crighton, B.Sc. (University of Toronto)

SUPERVISOR: Professor Deirdre Haskell

NUMBER OF PAGES: ix, 129

Lay Abstract

This thesis explores theoretical and computational aspects of the p-adic numbers, which are fundamental objects of study in number theory. The thesis approaches these problems using techniques from model theory, which is a branch of mathematical logic which is particularly effective at analyzing algebraic structures. The thesis has two parts. The first is purely theoretical and focuses on the proof of a theorem about p-adic functions which can be defined in certain logical languages. The second part takes a computational approach and focuses on developing theoretical results about p-adic numbers using a special software system called a proof assistant (the particular proof assistant is called Isabelle). By developing theory of p-adics in Isabelle, we can use software to automatically verify the correctness of proofs of results and coherence of definitions. The resulting libraries can then be imported and used in future developments of machine-verified mathematical theory for more complex results.

Abstract

This thesis explores various aspects of the model theory of p-adic fields. It is divided into two distinct parts. The first part pertains to the theory of P-minimal structures. The main focus is exploring a class of P-minimal structures which display a certain tameness property with respect to the approximation of definable functions by their Taylor polynomials, and builds to a proof of a theorem for passing from local approximations by Taylor polynomials to global (piecewise-definable) approximations of functions by their Taylor polynomials in such structures. The final chapter of this part discusses some aspects of classifying the scope of the class of structures that this theorem applies to. The second part of the thesis describes a formally verified proof of Macintyre's quantifier elimination theorem for p-adic fields in the Isabelle proof assistant. The algebraic formalisations to required state and prove this theorem are outlined, including constructions of the p-adic integers and fields, as well as a formally verified proof of Hensel's Lemma.

Acknowledgements

I would like to start by acknowledging the support and patience of Tashy Davidson, Mary Crighton, and Peter Crighton during my studies. Their love and support was crucial to my completing what was a circuitous journey out of my PhD studies.

To my supervisor Deirdre Haskell, I thank you for guiding me towards the completion of this thesis. Your support and patience as my interests oscillated between topics helped me to find topics that I wanted to study, and your feedback and advice as I worked through my research made its completion possible.

To Patrick Speissegger, Bradd Hart, and Matt Valeriote of the McMaster logic group, I thank you enriching my experience through the courses, seminars, and reading groups which helped shape my thinking on the topics outlines in this thesis. Additional thanks to Patrick for serving on my supervisory committee and providing feedback and support on the content of this thesis.

I would finally like to thank Wolfram Kahl both for his contributions to this thesis as a member of my supervisory committee, and for his support and encouragement as my research interests shifted late in my studies. Your classes and support helped open up research avenues that helped reinvigorate my passion for my work, and made the second part of thesis possible.

Contents

Part I

Approximating Definable Functions by Taylor Polynomials in P-minimal Structures

Chapter 1

Introduction

The p-adic numbers are fundamental objects of study in number theory. They provide, along with the real numbers, the only metric completions of the rational numbers on the standard absolute value. The real numbers and first-order structures which can be imposed on the reals have been an object of intense study since the inception of model-theoretic techniques. Among the earliest results include Tarski's quantifier elimination theorem for the elementary theory of the real numbers in the language of ordered fields [31]. Paul Cohen later produced a simplified proof of Tarski's Theorem in [11] and showed that similar ideas could be applied to prove a quantifier elimination result for p-adic fields. Later, in [26] Macintyre showed that one could prove a quantifier elimination result for any finite extension of a p-adic field, in a modified first order language which allowed for a more refined description of the underlying definable sets one could produce. In [18], Denef proved a cell decomposition theorem for definable subsets of $\mathbb{Q}_p^m$ in Macintyre's language which gave strong insights into the behaviour of the definable functions in this language. Denef was able to use this method to provide a powerful technique for the computation of p-adic integrals, yielding a siginificantly simpler proof that the Poincaré series associated to a p-adic variety is represented by a rational function. Denef's results also strengthened the analogy between the first order theory of p-adic fields in Macintyre's language and o-minimal expansions of real closed fields. This analogy was subsequently strengthened and formalized by Haskell and Macpherson in [21], with the introduction of the theory of P-minimal structures, which provided a class of structures which satisfied a minimality property with respect to Macintyre's language for valued fields which closely mirrors the definition of o-minimality in the context of ordered fields.

This thesis makes two contributions to the theory of P-minimal structures. In the first section, we prove a theorem on uniform Taylor approximations of definable functions in P-minimal structures. In the second section, we provide a machine-verified formalisation of Macintyre's quantifier elimination theorem for p-adic fields using the Isabelle proof assistant.

The theorem proved in the first section applies to P-minimal theories which also satisfy a stronger property known as *Hensel-minimality.* In [7], Cluckers, Comte, and Loeser proved a theorem analogous to the Pila-Wilkie theorem for counting points of bounded height on an o-minimal structure for certain non-archimedean valued fields, including analytic expansions of p-adic fields. A key step in this proof requires one to produce C^r parametrizations of definable sets whose C^r norms can be uniformly bounded on their domains. This in turn required showing that a function which is well-approximated locally by its degree r Taylor polynomial must also have this property globally on the pieces of a finite partition of its domain. This result itself generalizes similar results proved by the same authors in [6] for definable functions which are locally Lipschitz. Our generalization of the approximation lemma is a step towards situating the main theorems of [7] in the more general P-minimal context.

The second section of this thesis focuses on formalizing Denef's p-adic Cell Decomposition theorem and Denef's proof of Macintyre's p-adic quantifier elimination theorem (as exposited by Denef in [19]) in the Isabelle proof assistant. Formalisations of ordinary mathematics in proof assistants such as Isabelle, Lean, and Coq have become an increasing preoccupation of the mathematical community in the 21^{st} century, with a variety of motivations. As Jeremy Avigad and John Harrison observe in a survey on the subject [1], concerns over correctness and what can be said to constitute a correct proof have long been discussed in mathematics. Progress was made on clarifying this issue with the increased focus on rigorous definitions in the 19^{th} century, and the development of formal axiomatic systems in the early 20^{th} century. These projects were ultimately limited by the realization that to work fully within these systems to produce complete derivations of complex results from first principles would be hopeless by hand, and at best one could use these systems as guidelines for producing approximately rigorous proofs, where a formal system could be used for rectifying ambiguity when our intuitive faculties could not. However, with the advent of modern computing, we now have the tools to digitally implement these formal systems, and are increasingly able to automate the tedious details that previously left complete formalisations of complex mathematical results beyond our grasp. In order to make such a project practical and useful, one needs a robust library of basic mathematical results which future developments can import and use without the need to reprove them. For Isabelle, such a library exists both in libraries which come with the standard distribution of Isabelle as as well as in the Archive of Formal Proofs (AFP) which can be accessed at `https://www.isa-afp.org/`. The AFP is an online library which accumulates developments of various authors in Isabelle, and one can download the archive locally for use in their own Isabelle developments. Our own results in this thesis have been submitted to the archive, with some currently published [14].

Chapter 2

P-minimality

This section will introduce the notion of P-minimality and establish the basic algebraic and model-theoretic properties of P-minimal structures.

2.1 P-adically closed fields and Macintyre's language

Definition 2.1.1 (Haskell-Macpherson [21])**.** Let K be a valued field with valuation ν, value group Γ_K, valuation ring $\mathcal{O}$ and residue field $\overline{K}$. Let p be a prime integer. We say that (K, ν) is p-valued if $\overline{K}$ has characteristic p. K is of characteristic 0, and $\mathcal{O}/p\mathcal{O}$, viewed as a vector space over $\mathbb{F}_p$, is finite dimensional. If (K, ν) is p-valued, then we say that the p-rank of K is the dimension of $\mathcal{O}/p\mathcal{O}$ as a vector space over $\mathbb{F}_p$.

Definition 2.1.2. [21] The language of p-adically closed fields of p-rank d, denoted $\mathcal{L}_d$ is the one-sorted first order language consisting of the language of fields $(+, -, \cdot, 0, 1)$ with a binary predicate Div, unary predicates P_n for each positive integer n, and constant symbols $c_1, \ldots, c_d$.

For a p-valued field (K, ν) of rank d, the intended interpretation of the symbols in $\mathcal{L}_d$ is as follows:

1. The symbols $(+, -, \cdot, 0, 1)$ are interpreted as the usual field operations and constants on K.

2. The predicate $Div(x, y)$ holds if and only if $\nu(x) \leq \nu(y)$. That is, if and only if $y/x \in \mathcal{O}$.

3. The predicates P_n define the set of n-th powers in the p-adic closure of K, i.e. $P_n(x)$ holds if and only if $x = y^n$ for some $y \in K^\times$.

4. The constants $c_1, ..., c_d$ are elements of $\mathcal{O}$ such that the residues $c_i + p\mathcal{O}$ form a basis for $\mathcal{O}/p\mathcal{O}$ over $\mathbb{F}_p$.

Definition 2.1.3. A valued field (K, ν) is called *p-adically closed* if it is p-valued, and there is no algebraic extension L of K which is p-valued with the same p-rank. A valued field (K, ν) is called *henselian if for every algebraic extension L/K, there is a unique extension of ν to L.*

These two closure properties for valued fields are connected via the following theorem:

Fact 2.1.1. (Prestel-Roquette [29]) A p-valued field is p-adically closed if and only if it is henselian and its value group is a $\mathbb{Z}$-group.

In this result, an ordered abelian group Γ is defined to be a $\mathbb{Z}$-group if it is elementarily equivalent to the group $\mathbb{Z}$ as an ordered abelian group. Our particular choice of first-order language for p-adically closed fields is justified by the following theorem, which generalizes the p-adic quantifier elimination results of Macintyre [26]:

Fact 2.1.2. [29] Suppose (K, ν) is a p-adically closed field of rank d. Then (K, ν) admits elimination of quantifiers in $\mathcal{L}_d$.

In particular, the predicates P_n are used to eliminate the existential quantifier in formulas of the form $\exists y(x = y^n)$, and the constants c_i are necessary to express that the dimension of the residue field is exactly d, as this requires the assertion of the existence of a basis.

While our first definition of a henselian valued field K is formulated in terms of the uniqueness of extensions of valuations, one can formulate an equivalent notion which is readily expressed as a first-order axiom schema in $\mathcal{L}_d$. That is, a valued field K, with valuation ν and valuation ring $\mathcal{O}$ is Henselian if and only if it satifies Hensel's Lemma (see [20] Theorem 4.1.3 for a proof of this equivalence):

Fact 2.1.3 (Hensel's Lemma)**.** Suppose $f(x) \in \mathcal{O}[x]$ and $a \in \mathcal{O}$ such that $\nu(f(a)) > 2\nu(f'(a)$. Then there exists a unique $\alpha \in \mathcal{O}$ such that $f(\alpha) = 0$ and $\nu(a - \alpha) > \nu(f'(a))$.

The prototypical example of a Henselian p-adically closed field is the field $\mathbb{Q}_p$ of p-adic numbers. $\mathbb{Q}_p$ can be defined as the completion of the field $\mathbb{Q}$ of rational numbers with the p-adic absolute value $|\cdot|_p$, where

$$|a/b|_p := e^{\mathrm{ord}_p(b) - \mathrm{ord}_p(a)}$$

and $\mathrm{ord}_p(x)$ is defined to be the largest exponent n such that p^n divides x. Hensel's lemma was originally formulated as a theorem specifically for $\mathbb{Q}_p$ by Kurt Hensel [22] and later generalized to larger classes of valued fields.

We can close this section by offering the definition of P-minimality, as given in [21]:

Definition 2.1.4. Let $\mathcal{L}'_d$ be a language which extends $\mathcal{L}_d$, and $\mathfrak{K} = (K, \nu)$ an $\mathcal{L}'_d$-structure. We say that $\mathfrak{K}$ is P-minimal if every $\mathcal{L}'_d$-definable subset of K is definable by a quantifier-free $\mathcal{L}_d$ formula.

Haskell and Macpherson also proved that P-minimal structures are p-adically closed:

Fact 2.1.4. [21] Suppose that the $\mathcal{L}'_d$-structure (K, ν) is P-minimal. Then K is p-adically closed.

2.2 The value group of a P-minimal structure

Fact 2.1.4 tells us that P-minimal structures are p-adically closed. We can use this fact and Fact 2.1.1 to infer that the value group of a P-minimal structure will always be elementarily equivalent to the additive group of integers. In this section we will discuss a strengthening of this fact: if K is P-minimal, then it is not possible to have any extra structure on the value group beyond that imposed by the language of ordered abelian groups. The resulting complete theory of the value group is then exactly the well-understood theory of Presburger Arithmetic, which we describe below.

If we expand the language of ordered abelian groups by congruence relation symbols for the complete theory of the integers, we obtain a theory which admits quantifier elimination. This expansion is defined here.

Definition 2.2.1. The language $\mathcal{L}_{Pres}$ of Presburger arithmetic is given by the language $\mathcal{L}_{OAG}$ of ordered abelian groups, plus unary predicate symbols $x \equiv y \mod n$ for each $n > 0$.

As mentioned above, this definitional expansion of the theory of ordered abelian groups is justified by the following theorem due to Presburger:

Fact 2.2.1. The theory $Th(\mathbb{Z}, \mathcal{L}_{Pres})$ has definable Skolem functions, quantifier elimination, and is decidable.

The next theorem is due to Cluckers in [5], and guarantees that the structure of Presburger Arithetic is all one can have on the value group of a P-minimal Field.

Fact 2.2.2. Let (K, ν) be a P-minimal $\mathcal{L}_d'$-structure with value group Γ. Then every $\mathcal{L}_d'$-definable subset of Γ^m is $\mathcal{L}_{Pres}$-definable, and every $\mathcal{L}_{Pres}$-definable set $S \subseteq \Gamma^m$ is of the form $\nu(X)$ for some $\mathcal{L}_d'$-definable set $X \subseteq K^m$.

In [5], Cluckers also proves a cell decomposition theorem for $\mathcal{L}_{Pres}$-definable subsets of Γ^n, and proves that all $\mathcal{L}_{Pres}$-definable functions $f : \Gamma^n \to \Gamma$ are piecewise linear (see definition 2.2.3). This fact is frequently useful for analyzing definable sets in a P-minimal structure. We state these results precisely here as they will be useful later for understanding and manipulating P-minimal cells. In the next definition, we fix a model Γ of presburger arithemtic:

Definition 2.2.2. A Presburger (0)-cell is a singleton $\{a\} \subseteq \Gamma$. A Presburger (1)-cell is a subset $X \subset \Gamma$ of the form:

$$X = \{x \in \Gamma \mid \alpha \,\square_1\, x \,\square_2\, \beta \text{ and } x \equiv l \mod n\}$$

For constants $\alpha, \beta \in \Gamma$, natural numbers $l, n \in \mathbb{N}$, and symbols $\square_i$ which are either $\leq, <, =$, or no condition.

A concise way to characterize a Presburger (1)-cell is that it is the intersection of a definable convex subset of Γ and an arithmetic sequence. The following definition is just the familiar notion of a linear function adapted to the context of $\mathcal{L}_{Pres}$:

Definition 2.2.3. A function $f : X \subseteq \Gamma \to \Gamma$ is called linear if there exist integers l, n, c (where $0 \leq c < n$) and constant $d \in \Gamma$ such that for every $x \in X$:

$$f(x) = l\left(\frac{x-c}{n}\right) + d$$

Notice that integer division is clearly not a globally defined function in Presburger arithmetic, but on any subset the arithmetic sequence $\{x \mid x \equiv c \mod n\}$ such a function is well-defined and $\mathcal{L}_{Pres}$-definable. We can inductively define a linear function $f : X \subseteq \Gamma^{n+1} \to \Gamma$ as a function of the form:

$$f(x,t) = l\left(\frac{t-c}{n}\right) + d(x)$$

where x is a Γ^n variable, t is a Γ variable, and $d(x)$ is a linear function defined on a subset of Γ^n. Using this we can also inductively define Presburger $(i_1, ..., i_n, 1)$-cells as subsets of Γ^{n+1} of the form

$$\{(x,t) \mid x \in A \text{ and } \mid \alpha(x) \,\square_1\, t \,\square_2\, \beta(x) \text{ and } t \equiv l \mod n\}$$

where $A \subseteq \Gamma^n$ is a Presburger $(i_1, ..., i_n)$-cell, and α, β are definable linear functions from $A \to \Gamma$. We can define Presburger $(i_1, ..., i_n, 0)$-cells as graphs of definable linear functions $\alpha : A \subseteq \Gamma^n \to \Gamma$ where A is a Presburger $(i_1, ..., i_n)$-cell.

The Presburger cell decomposition theorem is Theorem 1 from [5] and characterizes definable subsets of Γ^n in Presburger arithmetic:

Fact 2.2.3. Let $X \subset \Gamma^m$ and $f : X \to G$ be $\mathcal{L}_{Pres}$-definable. Then there exists a finite partition $\mathcal{P}$ of X into Presburger cells, such that the restriction $f\mid_A: A \to \Gamma$ is linear for each $A \in \mathcal{P}$. Moreover, if X and f are S-definable, then the parts $A \in \mathcal{P}$ can be taken to be S-definable.

2.3 Multiplicative Subgroups of $K^\times$

The valuation map and the predicates P_n give two important ways to define multiplicative subgroups of $K^\times$. For each natural number n, the set $\{x \in K^\times \mid \nu(x) \equiv 0 \mod n\}$ will be a subgroup, and the sets $P_n \cap K^\times$ themselves are also subgroups. In this section we will explore other important $\mathcal{L}_d$-definable subgroups of $K^\times$ and their quotients.

2.3.1 Definable Angular Component Maps

Definition 2.3.1. Let (K, ν) be a valued field whose value group is a $\mathbb{Z}$-group. We say that π is a uniformizer for K if $\nu(\pi)$ is of minimal positive valuation.

Since it is clear that such an element must always exist in a P-minimal field, throughout this section we will work with a fixed P-minimal field K with a choice of uniformizer π. If our p-adically closed field has rank 1, then the natural choice of π will be the prime p itself.

Definition 2.3.2. Let (K, ν) be a valued field whose value group is a $\mathbb{Z}$-group, with valuation ring $\mathcal{O}$. Let $n > 0$ be an integer. The n-th residue ring of K is the ring $\mathcal{O}/\pi^n\mathcal{O}$. We will denote by $\text{res}_n : \mathcal{O} \to \mathcal{O}/\pi^n\mathcal{O}$ the canonical quotient map, and write R_n to denote the n^{th} residue ring.

The restrictions $\text{res}_n|_{\mathcal{O}^\times} : \mathcal{O}^\times \to R_n^\times$ are homomorphisms of multiplicative groups. The main goal of this section will be to show that there are unique extensions of these maps to definable homomorphisms $K^\times \to R_n^\times$. These maps are called angular component maps.

Lemma 2.3.1. *Each residue ring $\mathcal{O}/\pi^n\mathcal{O}$ is finite, and each element of this quotient has a $\emptyset$-definable representative in $\mathcal{O}$.*

Proof. By definition of P-minimality, the ring $\mathcal{O}/p\mathcal{O}$ is finite. Similarly, we see that for each k, the ring $\mathcal{O}/p^k\mathcal{O}$ has $\{p^j c_i | j < k, i < d\}$ as a basis (viewed as an $\mathbb{F}_p$-vector space), hence it is finite. If we set $m = \nu(p)$, we see that $p^k\mathcal{O} = \pi^{km}\mathcal{O}$, hence the rings $\mathcal{O}/\pi^{km}\mathcal{O}$ are all finite. The result follows, since $|\mathcal{O}/\pi^{km}\mathcal{O}| > |\mathcal{O}/\pi^n\mathcal{O}|$ whenever $km > n$. The above basis is $\emptyset$ -definable, which proves the second part of the lemma. □

Lemma 2.3.2. *Let $a \in K$ and $n, l \in \mathbb{N}$ such that $\nu(a) \equiv l \mod n$. There is some $u \in \mathcal{O}^\times$ and $z \in P_n$ such that $a = u\pi^l z$.*

Proof. By assumption there is an integer m such that $\nu(a) - l = nm$. Choose c such that $\nu(c) = m$ and set $z = c^n$, so that $\nu(z) = \nu(a) - l$. Then set $u = a\pi^{-l}z^{-1}$. This choice of u and z satisfy the claim of the lemma. □

Definition 2.3.3. With the same context as above, we call f a degree n angular component map for K if $f : K^\times \to (\mathcal{O}/\pi^n\mathcal{O})^\times$ is a homomorphism of multiplicative groups, $f(\pi) = 1$, and $f|_{\mathcal{O}^\times} = res_n|_{\mathcal{O}^\times}$.

The following lemma is essentially due to Denef [19], and Cluckers and Leenknegt [9]:

Lemma 2.3.3. *Suppose (K, ν) is a P-minimal field, with a uniformizer π. Then K has an $\mathcal{L}_d$-definable degree n angular component map. Furthermore this map is the unique degree n angular component map. We will denote this map by $\overline{ac}_n$.*

Proof. Let $N = |R_n^\times|$. We know that every $z \in K^\times$ can be written in the form $z = \pi^l u x^N$ for some $l \in \{0, \dots, N-1\}$, $u \in \mathcal{O}^\times$, and $x \in K^\times$. Then there is a definable homomorphism $\varphi : K^\times \to \mathcal{O}^\times P_N^\times$ given by $\varphi(z) = \pi^{-(ord(z) \mod N)}z$. If we further compose this with the natural quotient map $\mathcal{O}^\times P_N^\times \to \mathcal{O}^\times P_N^\times / P_N^\times \cong \mathcal{O}^\times/(\mathcal{O}^\times \cap P_N^\times)$ we get a homomorphism $\tilde{\varphi} : K^\times \to \mathcal{O}^\times/(\mathcal{O}^\times \cap P_N^\times)$ which maps $\pi^l u x^N \mapsto u(\mathcal{O}^\times \cap P_N^\times)$ (and where $u \in \mathcal{O}^\times$). Notice that by choice of N, the kernel of this map is contained in the kernel of the map res_n, so that the residue map descends to the quotient, to give the desired angular component homomorphism $\pi^l u x^N \mapsto res_n(u)$. This is clearly a homomorphism as defined, and also clearly restricts to res_n on $\mathcal{O}^\times$. Also, it is evident that $\varphi(\pi) = 1$, so that π is mapped to 1, as desired. Uniqueness follows from the fact that any other such homomorphism must also send $\pi^l x^N$ to 1, so it agrees with the above map at all values.

□

We can choose to define the angular component maps to send $0 \in K$ to $0 \in R_n$ to make them total, and to distinguish 0 from the field units. The degree n angular component should be thought of as a way to view the “first n digits” of the π-adic expansion of an element in K. Clearly for any $n \in \mathbb{Z}$, we can always find an element $x \in K$ with $\nu(x) = n$ and $ac_m(x) = 1$ by setting $x = \pi^n$. Despite a lack of a notion of general exponentiation for possibly non-standard elements of the value group, we can still find such witnesses in general:

Lemma 2.3.4. *Suppose $\gamma \in \Gamma$ and $n \in \mathbb{N}$. Then there exists some $x \in K$ such that $\nu(x) = \gamma$ and $ac_n(x) = 1$.*

Proof. Let $N = |R_n^\times|$ and $x \in P_n^\times$. We can write $\gamma = l + N\eta$ for some $\eta \in \Gamma$ and $l \in \{0, \dots, N-1\}$. Choosing $x \in K$ satisfying $\nu(x) = \eta$, we see that $\mathrm{ac}_n(x^N) = 1$ (as ac_n is a homomorphism to a group of order N), so $\pi^l x^N$ satisfies our requirements. □

Corollary 2.3.5. Suppose $\gamma \in \Gamma$, $n \in \mathbb{N}$, and $\xi \in R_n^\times$. Then there exists some $x \in K$ such that $\nu(x) = \gamma$ and $ac_n(x) = \xi$.

Proof. Choose some $x \in K$ with $\nu(x) = \gamma$ and $ac_n(x) = 1$, and $u \in \mathcal{O}^\times$ satisfying $res(u) = \xi$. Then ux satisfies the requirements. □

One very useful property of the angular component maps is that they allow us to parametrize open balls in the valuative topology on K, which we express in the following lemma:

Lemma 2.3.6. *Suppose (K, ν) is a P-minimal field, with degree $n > 0$ angular component map ac_n. Suppose $x, y \in K$ such that $\nu(x) = \nu(y) = \gamma$. Then $ac_n(x) = ac_n(y)$ if and only if $\nu(x - y) \geq \gamma + n$.*

Proof. Suppose we have $x, y \in K$ satisiying $\nu(x) = \nu(y) = \gamma$. We may assume neither of these elements are equal to 0. Let z be an element of K where $\nu(z) = \gamma$, and $ac_n(z) = 1$. Then we can obtain $u, w \in \mathcal{O}^\times$ such that $x = uz$ and $y = wz$. Note that $\nu(x - y) = \nu(u - w) + \nu(z)$.

First, suppose $ac_n(x) = ac_m(y)$. We may assume x and y are both elements of $K^\times$. Then $ac_n(u) = ac_n(w)$, which means that $res_n(u) = res_n(w)$. It follows that $u - w \equiv 0 (\mathrm{mod}\ \pi^n)$, hence $\nu(u - w) \geq n$. The desired result follows from the fact that $\nu(x - y) = \nu(u - w) + \nu(z)$.

Conversely, supposing that $\nu(x - y) \geq \gamma + n$, we see that $\nu(u - w) \geq m$, which means that $u \equiv w \mod \pi^m$, hence $ac_n(u) = ac_n(w)$, so the result follows. □

Up to this point, we have shown how angular component maps can be defined in terms of the valuation and n^{th} power sets. In what follows we will show that we can also recover the sets P_n from an angular component map in a definable way. The next lemmas shows that the sets $P_n^\times$ are large in the sense that these groups always contain an open neighbourhood around 1. This is an easy consequence of Hensel's Lemma:

Lemma 2.3.7. *Let $n \geq 2$, and $\lambda = 2\nu(n) + 1$. For every $u \in \mathcal{O}$, if $u \equiv 1$ mod π^λ then $u \in P_n$.*

Proof. Suppose $u \in \mathcal{O}$ and $u \equiv 1$ *mod* π^λ . Let $p(x) = x^n - u$. Then $\nu(p'(1)) = \nu(n)$ and $\nu(p(1)) = \nu(1-u) \geq \lambda > 2\nu(p'(1))$. By Hensel's lemma the result follows. □

Rather than work with the subgroups $P_n^\times$, we can define a related family of groups defined in terms of the valuation and angular components which carry the same data but can be more convenient to work with:

Definition 2.3.4. Let n, m be positive natural numbers. The set $Q_{n,m}$ is defined as:

$$Q_{n,m} := \{x \in K^\times \mid ac_m(x) = 1, \nu(x) \equiv 0 \mod n\}$$

It is clear that each $Q_{n,m}$ defines a subgroup of $K^\times$. Also note that if $ac_n(x) = ac_n(y)$, and $\nu(x) \equiv \nu(y) \mod m$, then $xy^{-1} \in Q_{n,m}$. This shows that the cosets of $Q_{n,m}$ lie in bijection with the (finite) set $\{0, ..., n-1\} \times R_m^\times$, so $Q_{n,m}$ is a finite index subgroup of $K^\times$. The next lemma will allow us to conclude the same thing for the groups $P_n^\times$.

Lemma 2.3.8. *If $\lambda = 2\nu(n) + 1$, then $Q_{n,\lambda} \subseteq P_n^\times$*

Proof. Let $M = lcm(n, |R_\lambda|)$. Using Lemma 2.3.2, any $x \in Q_{n,m}$ can be written $x = u\pi^l z$, where $l < M$, $u \in \mathcal{O}^\times$, and $z \in P_M$. We must have $ac_\lambda(z) = 1$ (since $z \in P_{|R_\lambda|}$), so $ac_\lambda(x) = ac_\lambda(u)$, which means that $u \in P_n$ by Lemma 2.3.7. Since n divides M, we must have that $z \in P_n$ as well. Finally, because $\nu(x) \equiv l \mod M$, we see that l must divide n. Then $\pi^l \in P_n$, hence $x \in P_n$. □

Corollary 2.3.9. Every group $P_n^\times$ has finite index in $K^\times$ and is a finite disjoint union of cosets of $Q_{n,\lambda}$, where $\lambda = 2\nu(n) + 1$.

2.3.2 The Language of Denef-Pas

The fact that the n^{th}-power sets can be construed as unions of the groups $Q_{n,m}$ means that a P-minimal $\mathcal{L}_d$-structure can always be interpreted in the following multi-sorted language, where extra sorts have been added for the residue rings and value group:

Definition 2.3.5. The infinite-sorted language of Denef-Pas is the first order language $\mathcal{L}_{DP}$ with home sort K for the valued field structure, residue ring sorts R_n for each $n > 0$, to be interpreted as the residue rings $\mathcal{O}/\pi^n\mathcal{O}$, and a sort Γ for the value group. The home sort K has the language of rings, plus a constant π, the residue ring sorts each have the language of rings on them, and the Γ sort is endowed with the language of Presburger arithmetic. In addition, there are functions $\mathrm{ac}_n : K \to R_n$

to be interpreted as the angular component maps on $K^\times$, and sending 0_K to 0_{R_n}, a function $ord : K \to \Gamma \cup \{\infty\}$ to be interpreted as the valuation on K, and maps $\mathrm{res}_{n,m} : R_n \to R_m$ for each $n \geq m$ to be reduction (mod m). The language $\mathcal{L}_{DP,d}$ is just $\mathcal{L}_{DP}$ with extra home sort constants $c_1, \dots, c_d$ added (to be interpreted as a preimage of a basis of R_1 over F_p).

Fact 2.1.2 tells us that the theory $PCF_{d,p}$ of p-adically closed fields of rank d can be axiomatized in the language $\mathcal{L}_d$ by axioms which state that the universe is a valued field and the constants $c_1, \dots, c_d$ induce an $\mathbb{F}_p$-basis of the residue field, the axioms of Presburger arithmetic for the value group, and an axiom schema asserting that Hensel's Lemma holds for polynomials of any degree. All of these axioms can also be stated in $\mathcal{L}_{DP,d}$. We will refer to the $\mathcal{L}_{DP,d}$ theory $PCF_{d,p}$ to mean the axioms of $PCF_{d,p}$ translated to $\mathcal{L}_{DP,d}$, along with axioms which state that the map ord is a surjective valuation, and that the maps ac_n are surjective angular component maps in the sense of Definition 2.3.3. The next proposition collects the basic information we will need to know about this theory.

Proposition 2.3.10. If the $\mathcal{L}_d$-structure (K, ν) is a model of $PCF_{d,p}$, then (K, ν) can be uniquely interpreted as an $\mathcal{L}_{DP,d}$ structure, which also satisfies $PCF_{d,p}$. Conversely, every $\mathcal{L}_{DP,d}$ structure induces a unique $\mathcal{L}_d$-structure which is a model of $PCF_{d,p}$. The $\mathcal{L}_{DP,d}$ theory $PCF_{d,p}$ eliminates quantifiers in every sort.

Proof. The first two claims follow easily from definitions and Lemma 2.3.3. Using Corollary 2.3.9, we see that every set P_n is quantifier-free definable from the sets $Q_{n,m}$, which themselves are clearly quantifier-free definable in $\mathcal{L}_{DP,d}$. The quantifier elimination claim then follows from Fact 2.1.2, Fact 2.2.2, and the fact that every residue ring R_n is finite of a fixed cardinality, with every element $\emptyset$-definable in $\mathcal{L}_{DP,d}$. □

Proposition 2.3.10 means that one could equivalently define a P-minimal structure to be a structure in a language $\mathcal{L}$ which extends $\mathcal{L}_{DP,d}$ only in the home sort, such that the $\mathcal{L}_{DP,d}$-theory of the structure is $PCF_{d,p}$, and every definable set in one home-sort variable is definable by a quantifier-free $\mathcal{L}_{DP,d}$-formula, which is the notion that we will use for the rest of this thesis.

2.4 Cells in P-minimal Structures

The main benefit of working in $\mathcal{L}_{DP,d}$ as the base language for P-minimality is that it allows one to readily define and reason about a class of sets called P-minimal cells. In particular, P-minimal cells can be parametrized by definable sets in the sorts Γ and R_n using the angular component and valuation maps.

Definition 2.4.1. An $\mathcal{L}_{Pres}$-formula $I(x;\alpha,\beta)$ is called a *convex condition* if I is of the form

$$\alpha \,\square_1\, x \,\square_2\, \beta$$

where the symbols $\square_i$ are either $\leq$, $<$, or no condition, and $\alpha, \beta \in \Gamma \cup \{\infty\}$

Definition 2.4.2. The $\mathcal{L}_{DP}$-formula $Q_{n,m}(x)$ is defined to be:

$$\nu(x) \equiv 0 \text{ mod } n \wedge \mathrm{ac}_m(x) = 1$$

The sets defined by the formulas $Q_{n,m}$ were already introduced in Section 2.3.1, and as shown there, are finite index multiplicative subgroups of $K^\times$. The next proposition expresses a useful property of these sets — that they are closed under finite intersections. This provides one of the chief advantages of working with the groups $Q_{n,m}$ over the groups $P_n^\times$, as the latter groups have a less obvious combinatorial structure. The proof is a straightforward computation.

Proposition 2.4.1. Let $N = \mathrm{lcm}(n,l)$ and $M = \max(m,k)$. Then $Q_{n,m}(K) \cap Q_{l,k}(K) = Q_{N,M}(K)$

Proof. For any $x \in K$,

$$\begin{aligned} x \in Q_{n,m}(K) \cap Q_{l,k}(K) &\iff \nu(x) \equiv 0 \text{ mod } n \wedge \mathrm{ac}_m(x) = 1 \text{ and} \\ &\quad\;\; \nu(x) \equiv 0 \text{ mod } l \wedge \mathrm{ac}_k(x) = 1 \\ &\iff \nu(x) \equiv 0 \text{ mod } N \text{ and } \mathrm{ac}_M(x) = 1 \\ &\iff x \in Q_{N,M}(K) \end{aligned}$$

□

Definition 2.4.3. A 1-cell condition is the data $C = (c,n,l,m,\xi,I,\alpha,\beta)$ where n,m are positive integers, $0 \leq l < n$, $c \in K$, $\xi \in \mathcal{O}^\times$ is $\emptyset$-definable, I is a convex condition, and $\alpha,\beta \in \Gamma \cup \{\infty\}$. Given a 1-cell condition, the associated $\mathcal{L}_{DP}$-formula $\varphi_C(x)$ is the formula:

$$I(\nu(x-c);\alpha,\beta) \wedge Q_{n,m}(\pi^{-l}\xi^{-1}(x-c))$$

While this definition makes the dependence on the formulas $Q_{n,m}$ and I explicit, the formula $\varphi_C(x)$ is equivalent to the formula

$$\alpha \,\square_1\, \nu(x-c) \,\square_2\, \beta \wedge ac_m(x-c) = \overline{\xi} \wedge \nu(x-c) \equiv l \mod n.$$

A set defined by the above formula is called a 1-cell. We will frequently abuse notation and write C to denote the 1-cell defined by φ_C. Note that for n fixed, all possible elements of the form $\pi^{-l}\xi^{-1}$ are $\emptyset$-definable, hence the 1-cell C is always $c\alpha\beta$-definable.

Definition 2.4.4. An $(n+1)$-cell condition is the data $C = (c, C', n, l, m, \xi, I, \alpha, \beta)$ where $n, m > 0$, $c : K^{n+1} \to K$ is a definable function, C' is an n-cell condition, ξ is a $\emptyset$-definable representative of R_m, I is a convex condition, and $\alpha, \beta : K^{n+1} \to \Gamma \cup \infty$ are definable. Given an $n+1$-cell condition C, the associated $\mathcal{L}_{DP}$-formula $\varphi_C(x; y)$ is the formula:

$$\varphi_{C'}(y) \wedge I(\nu(x - c(y)); \alpha(y), \beta(y)) \wedge Q_{n,k}(\pi^{-l}\xi^{-1}(x - c(y)))$$

Note that the cell associated to C is definable over any parameter set S, for which C', α, β, and c are S-definable.

Definition 2.4.5. Let $S \subseteq K^n$. A cell decomposition of S is a finite collection of cells $C_0, \dots C_n$ such that $S = \bigcup_i \varphi_{C_i}(K)$ and the sets φ_{C_i} are pairwise disjoint.

Definition 2.4.6. We say that K admits cell decomposition if every definable set $S \subseteq K^n$ has a cell decomposition.

The definition below is introduced by Mourgues in [27]. While Mourgues calls this property as having "definable selection", the functions mentioned below are also frequently referred to as "definable Skolem functions".

Definition 2.4.7. [27] Let M be an $\mathcal{L}$-structure, for $\mathcal{L}$ an arbitrary first-order language. We say that M admits definable selection if for each definable set $S \subseteq K^{n+m}$ there exists a definable function $g : \pi(S) \to K^m$ whose graph is contained in S (where $\pi : K^{n+m} \to K^n$ is the projection onto the first n coordinates).

Mourgues [27] characterized P-minimal fields for which cell decomposition is possible in terms of definable selection:

Fact 2.4.1. [27] Let K be a P-minimal structure. Then K admits cell decomposition if and only if K has definable selection.

A precise classification of which P-minimal structures have the definable selection property is not yet known, however it is now known that there do exist P-minimal structures which do not have definable selection, as discovered by Kovacsics and Nguyen in [23]. The known examples of such structures are obtained as reducts of larger structures which do have definable selection, so it is also unknown whether every P-minimal structure has a P-minimal Skolemization.

2.5 Balls in P-minimal fields

Every valued field has a notion of ball induced by the valuation. For completeness we repeat this definition below. Throughout this section, we assume K is a P-minimal structure.

Definition 2.5.1. A subset $B \subseteq K$ is called a ball if it is of the form $\{x \in K|\nu(x-c) \geq \gamma\}$ for some $c \in K$ and $\gamma \in \Gamma$. We call γ the radius of the ball B, which we will denote by $r(B)$. Given a radius γ and $c \in K$, we will write $B_\gamma(c)$ to denote the ball $\{x \in K|\nu(x-c) \geq \gamma\}$.

This lemma is a simple consequence of the ultrametric inequality and will be useful in computations involving P-minimal cells later on.

Lemma 2.5.1. *Let B, B' be disjoint balls in K, such that $B = B_\gamma(c)$ and $B' = B_{\gamma'}(c')$. Then for any $x \in B$, and $x' \in B'$, $\nu(x-x') = \nu(c-c')$.* □

The set of all balls in K will be denoted by $T(K)$. This set can be ordered by reverse-inclusion, i.e. $B \leq B'$ if and only if $B' \subseteq B$. Given two balls $B, B' \in T(B)$, there is a unique $B'' \in T(K)$ such that $B'' = \inf(B, B')$, where $\inf(B, B')$ denotes the pairwise greatest lower bound on the above ordering. If $B \leq B'$ then clearly $\inf(B, B') = B$. If B and B' are disjoint, then $\inf(B, B') = B_\gamma(c)$, where c, c' are the centres of B, B', respectively, and $\gamma = \nu(c-c')$ (this is an easy consequence of 2.5.1).

Lemma 2.5.2. *The function $r : T(K) \to \Gamma$ is $\emptyset$-definable.*

Proof. Given a ball B, $r(B) = \min\{v(x-y)|x, y \in B\}$. □

Lemma 2.5.3. *Suppose $S \subset K$ is a proper open definable subset of K. Then for every $s \in S$, there is a unique ball $B \in T(K)$ such that $s \in B$ and $B \subseteq S$ and which is maximal with respect to this containment condition.*

If B is such a ball in a set S, we will call B a maximal ball of S. Clearly any proper open definable subset of K is the disjoint union of its maximal balls.

Lemma 2.5.4. *Suppose $c \in K$, $\gamma \in \Gamma$ and ξ is a unit of R_m. Then the formula $\nu(x-c) = \gamma \wedge ac_m(x-c) = \xi$ defines the ball $B_{\gamma+m}(a)$, where a is any element of K which satisfies this formula.*

Proof. Let B denote the set defined by this formula. Fix $a \in B$, and let $x \in B$ be any other element. Then $\nu(x-c) = \nu(a-c) = \gamma$ and $\mathrm{ac}_m(x-c) = \mathrm{ac}_m(a-c)$, hence $\nu(x-a) \geq \gamma+m$, which proves that $B_{\gamma+m}(a) \subseteq B$. The opposite containment follows similarly. □

Corollary 2.5.5. Suppose $C = (c, n, k, m, \xi, I, \alpha, \beta)$ is a 1-cell and B is a maximal ball of C. Then B is of the form $B = \{x \in K|\nu(x-c) = \gamma \wedge \mathrm{ac}_m(x-c) = \xi\}$ for some γ such that $I(\gamma; \alpha, \beta)$ holds, and such that $\gamma \equiv k \mod n$. Conversely, every set of this form is a maximal ball of C. □

Given a cell C as in the corollary, we will refer to a maximal ball $B = \{x \in K | \nu(x-c) = \gamma \wedge \mathrm{ac}_m(x-c) = \xi\}$ as the maximal ball of C at height γ, and write $L_C(\gamma)$ to refer to it. Note that this ball is well-defined from the parameters of C and γ. Also note that the definable set $\varphi_C(K)$ for a cell $C = (c, n, k, m, \xi, I, \alpha, \beta)$ is entirely determined by the definable set $I(\alpha, \beta) \subseteq \Gamma$ as well as the function $L_C : I(\alpha, \beta) \cap (n\Gamma + k) \to T(K)$. Frequently this perspective is conceptually simpler for reasoning about cells.

Lemma 2.5.6. *Suppose $C = (c, n, k, m, \xi, I, \alpha, \beta)$ is a 1-cell condition and and $L_C(\gamma), L_C(\eta)$ are maximal balls of C, with $\gamma < \eta$. Then* $\inf(L_C(\gamma), L_C(\eta)) = B_\gamma(c)$. □

Definition 2.5.2. If B, B' are two distinct maximal balls of a cell C with $r(B) < r(B')$, we say they are adjacent if for all maximal balls B'' of C such that $r(B) < r(B'')$, we have $r(B') \leq r(B'')$.

2.6 Lemmas on Cell Decompositions

In sections 3 and 4 we will prove several existence results about particular kinds of cell decompositions of definable sets in a P-minimal field. This section compiles some technical lemmas for producing new cell decompositions from old ones which will be useful to that end. Of particular interest is understanding the possible relationships of two cells C and D such that $D \subseteq C$. Given such cells, we would like to understand conditions under which we may assume that D and C are defined by cell conditions which have the same center.

Definition 2.6.1. Suppose C and D are cells, with $D \subset C$. We call D a subcell of C if no two maximal balls of D are contained in the same maximal ball of C.

Definition 2.6.2. Suppose D is a cell. We call a partition $D_0 \sqcup D_1$ of D a partition by maximal balls of D if for each maximal ball B of D, either $B \subseteq D_0$ or $B \subseteq D_1$.

In other words, a partition $D_0 \sqcup D_1$ of a cell D is a partition by maximal balls if there is some partition $S_1 \sqcup S_2$ of the maximal balls of D such that $D_1 = \bigcup S_1$ and $D_2 = \bigcup S_2$.

Lemma 2.6.1. *Suppose D is a subcell of C, and C has center c. Then there exists a partition by maximal balls $D_0 \sqcup D_1$ of D such that D_0 is a cell which can be presented with center c and D_1 is the union of a finite collection of balls.*

Proof. Let $C = (c, n, l, m, \xi, I, \alpha, \beta)$, and $D = (c', n', l', m', \eta, I', \alpha', \beta')$. Let $\gamma = \nu(c - c')$. If $\nu(x - c) < \gamma$, then the ultrametric inequality tells us that $\nu(x - c') = \nu(x - c)$.

Furthermore, if $\nu(x-c) < \gamma - n'$ then we must also have that $\text{ac}_{n'}(x-c) = \text{ac}_{n'}(x-c')$. Then the cell $D \mid_{(-\infty,\gamma)} = (c, n', l', m', \eta, I', \alpha', \beta') \mid_{(-\infty,\gamma)}$ as sets. We claim that there are only finitely many maximal balls of D in $D - D \mid_{(\infty,\gamma)}$. To see this, let x, x' be elements of D, which lie in distinct maximal balls above the height of γ. Suppose without loss of generality that $\nu(x - c') > \nu(x' - c')$. Then we have that

$$\nu(c - c') < \nu(x - c')$$

which implies that

$$\nu(x - c) = \nu(c - c')$$

and similarly that

$$\nu(x' - c) = \nu(c - c')$$

This implies that x, x' lie in the same maximal ball of C, but different maximals balls of D, contrary to our assumption. Then let $D_0 = D \mid_{(-\infty,\gamma)}$ and $D_1 = D - D_0$ to get our desired decomposition. □

Lemma 2.6.2. *Let C be the cell condititon $(\alpha, \beta, n, m, \lambda)$, and $D \subset C$ a cell with center $d \neq c$. Then we can decompose D into a finite union:*

$$D = D_0 \sqcup D_1 \sqcup \bigsqcup_{i=0}^{ord(m)} B_i$$

where D_0 is a subcell centred at c, D_1 is a cell entirely contained in a maximal ball of C, and each B_i is an open ball.

Proof. Set $\gamma = \nu(c - d)$. Let $D_0 := D|_{(\gamma,\infty)}$. If some element $x \in X$ is in the cell D_0, then $\gamma < \nu(x - d)$, therefore $ord(x - c) = \gamma$. It follows that D_0 must be totally contained in the maximal ball at level γ in X. Let $D_1 = D|_{(-\infty,\gamma-\nu(m))}$. Then for any $x \in D_1$, $ord(x - d) + n < \nu(c - d)$. It follows that $ord(x - d) = \nu(x - c)$ and also that $\text{ac}_m(x - d) = \text{ac}_m(x - c)$. Then the cell D_1 can be presented with center c. If we let B_i be the ball of D at level $\gamma + i$ for $0 \leq i \leq \nu(m)$, then we get:

$$D = D_0 \sqcup D_1 \sqcup \bigsqcup_{i=0}^{ord(m)} B_i$$

as desired.

□

Chapter 3

Tameness Conditions For P-Minimal Structures

3.1 Differentiability and the Jacobian Property

In this section we will give basic defintions of differentiability in the P-minimal context, and explore the family of properties that Cluckers, Comte, and Loeser call the Jacobian Property for definable functions [6]. The Jacobian property is also closely related to the prior and more general notion of "b-minimality with preservation of balls" introduced by Cluckers and Loeser in [10]. We will discuss versions of the property which apply to specific definable functions on certain domains, as well as a version which holds for theories which says that all definable functions have this property up to definable finite partition. The Jacobian property for a definable function says that the image of any ball under the function is itself a ball, and that the radius of the image ball is strictly controlled by the valuation of the derivative of the function.

A local version of this property is given by Schikhof for complete non-archimedean fields in Proposition 27.3 of [30], which is generalized to a P-minimal context by Leenknegt and Kuijpers in [24]. In both of the above, the Jacobian property is derived from the notion of *strict differentiability*, and provides a valuable substitute for the mean value theorem in classical calculus. In particular, it is used to prove elementary facts such as that a function with a nonzero derivative is locally injective, and a function whose derivative is zero is locally constant. These facts are false in general for differentiable functions on a non-Archimedean field, but true in the presence of a Jacobian property.

To begin we note that it is possible to define the notions of limit and derivative of a function in a valued field using the valuation in place of the standard absolute value function in the obvious way. Unless stated otherwise, the definitions that follow only assume that K is a valued field with value group Γ. We will often find it useful to write the valuation multiplicatively, using absolute value bars around a field element to denote the valuation of an element. In this case we will reverse the ordering on the group, denote 0_Γ as 1, and ∞ as 0. This makes for increased readability in certain contexts where the arguments closely mimic similar arguments that one might make using an absolute value on a field.

Definition 3.1.1. Let c be an accumulation point of $X \subseteq K$, and $f : X \to K$. The limit of f at c is equal to L if for every $\gamma \in \Gamma$ there exists some $\eta \in \Gamma$ such that for all $x \in X$, if $\nu(x - c) > \eta$, then $\nu(f(x) - L) > \gamma$. In this case we write $\lim_{x \to c} f(x) = L$.

Definition 3.1.2. Let $X \subset K$ be a definable set, and $f : X \to K$ a definable function. We say that f is differentiable at $c \in X$, with derivative $l \in K$ if

$$\lim_{y \to c} \frac{f(c) - f(y)}{c - y} = l.$$

If f is differentiable at every point of X we say that f is differentiable on X and denote by $f' : X \to K$ the function sending $x \in X$ to the derivative of f at x. We recursively define $f^{(0)} : X \to K := f$ and $f^{(n+1)} : X \to K := f^{(n)\prime}$ if these functions exist.

The following definition is just the familiar notion of a Lipschitz function, adapted to the context of general valued fields:

Definition 3.1.3. Suppose K is a valued field with value group Γ, $\gamma \in \Gamma$, and $f : X \to K$, where $X \subseteq K^n$ for some n. We say that f is γ-Lipschitz on X if for all $x, y \in X$:

$$|f(x) - f(x)| \leq \gamma |x - y|$$

We say a function is locally γ-Lipschitz on X if every $x \in X$ has a neighbourhood in X on which f is γ-Lipschitz.

We can now state the definition of the Jacobian property for a function whose domain is a ball.

Definition 3.1.4. Let $F : B_1 \to B_2$ be a function with $B_1, B_2 \subset K$. We say that F has the Jacobian property if the following hold:

1. F is a bijection $B_1 \to B_2$ and B_1, B_2 are balls;

2. F is C^1 on B_1;

3. $|F'(x)|$ is constant (and finite) on B_1;

4. for all $x, y \in B_1$ with $x \neq y$, one has

$$|F'(x)||x-y| = |F(x) - F(y)|.$$

As previously stated, if a function has the Jacobian property then it is easy to track the radii of the images of balls in the domain, which is the content of the next lemma.

Lemma 3.1.1. *Suppose $f : B \to K$ has the Jacobian property (B is a ball). Let $\gamma \in \Gamma$ be the constant value $\nu(f'(x))$ for $x \in B$. Then $rad(f(B)) = \gamma + rad(B)$ (recall that $f(B)$ is a ball by the definition of the Jacobian property).*

Proof.

$$\begin{aligned} rad(f(B)) &= min\{\nu(x-y) : x, y \in f(B)\} \\ &= min\{\nu(f(x) - f(y)) : x, y \in B\} \\ &= min\{\gamma + \nu(x-y) : x, y \in B\} \\ &= \gamma + rad(B) \end{aligned}$$

□

Our interest will be in studying P-minimal structures for which every definable function satisfies the Jacobian property on every ball in its domain, up to definable partition.

Definition 3.1.5. Let K be a valued field which is an $\mathcal{L}$-structure, where $\mathcal{L}$ extends the language $\mathcal{L}_d$ for some d with value rings R_n and value group Γ. We say that K satisfies the global Jacobian property if the following statement holds for K:

Suppose $X \subseteq K \times Y$ and Y are definable sets, and

$$f : X \to K$$

is an $\mathcal{L}$-definable function. Then there exists an $\mathcal{L}$-definable decomposition $A_1, \ldots, A_n$ of X into sets A_i such that for each $i \leq n$ it is either the case that the restriction of $f(\cdot, y)$ to the fibre $A_{i,y} \equiv \{t \in K \mid (t, y) \in A_i\}$ is injective for each $y \in Y$, or it is constant for each $y \in Y$. In the case of injectivity, for each $y \in Y$, and for each ball $B \subseteq A_{i,y}$, the image $f(B, y)$ is also a ball, and $f(\cdot, y) : B \to f(B, y)$ has the Jacobian property.

3.2 Mapping Cells to Cells with the Jacobian Property

By definition, a function with the Jacobian Property maps a cell to a union of balls, one for each ball in the cell. We would now like to show that it can be arranged that after passing to a sufficiently fine cell decomposition, a function with the Jacobian Property maps cells to cells, carrying maximal balls of the domain cell to the maximal balls of the image cell. We will also be interested in performing this procedure for finite families of functions. The following definitions and lemmas will help us towards this end.

We will first state a simple lemma:

Lemma 3.2.1. *Suppose $d \notin B$ and $B' \subseteq B$, for balls B, B' where*

$$B' = \{z \in K \mid \nu(z-d) = \gamma, ac_n(z-d) = \xi\}$$

Then there exists some $m \leq n$ such that

$$B = \{z \in K \mid \nu(z-d) = \gamma, ac_m(z-d) = \xi'\}$$

where ξ' is the image of ξ under the projection from R_n to $R_{n'}$.

Proof. There is a finite chain of balls $B_0 = B' \subsetneq B_1 \subsetneq \ldots B_k$ with no intermediate balls, where B_k is the minimal ball containing both d and B'. Each of these balls (except for B_k) is of the desired form, and so B must be one of these intermediate balls. □

The next theorem is adapted from Proposition 3.11 of Cluckers, Comte and Loeser [6], with adaptations made for out slightly different axiomatic framework:

Theorem 3.2.2. *Suppose K is a P-minimal $\mathcal{L}$-structure with definable Skolem functions and which satisfies the global Jacobian property. Suppose $F : X \subseteq K \to K$ is a definable function. Then there exists a cell decomposition $\{X_i \mid i \leq n\}$ of X into cells such that for each X_i, $F|_{X_i}$ is either injective or constant. Furthermore, if $F|_{X_i}$ is injective, then $F(X_i)$ is a cell and F maps maximal balls of X_i to maximal balls of $F(X_i)$.*

Proof. By passing to a cell decomposition as in the definition of the Jacobian property for theories, we may assume that X is a cell on which F is injective, and that F has the Jacobian property on each maximal ball of X. We can take a cell decomposition $X_1, \ldots, X_n$ of $F(X)$ such that each X_i is of the form:

$$X_i = \{x \in K \mid \nu(x-d_i) \in (n_i\Gamma + k_i \cap I_i), ac_{m_i}(x-d_i) = \xi_i\}$$

Note that for any maximal ball B of X, the set $F(B)$ is a ball. It follows that each maximal ball of a set X_i must either be entirely contained within $F(B)$, or disjoint from it. Define

$$I_1 = \{i \leq n \mid d_i \in F(B) \text{ for some maximal ball } B \text{ of } X\}$$

and let $I_2 = \{1, .., n\} - I_1$. For each $i \in I_1$, let B_i be the ball of X such that $F(B_i)$ contains d_i. Note that for each $i \in I_1$ and $j \leq n$, the sets $X_j \cap F(B_i)$ and $X_j - F(B_i)$ form a partition by maximal balls of X_j, so if necessary we can pass to a finer cell decomposition of $F(X)$ where no new centers are required, and for each ball B_i, each cell in our decomposition is either entirely contained in $F(B_i)$, or disjoint from it. Appropriately numbering this new decomposition, and redefining the sets I_1 and I_2, we may assume without loss of generality that the original decomposition $X_1, \ldots, X_n$ has this property. Then (removing the balls B_i from X if necessary) we can assume without loss of generality:

1. X is a cell on which F is injective and has the Jacobian property on maximal balls.
2. $X_1, \ldots, X_n$ is a cell decomposition of $F(X)$ where X_i has center d_i.
3. Each d_i is disjoint from all of $F(X)$.

For each maximal ball B of X, choose some i such that $F(B)$ contains a maximal ball B_i of X_i. We know that B_i is of the form:

$$\{z \in K \mid \nu(z - d_i) = \gamma_{(B,i)}, ac_{m_i}(z - d_i) = \xi_i\}$$

For some $\gamma_{(B,i)} \in \Gamma$. Using 3.2.1 we can find $m \leq m_i$ and ξ such that

$$F(B) = \{z \in K \mid \nu(z - d_i) = \gamma_{(B,i)}, ac_m(z - d_i) = \xi\}$$

Then for some positive integer N, there are definable subsets $G_1, \ldots, G_N$ of Γ, definable centers $c_1, \ldots, c_N$, positive integers $M_1, \ldots, M_N$ and angular components $\nu_1, \ldots, \nu_N$ such that the set $F(X)$ is a finite disjoint union of the sets

$$\{z \in K \mid \nu(z - c_i) \in G_i, ac_{M_i}(z - c_i) = \nu_i\}$$

for $i \leq N$, and furthermore, the maximal balls of these sets are all of the form $F(B)$, where B is a maximal ball of X. By Presburger cell decomposition we can assume without loss of generality that the sets G_i are Presburger cells, hence the above sets can be taken to themselves be cells. In this case, the preimages of these cells under F form a partition by maximal balls of X, so refining further if necessary we can assume that these preimages are cells, hence we have found the desired decomposition of X to prove the theorem. □

This theorem can be generalized to any finite collection $F_1, \ldots, F_n$ of definable functions as well:

Corollary 3.2.3. Suppose K is a P-minimal $\mathcal{L}$-structure with definable Skolem functions and which satisfies the global Jacobian property. Suppose $F_i : X \subseteq K \to K$ is a definable function for $i = 0, \ldots, k$. Then there exists a cell decomposition $\{X_j \mid j \leq n\}$ of X into cells such that for each X_j and each i, $F_i \mid_{X_j}$ is either injective or constant. Furthermore, if $F_i \mid_{X_j}$ is injective, then $F_i(X_j)$ is a cell and F_i maps maximal balls of X_j to maximal balls of $F_i(X_j)$.

Proof. The proof proceeds by induction on k with the previous theorem as the base case. So we may assume without loss of generality that X is already a cell such that $F_i(X)$ is a cell for each $i < k$. Since being constant on a cell is clearly a hereditary property, we can assume without loss of generality that F_i is injective on X for each $i < k$. We can also assume without loss of generality that F_i has the Jacobian property on X for each $i \leq k$. Then any maximal ball of X is a cell which is mapped by F_i to another cell (since a ball is a cell) for $i \leq k$. Let $X_1, \ldots, X_m$ be a cell decomposition of X such that F_k is as desired on each cell X_j. If some maximal ball of X contains infinitely many balls of some cell X_j, we can pass to partitions by maximal balls, and add that ball itself as a further cell in our decomposition. In this way, applying Lemma 2.6.2 we can assume that each cell X_i is a subcell of X. If F_k is constant on X_j for some X_j, then we can apply the inductive hypothesis to X_j and obtain the desired result on each cell obtained. So without loss of generality, (by passing to some $X = X_j$ where F_k is injective, and renaming Y = X) we can reduce to the case where:

1. There exists a cell Y such that $F_i(Y)$ is a cell with center d_i and $F_i \mid_Y$ is injective for each $i < k$.

2. $X \subseteq Y$ is a subcell of Y with the same center as Y (call it c) such that $F_k \mid_X$ is injective, $F_k(X)$ is a cell with center d_k, and F_i has the Jacobian property on X for all $i \leq k$.

We can now mimic the argument in Lemma 3.2.6 to obtain the desired partition. Let H be the set:

$$H := \{rad(B) \mid B \text{ is a maximal ball of } X\}$$

This is a definable subset of the value group, and is a Presburger 1-cell. For each function F_i, there is a definable function $r_i : H \to \Gamma$ defined:

$$r_i(h) = rad(F_i(B_h))$$

where $h = rad(B_h)$ and B_h is the unique maximal ball of X of radius h. By the Jacobian property, we know that if $F_i(X)$ is a cell whose residue condition comes from residue ring R_{n_i}, then we actually have an explicit formula for $r_i(h)$:

$$r_i(h) = h + \nu(F_i'(B_h))$$

where $\nu(F_i'(B))$ denotes the constant value of F_i' on this ball. Since the image $F_i(Y)$ is a cell for $i < k$ and the image $F_k(X)$ is a cell, we know that each function r_i is linear, hence so is the function $h \to \nu(F_i'(B_h))$. It follows that the radius function induced by F_i is linear for any subcell of X as well. Then we can take a further decomposititon of X using finiteness of residue rings and presburger cell decomposition to get our desired decomposition. □

Frequently in our arguments, we will need to perform several cell decompositions successively, and this naturally raises the question as to how stable this construction is under further decompositions. More precisely, suppose C is a cell, f has the Jacobian property on C, and $f(C)$ is also a cell. Now suppose we have a cell decomposition $C_1, \ldots, C_n$ of C. We would like to understand under what conditions we can be guaranteed that $f(C_1), ..., f(C_n)$ also gives a cell decomposition of $f(C)$. Furthermore, if some C_i has the same center as the larger cell C, we would like to understand when $f(C_i)$ will have the same center as $f(C)$.

Definition 3.2.1. Suppose C is a cell. Then the set of radii of the maximal balls of C will be denoted by $Rads(C)$.

$Rads(C)$ is clearly a definable subset of Γ, and in fact is always a Presburger 1-cell. That is, it is of the form $(k\Gamma + l) \cap I$ for some definable convex set $I \subset \Gamma$. If some function f has the Jacobian Property and maps C to another cell, then f also induces a definable function $rad_{f,C} : Rads(C) \to Rads(f(C))$ mapping a value $\gamma := rad(B)$ to $rad_{f,C}(\gamma) := rad(f(B))$. In fact, this function is defined for any function f with the Jacobian property on C, since f is guaranteed to map balls to balls.

Lemma 3.2.4. *Suppose that C is a cell and f is a definable function on C such that f has the Jacobian property on C. Then there is a cell decomposition $C_1, \ldots, C_n$ of C by maximal balls such that $rad_{f,C_i} : Rads(C_i) \to \Gamma$ is linear for each i.*

Proof. Suppose without loss of generality that C is a cell of the form

$$\{x \in K \mid \nu(x) \in (k_0\Gamma + l_0) \cap I_0 \wedge ac_{m_0}(x) = \xi_0\}$$

The function $rad_{f,C} : Rads(C) \to \Gamma$ is definable, hence it is Presburger definable. By presburger cell decomposition it follows that there is a decomposition of $Rads(C) \subseteq \Gamma$ into presburger cells $D_1, \ldots, D_n$ such that $rad_{f,C} \mid_{D_i}$ is linear for each i. If we set C_i to be the restriction of C to those maximal balls whose radii lie in D_i, then $C_1, \ldots, C_n$ is our required cell decomposition.

□

Lemma 3.2.5. *Suppose that C is a cell and f is a definable function on C such that f has the Jacobian property on C and $rad_{f,C} : Rads(C) \to \Gamma$ is linear. Suppose that D is any subcell of C with the same center as C. Then the function $rad_{f,D} : Rads(D) \to \Gamma$ is a linear function. In fact, there exists a constant $M \in \mathbb{N}$ such that $rad_{f,D}(\gamma) = rad_{f,C}(\gamma - M) + M$, where M only depends on C and D (not on f).*

Proof. Using Lemma 3.1.1 we know that for any ball B in C,

$$rad(f(B)) = \nu(f'(x)) + rad(B)$$

where x is any point in B. In particular, this holds for maximal balls of C. Then we have that for any maximal ball B of C, and any $x \in B$:

$$\nu(f'(x)) = rad(f(B)) - rad(B)$$

Since both terms in the difference on the left are linear functions in the radius of B, the function mapping a radius γ of a maximal ball $B \subseteq C$ to the constant value $\nu(f'(x))$ for $x \in B$ must also be linear. Furthermore, there is some constant M such that whenever $B' \subseteq B$, where B' is a maximal ball of D and B is a maximal ball of C, then $rad(B') = rad(B) + M$ (in particular M is the difference of the angular component degrees required to specify balls in D and C respectively). It follows that for a value γ which is a radius of a maximal ball B' in D, we know that $B' \subseteq B$ where B is a maximal ball of C, hence:

$$\begin{aligned} rad_{f,D}(\gamma) &= \gamma + \nu(f'(x)) \text{ for } x \in B' \\ &= \gamma + (rad(f(B) - rad(B)) \\ &= \gamma + rad_{f,C}(\gamma - M) - (\gamma - M) \\ &= rad_{f,C}(\gamma - M) + M \end{aligned}$$

which is also linear. □

Lemma 3.2.6. *Suppose that C is a cell and $f_1, \ldots, f_n$ are definable functions such that*

1. *$f_i(C)$ is a cell for each i*

2. f_i has the Jacobian property on C for each i.

Suppose furthermore that D is a subcell of C with the same center as C. Then there is a partition by maximal balls of D into subcells $D_1, \ldots, D_m$ such that the above two properties hold for each f_i on each D_j. Furthermore, we can arrange so that for each f_i and each j, the cell $f_i(D_j)$ is a subcell of $f_i(C)$ with the same center as $f_i(C)$.

Proof. We know from Lemma 3.2.5 that there is a fixed M such that $rads_{f_i,D}(\gamma) = rads_{f_i,C}(\gamma - M) + M$ for each i. This means that whenever $B' \subseteq B$ for a maximal ball B' of D, and maximal B of C, with $\gamma = rad(B)$, we have that $f_i(B')$ is a ball in $f_i(B)$, and their radii differ by M. Then supposing

$$f_i(C) = \{x \in K \mid \nu(x - d_i) \in S \wedge \mathrm{ac}_m(x - d_i) = \xi\}$$

we have that every maximal ball of $f_i(C)$ is of the form

$$\{x \in K \mid \nu(x - d_i) = \gamma \wedge \mathrm{ac}_m(x - d_i) = \xi\}$$

and therefore every maximal ball of $f_i(D)$ must be of the form

$$\{x \in K \mid \nu(x - d_i) = \gamma \wedge \mathrm{ac}_{m+M}(x - d_i) = \xi'\}$$

for some ξ'. By partitioning the balls of D according to the possible values of ξ', and possibly partitioning them again to ensure that the heights are an arithmetic progression, we can get our desired decomposition $D_1, \ldots, D_n$ of D. □

One should note that the next lemma uses multiplicative absolute value notation for the valuations of field elements:

Lemma 3.2.7. *Suppose $f : C \to K$ is a definable differentiable function on a cell C such that $f(C)$ and $f'(C)$ are all cells centred at 0 and have the Jacobian Property on C. Then there is a rational number q such that for all $x \in C$*

$$|f(x)| = |q||f'(x)||x|$$

Proof. Let x be some point in C, and choose $y \in C$ from the same maximal ball as x so that $|x - y|$ is as large as possible (i.e. $|x - y$ is the radius of this maximal ball). Let m be the angular component condition in the definition of C, and let m' be the condition in a cell condition defining $f(C)$. Choose n, n' such that $\nu(n) = m$ and $\nu(n') = m'$. We know by the Jacobian Property that $|f(x) - f(y)| = |f'(x)||x - y|$, and therefore that $|f(x) - f(y)|$ is the radius of $f(C)$ (by choice of x, y). Furthermore, we know by 2.5.4 that $|f(x) - f(y)| = |n'||f(x)|$ and $|x - y| = |n||x|$ (using that the centres of C and $f(C)$ are both 0). Then we get:

$$\begin{aligned}|f(x)| &= |n'|^{-1}|f(x) - f(y)| \\ &= |n'|^{-1}|f'(x)||x - y| \\ &= |n'|^{-1}|f'(x)||n||x| \\ &= |q||f'(x)||x|\end{aligned}$$

where $q = n/n'$. □

3.3 Property T_r

The idea of mapping cells to cells with the Jacobian property was used in [6] to prove the following theorem:

Fact 3.3.1. Let $\epsilon > 0$ be given. Let $f : X \subseteq K^m \to K$ be an $\mathcal{L}$-definable function, where K is a finite field extension of $\mathbb{Q}_p$ for some p, and $\mathcal{L}$ is either the subanalytic or semi-algebraic language for valued fields. Suppose that f is ϵ-Lipschitz continuous locally. Then there exists a $C > 0$ and a finite definable partition of X into parts A_i such that the restriction of f to A_i is globally C-Lipschitz continuous for each i.

In fact, the proof of this result can be made in the more general context of a P-minimal $\mathcal{L}$-structure with definable Skolem functions and the Jacobian Property, as the assumptions on K and $\mathcal{L}$ are used to provide concrete contexts which obtain these more general properties. In [7], a similar question is explored under the context of Fact 3.3.1, but about functions which are locally well-approximated by their degree r Taylor polynomials at nearby points. This section will further elaborate on this property, establishing a suitable P-minimal framework for proving some of the results from [7], but without the concrete assumptions on the language and fields involved.

Definition 3.3.1. Suppose $f : X \subseteq K \to K$. We define the sup norm of f to be:

$$|f|_{\text{sup}} := \max\{|f(x)| \mid x \in X\}$$

if this quantity exists.

Note that in a model of Presburger arithmetic, any definable set which is bounded below has a definable infimum, so any definable bounded function $f : X \to K$ must have a well-defined sup norm.

Definition 3.3.2. Suppose $f : X \subseteq K \to K$ is r-times continuously differentiable. We define the C^r-norm of f to be the value:

$$|f|_r := \max_{\substack{0\leq k\leq r\\ x\in X}} \frac{|f^k(x)|}{k!}$$

As above, this value is well defined as long as $f^{(i)}$ is bounded on X for $i \leq r$. The following definition was introduced by Cluckers, Comte, and Loeser in [7].

Definition 3.3.3. Let $f : X \to K$ with $X \subset K$ be a C^r function. We say f satisfies property T_r on X if X is open, $|f|_r \leq 1$, and for every $x, y \in X$ one has:

$$|f(x) - T^{<r}_{y,f}(x)| \leq |x - y|^r$$

where $T^{<r}_{y,f}(x)$ denotes the degree $r-1$ Taylor polynomial of f, centred at y. In the case where f instead satisfies

$$|f(x) - T^{<r}_{y,f}(x)| \leq |c||x - y|^r$$

for all $x, y \in X$ and some $c \in K$, we will say that f has property T_r with coefficient c.

The first observation we can make is that polynomials have this property:

Lemma 3.3.1. *Suppose f is a polynomial with $|f|_r \leq 1$, and $X \subseteq \mathcal{O}$. Then f has property T_r on X.*

Proof. Let $n = deg(f)$. Since $f(x) = T^{<k}_{f,y}(x)$ for any $k > n$, and any y, it follows that $f(x) - T^{<r}_{f,y}(x) = 0$ if $r > n$ and if $r \leq n$:

$$\begin{aligned} f(x) - T^{<r}_{f,y}(x) &= T^{<n+1}_{f,y}(x) - T^{<r}_{f,y}(x) \\ &= (x-y)^r \sum_{k=r}^{n} \frac{f^{(k)}(x)}{k!}(x-y)^{k-r} \end{aligned}$$

We know by assumption that

$$|\frac{f^{(k)}(x)}{k!}(x-y)^{k-r}| \leq 1$$

for each $k \leq n$, so it follows by the ultrametric inequality that

$$|f(x) - T^{<r}_{f,y}(x)| \leq |x-y|^r$$

for any $x, y \in X$. □

Similarly to our abstraction of the global Jacobian property for theories, whereby we identify those theories where all functions satisfy the Jacobian property up to definable partition, we introduce a similar abstraction for property T_r.

Definition 3.3.4. A theory T is T_r-tame if for every definable function $f : X \subseteq \mathcal{O} \to K$, where $|f|_r \leq 1$: If f has property T_r locally on X then, there is a cell decomposition $\mathcal{C}$ of X and an integer m such that for any cell $C \in \mathcal{C}$, either f is constant on C, or for every maximal ball $B \subset C$ and $i \leq r$:

1. $|f^{(i)}(x)|$ is constant as $x \in B$ varies
2. If $x, y \in B$, then
$$|f(x) - T^{<r}_{f,y}(x)| \leq |m^{-1}||x-y|^r$$

In particular, property 2 implies that f has property T_r with coefficient $1/m$ on each maximal ball of each cell C. The main examples of theories with the Jacobian property are the theory of a finite extension of the p-adics in Macinytre's semi-algebraic language, or the same field with an analytic expansion, which are also T_r-tame. The idea behind this property is that we can pass from a function having property T_r locally on a set to having an actual infinite definable family of neighbourhoods on which f has this property globally, with these neighbourhoods being parametrized by the auxiliary sorts of K. What we will see later on that we can situate the class of P-minimal and T_r-tame theories in the more general setting of Hensel minimality.

Our main motivation for introducing T_r-tameness is to prove the following theorem, which can be viewed as a generalization of Fact 3.3.1, and is later restated as Theorem 3.4.1:

Theorem. *Suppose K is P-minimal with definable Skolem functions, and T_r-tame for some fixed r. Suppose X is a definable subset of K, $f : X \to K$ satisfies $|f|_r \leq 1$, and f has property T_r locally on X. Then there is a cell decomposition $C_1, \ldots, C_n$ of X such that for each cell C_i, f has property T_r on all of C_i.*

A version of this theorem was proved in [7] to establish the existence of T_r-parametrizations for definable sets in analytic expansions of a finite extension of $\mathbb{Q}_p$. Our goal is to provide a proof of this result which formally follows from the property of T_r-tameness and P-minimality, but does not require any concrete assumptions about the underlying structure. The next definition describes a configuration for a function $f : C \to K$, where C is a one-cell, which will be essential for proving this theorem:

Definition 3.3.5. Let $C \subset K$ be a cell, and $f : C \to K$ a definable function. We will say that f is T_r-compatible with C if the following hold:

1. f is C^r on C.
2. $f^{(i)}$ has the Jacobian property on each maximal ball of C_i for each $i \leq r$.
3. The image $f^{(i)}(C)$ is a cell for each $i \leq r$.
4. f has property T_r on each maximal ball of C.

3.4 Main Result

Our goal is to decompose an arbitrary set into one on which our function f has property T_r globally on each piece:

Theorem 3.4.1. *Suppose K is P-minimal with definable Skolem functions, and T_r-tame for some fixed r. Suppose X is a definable subset of K, $f : X \to K$ satisfies $|f|_r \leq 1$, and f has property T_r locally on X. Then there is a cell decomposition $C_1, \dots, C_n$ of X such that for each cell C_i, f has property T_r on all of C_i.*

Throughout the rest of this section, we will be adopting the background assumption that we are working in the context of this theorem. That is, we have a fixed background field K which is P-minimal with definable Skolem functions, and T_r-tame for some fixed r.

Lemma 3.4.2. *Suppose X is a definable subset of K, $f : X \to K$ satisfies $|f|_r \leq 1$, and f has property T_r locally on X. Then there is a cell decomposition $C_1, \dots, C_n$ of X such that f is T_r compatible with each cell C_i.*

Proof. By using T_r-tameness we may assume that X already is a cell for which f has property T_r on each maximal ball. By passing to a finer cell decomposition of X, using the Jacobian property, and using Corollary 3.2.3 on $f^{(i)}$ for $i \leq r$ we can also arrange that f is T_r-compatible with each cell in the decomposition. □

Lemma 3.4.3. *Suppose that C is a cell with center c, f is T_r compatible with C, and $D \subseteq C$ is a cell. Then D can be decomposed into cells $D_0 \sqcup D_1 \sqcup \cdots \sqcup D_n$, where D_0 is a subcell of C with center c and f has property T_r on D_i for $i > 0$.*

Proof. Taking the decomposition from the Lemma 2.6.2 we can break D into a subcell D_0 centered at c and finitely many sets $D_1, ..., D_n$, each of which are contained in a maximal ball of C. Since f has property T_r on maximal balls of C and property T_r is hereditary, it must also have property T_r on each piece D_i for $i > 0$. □

The next lemma is a technical result which solves a common problem in the constructions that will follow. We start from a cell C with center c on which f is T_r compatible. Using the centers of the cells $f^{(i)}(C)$ as well as c as input data, we then construct some related function h. We can pass to a cell decomposition $C_1, \dots, C_n$ of C, so that h is T_r-compatible with each C_i. The problem now is that the images $f^{(i)}(C_j)$ may no longer be cells. If we refine these so that they are cells, we then may lose this property for h, and furthermore the centers of the cells may change, meaning our definition of h should change as well (since it depends on the centers of the image cells). This lemma allows us to iteratively refine cell decompositions to get T_r compatibility for new functions, without losing this property for existing functions and without changing the centers of image cells.

Lemma 3.4.4. *Let C be a cell with center c. Suppose f is T_r compatible with C and $D \subseteq C$ is a cell (possibly with a different center). Suppose further that some function h is T_r-compatible with D. Then there is a finite decomposition of D into cells $D_1, ..., D_n$ and $D'_1, ..., D'_k$ such that each D_i is a subcell of D with center c, f has property T_r globally on each definable set D'_i and f and h are T_r compatible with each D_i. We may also arrange so that the centers of the cells $f^{(i)}(D_j)$ are the same as the centers of the cell $f^{(i)}(C)$, and the centers of the cells $h^{(i)}(D_j)$ are the same as the centers of the cell $h^{(}i)(D)$.*

Proof. We can first use Lemma 3.4.3 to decompose D into a subcell D_0 of D with center c, and finitely many definable pieces $D'_1, ..., D'_k$ on which f has property T_r globally. We can then apply Lemma 3.2.6 to D_0 to get a decomposition by cells $D_1, ..., D_n$ with center c with the functions $f_i = f^{(i)}$ such that each cell D_j is a subcell of D and $f^{(i)}(D_j)$, have the same centers as $f^{(i)}(D)$, respectively. The problem now is that potentially $h^{(i)}(D_j)$ may fail to be a cell with the same center as $h^{(i)}(D)$. However, since $h^{(i)}$ has the Jacobian property on D, and each D_j is a subcell of D, we know that each $h^{(i)}$ maps maximal balls of D_j to balls contained in maximal balls of $h(D)$. Then we can repartition each cell D_j by maximal balls into even finer cells such that both $f^{(i)}$ and $h^{(i)}$ map to cells with the appropriate centers.

□

The next lemma gives a pathway to showing that a function f has property T_r on a cell C, in the special case that f and all of its partial derivatives up to r map to cells which are centered at 0. Our eventual proof of the general case of theorem 3.4.1 will entail iteratively decomposing a set and manipulating the function f until we can reduce to the case of this lemma.

Lemma 3.4.5. *Suppose that $f : C \subset \mathcal{O}(K) \to K$ is a definable function, where C is a cell which is T_r-compatible with f. Suppose furthermore that C and $f^{(i)}(C)$ are* *all centered at* 0 *for* $i \leq r$*. Then there is some rational* q *such that* f *has property* T^r *with coefficient* $|q|$ *on all of* C*.*

Claim 1. For all $i < r$ and all $x \in C$, there is a rational number q_i such that $|f^{(i)}(x)| = |q_i||f^{(r)}(x)||x|^{r-i}$.

Proof of claim. This follows by iteratively applying Lemma 3.2.7, using that the functions $f^{(i)}, f^{(i+1)}, \dots, f^{(r-1)}$ all satisfy the hypotheses of that lemma. □

Claim 2. There is a fixed rational number q such that if x, y lie in different balls of C, then:

$$|f(x) - T^{<r}_{y,f}(x)| \leq |q||x-y|^r. \tag{3.1}$$

Proof of claim. Let $x, y \in C$ as stated. Then by the first claim, we obtain a rational q and have that for all $i \leq r$:

$$|f(x)| \leq |q||f^{(r)}(x)||x|^r \tag{3.2}$$

and

$$|f^{(i)}(y)| \leq |q||f^{(r)}(y)|y|^{r-i} \tag{3.3}$$

Since x, y lie in different balls, we know that $|x-y| = max\{|x|, |y|\}$, hence from (4) we get

$$\begin{aligned} \left|\frac{f^{(i)}(y)}{i!}(x-y)^i\right| &\leq |q|\left|\frac{f^{(r)}(y)}{i!}\right||y|^{r-i}|x-y|^i \\ &\leq |q||x-y|^r \end{aligned} \tag{3.4}$$

Furthermore, from (1) and the assumption of bounded C^r-norm we get:

$$|f(x)| \leq |q||x-y|^r \tag{3.5}$$

(5) and (6) give us that every term in $f(x) - T^{<i}_{y,f}(x)$ is bounded by $|q||x-y|^r$ so by the ultrametric inequality the result follows. □

Proof of lemma. Let $x, y \in C$. If x, y lie in the same maximal ball, then the T_r inequality holds by hypothesis. Otherwise, apply the second claim. □

The next lemma and its corollary will be important for reducing the proof of Theorem 3.4.1 to the case of Lemma 3.4.5. The idea is that starting from a general T_r-compatible function f on a cell C, we can iteratively perform polynomial translations of f and further cell decompositions until we obtain a new function $f+g$ and finitely many cells, such that the hypothesis of Lemma 3.4.5 are satisfied for $f+g$. Concluding that $f+g$ has property T_r on each new cell will imply the result for f.

Lemma 3.4.6. *Suppose $f : X \to K$ is a function and $h : X \to K$ has property T_r. Then f has T_r if and only if $f + h$ does.*

Proof. Notice that for any y, we have that $T^r_y(f + h) = T^r_y(f) + T^r_y(h)$. Suppose f has T_r. Then:

$$\begin{aligned} |f(x) + h(x) - T^r_{y,f+h}(x)| &= |f(x) - T^r_{y,f}(x) + h(x) - T^r_{y,h}(x)| \\ &\leq \max\{|f(x) - T^r_{y,f}(x)|, |h(x) - T^r_{y,h}(x)|\} \\ &= \max\{|f(x) - T^r_{y,f}(x)|, |x - y|^{r+1}\} \end{aligned}$$

Replacing h with $-h$ and applying the forward direction to $f + h$ gives the converse. □

We can combine this fact with Lemma 3.3.1 to obtain an important version of this fact:

Corollary 3.4.7. Suppose $f : X \to K$ is a function, $X \subseteq \mathcal{O}$, and h is a polynomial. Then f has T_r on X if and only if $f + h$ does. □

Lemma 3.4.8. *Suppose $(K, \mathcal{L})$ is a P-minimal structure with definable Skolem functions and the Jacobian property. Suppose that X is definable, and $f : X \to K$ is a differentiable definable function, and $f^{(r)} \equiv 0$ for some $r \geq 0$. Then there is a cell decomposition $X_1, \ldots X_n$ of X such that $f|_{X_i}$ is a polynomial of degree no more than r for each i.*

Proof. The proof is by induction on r. The $r = 0$ is a consequence of the Jacobian property, since there is a cell decomposition of X where the restriction of f to each cell is constant. If the result holds for some k, and $r = k + 1$, then we can find a cell decomposition where f' is a polynomial on each cell. Suppose $f'|_C = p|_C$ for some cell C and polynomial p of degree $\leq k$. Let P be any polynomial of degree $deg(p) + 1$ which is an antiderivative of p. Then $f - P$ has derivative 0 on C, so we can apply the $r = 0$ case to see that (perhaps after passing to a finer decomposition) $f|_C = (P + c)|_C$ for some constant $c \in K$, finishing the proof. □

Lemma 3.4.9. *Suppose X is a cell, and $f : X \to K$ a T_r-compatible function, such that $f^{(i)}$ is injective on X for all $i \leq r$. Then there is a cell decomposition $C_1, \ldots, C_n$ of X, an integer k, and a polynomial g of degree $\leq r$ of C_r-norm $\leq |k|^{-1}$ such that for each cell C_i, either f has property T_r globally on C_i, or $f + g$ is T_r-compatible with C_i and for each $j \leq r$, $(f + g)^{(j)}(C_i)$ is a cell centered at 0.*

Proof. Adopting the context of the lemma, we will show by induction on m that for any $m \leq r$, the result can be proved for all j^{th} derivatives of $f + g$ with $j \leq m$.

Base Case: By assumption, we know that $f^{(i)}(X)$ is a cell for each $i \leq r$. Let d_0 be the center of $f(X)$, and $g(x) = d$ (a constant polynomial). Then clearly $(f-g)(X)$ is a cell centered at 0. Since $(f+g)^{(i)} = f^{(i)}$ for $i > 0$, the singleton cell decomposition $\{X\}$ of X suffices to establish the case.

Inductive Step: Suppose $0 < m \leq r$. Suppose we have a cell decomposition $C_1, \ldots, C_n$ of X, an integer k_m, and a polynomial g of degree $\leq r$ of C_r-norm $\leq |k_m|^{-1}$ such that for each cell C_i, either f has property T_r globally on C_i, or $h := f + g$ is T_r-compatible with C_i and for each $j < m$, $h^{(j)}(C_i)$ is a cell centered at 0. We only need to perform the required decomposition on a single cell C_i to establish the case. So fix $C := C_i$ for some i. If f has property T_r globally on this cell then we are done, so we may assume we are not in this case.

Let d_m be the center of the cell $h^{(m)}(C)$. By partitioning C by its maximal balls, we may assume that either $d_m = 0$, or $|h^{(m)}(x)| = |d_m|$ for all $x \in C$. If $d_m = 0$ then there is nothing to prove, so we may assume that the latter case holds. Define

$$h_0 := h - \frac{d_m}{m!}x^m$$

and note that the monomial above has C_r-norm bounded by 1. Additionally, note that h_0 clearly has property T_r locally on C, since it is the sum of such functions. We can therefore use T_r-tameness to pass to a decomposition of C such that h_0 is T_r-compatible with each cell. Let D be a cell in this new decomposition. By Lemma 3.4.4, we can further decompose D into cells D_i such that either h (and hence f) has property T_r globally on D_i, or D_i is a subcell of D with the same center as C, both h_0 and h are T_r-compatible with D_i, and the centers of the cells $h^{(j)}(D_i)$ are the same as the centers of $h^{(j)}(C)$. Our desired result holds on those D_i for which f has property T_r, so we can now focus on further decomposing the latter kind of D_i.

Suppose now that D_i is as above. At this point that $h(D_i)$ is a cell centred at 0 for $i < m$. Further, since $h_0^{(m)} = h - d_m$, we know that $h_0^{(m)}(D_i)$ is a cell centered at 0. What remains is to modify h_0 and repartition D_i so that the images of $h_0^{(j)}$ are also cells centered at 0 for $j < m$. If these modifications are translations by polynomials of degree less than m, then we can guarantee that this won't change the fact that the image of $h_0^{(m)}$ is also centered at 0 and we will be done.

For each $i < m$, there exists a rational constant q_i such that for each maximal ball B in D_i we have for all $x \in B$ and $i < n$:

$$|h^{(i)}(x)| = |q_i||h^{(i+1)}(x)||x|$$

By induction there is some rational constant l such that

$$\begin{aligned}|h(x)| &= |l||h^m(x)||x|^m \\ &= |l||d_m||x|^m\end{aligned}$$

and similarly we can find rational constants l_i such that

$$\begin{aligned}|h^{(i)}(x)| &= |l_i||h^m(x)||x|^{m-i} \\ &= |l_i||d_m||x|^{m-i}\end{aligned}$$

for all $x \in D_i$.

Let e_j be the center of $h_0^{(j)}(D_i)$ for each $j < m$. Let i_0 be maximal such that $e_{i_0} \neq 0$ and $i_0 < m$. Without loss of generality we may assume that $|h_0^{(i_0)}(x)| = |e_{i_0}|$ for each $x \in D_i$. On the other hand we know that

$$h_0^{(i_0)}(x) = h^{(i_0)}(x) - \frac{d_m}{(m-i_0)!}x^{m-i_0}$$

So by the ultrametric inequality we see that there exists a rational number q such that for all $x \in D_i$:

$$\begin{aligned}|h_0^{(i_0)}(x)| &\leq |q||h^{(i_0)}(x)| \\ &= |ql_i||d_m||x|^{m-i_0}\end{aligned}$$

Define the function:

$$h_1(x) = h_0(x) - \frac{e_{i_0}}{i_0!}x^{i_0}$$

Passing again to a decomposition, we can assume that the image of D_i under every function mentioned so far is a cell with the same center as before, and that $h_1^{(j)}(D_i)$ is a cell for each $j \leq r$ as well. Furthermore, we see that for $m \geq j > i_0$ we must have that $h_1^{(j)}(x) = h_0^{(j)}(x)$, so that for all such j we must have that $h_1^{(j)}(C)$ is centered at 0 (by choice of i_0). Also, since

$$h_1^{(i_0)}(x) = h_0^{(i_0)}(x) - e_{i_0}$$

we may also assume that $h_1^{(i_0)}(D_i)$ is a cell centered at 0. Now we can let i_1 be maximal such that $i_1 < i_0$ and $h_1^{(i_1)}(D_i)$ is not centered at 0, and repeat this process until we have a function g which differs from f by a polynomial, and for which $g^{(i)}(D_i)$ is a cell centered at 0 for all $i \leq m$, completing the inductive step.

□

Lemma 3.4.10. *Suppose X is a cell, and $f : X \to K$ a T_r-compatible function. There is an integer n and a decomposition of X into finitely many cells C_i such that $f|_{C_i}$ has property T_r with coefficient $|n|^{-1}$ for some $n \in \mathbb{N}$.*

Proof. Since f is T_r-compatible with X, we know that $f^{(i)}$ has the Jacobian Property on X for all $i \leq r$. If any of the $f^{(i)}$ are constant on X, then by Lemma 3.4.8 we can further decompose X into pieces on which f is a polynomial, hence f will have T_r globally on each piece and we are done. So instead we will suppose that $f^{(i)}$ is injective on X for all $i \leq r$. This places us in the context of Lemma 3.4.9, so we can obtain a further decomposition of X into finitely many pieces $X_1, ..., X_n$ such that for each X_j, f either has T_r there, or there is some polynomial g_j such that $f + g_j$ is T_r-compatible with X_j, and the image $(f + g_j)^{(i)}(X_j)$ is a cell centred at 0 for $i \leq r$. The latter case implies that $f + g_j$ has property T_r globally on X_j by lemma 3.4.5, and by corollary 3.4.7 this means that f also has property T_r on X_j, which completes the proof. □

We can finally establish the main result of this section:

Proof of Theorem 3.4.1. Using Lemma 3.4.2, there is a cell decomposition $C_1, \ldots, C_n$ of X such that f is T_r compatible with each cell C_i. Then by Lemma 3.4.10, for each cell C_i we can obtain an integer m_i, natural number k_i and a cell decomposition $C_{i,1}, \ldots, C_{i,k_i}$ such that $f\ |_{C_{i,j}}$ has property T_r with coefficient $|m_i|^{-1}$ for each $j \leq k_i$. Taking $M := m_i$ where $|m_i|^{-1}$ is maximal, we see that $\{C_{i,j} \mid i \leq n, j \leq k_i\}$ is a cell decomposition of X, and on each cell of this decomposition f has property T_r with coefficient $|M|^{-1}$. □

Chapter 4

Hensel Minimality

The main theorem of the previous chapter applies to theories which are T_r-tame. We do not yet know another way to characterise these theories, but in this section we show that it includes a broad class of theories known as Hensel-minimal theories, which are introduced in [8] by Cluckers, Halupczok, and Rideau-Kikuchi. The definition of Hensel-minimality is first given for valued fields of equicharacteristic zero, and is then generalized to the mixed characteristic case (when the base field has characteristic zero). Intuitively, Hensel-minimal structures are henselian valued fields for which one dimensional definable sets admit a form of cell decomposition. The formulation in [8] relies on leading term structures, which are (parameter-definable) imaginary sorts which combine the data of a residue ring and the value group, and will be explained in the next section.

4.1 Leading Term Structures

In what follows, let K be a valued field with value group Γ, valuation ν, and valuation ring $\mathcal{O}$. Let $\lambda \in \Gamma$ such that $\lambda \geq 0$. We define the ideal I_λ of $\mathcal{O}$ to be:

$$I_\lambda := \{x \in \mathcal{O} \mid \nu(x) > \lambda\}$$

Definition 4.1.1. With the notation from above, the leading term structure $RV_\lambda^\times$ is defined to be the multiplicative group $K^\times/(1+I_\lambda)$, and we let $rv_\lambda : K \to RV_\lambda := RV_\lambda^\times \cup \{\infty\}$ be the quotient map on $K^\times$, sending $0 \mapsto \infty$.

If K is interpreted as a structure in the language of valued fields, then each RV_λ, along with the leading term map rv_λ is a λ-definable imaginary sort. Note that $1 + I_\lambda$ is a multiplicative subgroup of $\mathcal{O}^\times$, hence there is a natural quotient map $RV_\lambda^\times \to \Gamma := K^\times/\mathcal{O}^\times$. Additionally the residue ring $k_\lambda := \mathcal{O}/I_\lambda$ has a natural inclusion map $k_\lambda^\times \to RV_\lambda$ defined by the map $a + I_\lambda \mapsto a(1 + I_\lambda)$.

Lemma 4.1.1. *The natural map $a + I_\lambda \mapsto a(1 + I_\lambda)$ is a well defined injective group homomorphism $k_\lambda^\times \to RV_\lambda$.*

Proof. Suppose $a, b \in \mathcal{O}$ and $a + I_\lambda = b + I_\lambda$ and $a + I_\lambda \in k_\lambda^\times$. Then $\nu(a) = \nu(b) = 0$. Obtain $i \in I_\lambda$ such that

$$a = b + j$$

Hence

$$a = b\left(1 + \frac{j}{b}\right)$$

So $a(1 + I_\lambda) = b(1 + I_\lambda)$. That this map is a homomorphism follows from well-definedness. Injectivity follows from the fact that $a = b(1 + j)$ implies that $a = b + bj$ hence $a + I_\lambda = b + I_\lambda$. □

The above implies that there is a natural short exact sequence

$$0 \to k_\lambda^\times \to RV_\lambda \to \Gamma \to 0$$

This implies that we can coherently refer to $\nu(\xi)$ where $\xi \in RV_\lambda$, and this is just defined as the value $\nu(a)$ for any $a \in K^\times$ where $\xi = a(1 + I_\lambda)$. In many cases (including the P-minimal case) this sequence splits:

Lemma 4.1.2. *Suppose K is P-minimal, $n \in \mathbb{N}^{\geq 0}$, and $\Gamma = \mathbb{Z}$. Then the sequence $0 \to k_n^\times \to RV_n \to \Gamma \to 0$ splits.*

Proof. Let π be a uniformizer for K. Then the map $\sigma : \Gamma \to RV_n$ defined by

$$\sigma(m) = \pi^m(1 + I_n)$$

is a section of the quotient map $RV_n \to \Gamma$. □

It should be noted that the above splitting map is not definable in the language of valued fields, since it relies on exponentiation, and therefore it does not follow that the above sequence splits for every P-minimal field. However we can say the following in the general P-minimal case:

Lemma 4.1.3. *Suppose K is P-minimal, $n \geq 0$, and $a, b \in K^\times$. Then $rv_n(a) = rv_n(b)$ if and only if $ac_n(a) = ac_n(b)$ and $\nu(a) = \nu(b)$.*

Proof. That $rv_n(a) = rv_n(b)$ implies $\nu(a) = \nu(b)$ has already been noted. To see that we also get $ac_{n+1}(a) = ac_{n+1}(b)$, we note:

$$\begin{aligned} rv_n\left(\frac{a}{b}\right) = rv_n(1) &\implies \frac{a}{b} = 1 + i \text{ for some } i \in I_n \\ &\implies ac_{n+1}(a) = ac_{n+1}(b) \end{aligned}$$

Conversely, if $\nu(a) = \nu(b)$ and $ac_{n+1}(a) = ac_{n+1}(b)$ then

$$\begin{aligned} \nu(\frac{a}{b} - 1) &= \nu(a - b) - \nu(b) \\ &\geq n + 1 \\ &> n \end{aligned}$$

hence $a/b - 1 \in I_n$, and therefore $rv_n(a) = rv_n(b)$.

□

This lemma means that we can therefore also meaningfully write $ac_n(\xi)$ for $\xi \in RV_n$ in the P-minimal case, where $ac_n(\xi)$ denotes $ac_n(a)$ for any $a \in K$ such that $rv_n(a) = \xi$. One key property of the leading term maps rv_λ is that the fibres of these maps parametrize open balls in the field K:

Lemma 4.1.4. *Let $\lambda \in \Gamma$ satisfy $\lambda \geq 0$, and $\xi \in RV_\lambda^\times$. Let $\eta = \lambda + \nu(\xi)$, and $c \in K$ such that $rv_\lambda(c) = \xi$. Then the fibre $rv_\lambda^{-1}(\xi) \subseteq K$ is the open ball:*

$$rv_\lambda^{-1}(\xi) := \{x \in K \mid \nu(x - c) > \eta\}$$

Proof. If some a satisfies $rv_\lambda(a) = \xi$ then $a = c(1+i)$ for some $i \in I_\lambda$. Then $a - c = ci$ which implies that $ord(a - c) = \nu(c) + \nu(i) > \eta$. Conversely, if $ord(a - c) > \eta$ then we can write:

$$a = c\left(1 + \frac{a - c}{c}\right)$$

where

$$\nu\left(\frac{a - c}{c}\right) > \eta - \nu(\xi) = \lambda$$

which shows that $rv_\lambda(a) = \xi$. □

In the case where K is a P-minimal field and n is a non-negative integer, lemma 4.1.3 gives us a more precise description of these fibres: $rv_n^{-1}(\xi)$ is exactly equal to the ball $\{x \in K \mid ac_n(x) = ac_n(\xi) \text{ and } \nu(x) = \nu(\xi)\}$.

4.2 Prepared Sets and l-ω-Minimality

Having introduced leading term structures, we can now introduce the preliminary notions needed to define Hensel-minimality.

Definition 4.2.1. Fix a valued field K, element $\lambda \geq 0$ in Γ, and a ball $B \subset K$

1. Fix an element $c \in K$. We say that B is λ-next to c if there exists some $\xi \in RV_\lambda$ such that

$$B = \{x \in K \mid rv_\lambda(x - c) = \xi\}$$

2. Fix a finite subset $C \subset K$. We say that B is λ-next to C if B is of the form

$$B := \bigcap_{c \in C} B_c$$

where each B_c is a ball which is λ-next to c.

A key property of the λ-nextedness is that if two distinct balls B and B' are both λ-next to c (or to a finite set C), then $B \cap B' = \emptyset$. Additionally, the collection of balls λ-next to c forms a partition of $K - \{c\}$, and similarly the balls λ-next to a finite set C form a partition of $K - C$.

In the P-minimal case, there is a close relationship between balls λ-next to a point c and one-cells centered at a point $c \in K$. In particular, for a non-negative integer n, the union of all balls B which are n-next to c is the finite union of cells

$$\bigcup_{\eta \in R_n^\times} \{x \in K \mid \mathrm{ac}_n(x - c) = \eta\}$$

and the maximal balls of each of the above cells are exactly the balls n-next to c.

The next definition is a general notion of cell decomposition for a valued field:

Definition 4.2.2. Let C be a finite subset of K and $X \subset K$. We say that C λ-prepares X if for any elements $x, x' \in K$ such that $rv_\lambda(x - c) = rv_\lambda(x' - c)$ for each $c \in C$, either $x \in X$ and $x' \in X$, or $x \notin X$ and $x' \notin X$.

In other words, C λ-prepares X if X is a union of balls which are λ-next to C. In the P-minimal case, and fixing a nonnegative integer n, one can use Presburger cell decomposition to show that a definable set X is n-prepared by a finite set C if and only if there is a cell decomposition of X into cells with centers $c \in C$ which are defined by degree $n + 1$ angular components.

We can now state the definition of hensel minimality which will be of concern in our context, and comes from definition 1.1.4 in [8]:

Definition 4.2.3. Let $\mathcal{T}$ be a complete theory of valued fields of equi-characteristic 0, in a language $\mathcal{L}$ expanding the language of valued fields. We say that $\mathcal{T}$ is *ω-h-minimal* if every model $K \models \mathcal{T}$ has the following property:

For every $\lambda \in \Gamma^{\geq 0}$, for every set $A \subset K$ and for every set $A' \subset RV_\lambda$, every $(A \cup A')$-definable set $X \subseteq K$ can be λ-prepared by a finite A-definable set $C \subseteq K$.

4.3 Equicharacteristic zero coarsenings

While the definition of hensel minimality applies only to equi-characteristic zero theories, it is possible to generalize to the mixed case.

Definition 4.3.1. Let (K, ν, Γ) be a valued field. We call another valuation ν' on K a coarsening of ν if there exists some convex subgroup $\Delta \leq \Gamma$ such that ν' is the valuation induced by composing ν with the quotient map $\Gamma \to \Gamma/\Delta$. That is:

$$\nu'(x) = \nu(x) + \Delta$$

for each $x \in K$. We say that a coarsening is nontrivial when Δ is a proper convex subgroup (hence the value group Γ/Δ is nontrivial).

If we take $K = \mathbb{Q}_p$ and ν_p to be the p-adic valuation on $\mathbb{Q}_p$, there do not exist any coarsenings of ν_p, since $\mathbb{Z}$ has no proper convex subgroups. However, there do exist fields K where $\mathbb{Q}_p \preccurlyeq K$ and K has nontrivial coarsenings (such as a saturated elementary extension). In fact, there may also exist coarsenings of such fields for which the residue field has characteristic zero.

Definition 4.3.2. Given a valued field (K, ν, Γ), define $\mathcal{O}_{K,eqc}$ to be the smallest subring of K containing $\mathcal{O}_K$ and $\mathbb{Q}$, and let ν_{eqc} be the corresponding valuation.

Note that since $\mathcal{O}_{K,eqc}$ contains $\mathbb{Q}$, the residue field must have characteristic 0 ($\mathbb{Q}$ injectively embeds into the residue field by the standard residue map). Also note that $x \in \mathcal{O}_{K,eqc}$ if and only if $x = N^{-1}y$ for some $N \in \mathbb{Z}$ and some $y \in \mathcal{O}_K$.

Example 4.3.1. Let K be some $\aleph_0$-saturated elementary extension of $\mathbb{Q}_p$. Then the value group Γ of K is an $\aleph_0$-saturated elementary extension of $\mathbb{Z}$. Suppose that $\nu_{eqc}(x) = 0$. Then we can write:

$$x = \frac{y}{N}$$

for some $y \in \mathcal{O}_K$ and some $N \in \mathbb{Z}$. We know that $\nu(N) = n$ for some integer n, since K is an elementary extension of $\mathbb{Q}_p$. Then we have:

$$\nu(x) = \nu(y) - n$$

Furthermore we know that

$$x^{-1} = \frac{z}{M}$$

for some $z \in \mathcal{O}_K$ and some $M \in \mathbb{Z}$. Then

$$y^{-1} = \frac{z}{NM}$$

So that for some $m \in \mathbb{Z}$:

$$m \geq \nu(y) \geq 0$$

and therefore

$$n + m \geq \nu(x) \geq -n$$

which shows that the convex subgroup Δ which induces ν_{eqc} must actually be the subgroup $\mathbb{Z} \leq \Gamma$. To compute the residue field k, we note that $\mathbb{Q}_p$ maps injectively to k via the residue map, and that $\nu_{eqc}(x) \in \mathbb{Z}$ if and only if $x \in \mathbb{Q}_p$ or there exists some $y \in \mathbb{Q}_p$ such that $v(x - y) > N$ for every $N \in \mathbb{N}$. In the latter case, we see that the residues of x and y must be equal, which shows that the residue map maps surjectively onto the image of $\mathbb{Q}_p$, hence $k = \mathbb{Q}_p$.

4.4 Generalizing Hensel-Minimality to Mixed Characteristic

The notion of hensel minimality in the mixed characteristic case can be defined in terms of the equicharacteristic zero coarsening:

Definition 4.4.1. Let $\mathcal{T}$ be a complete theory of valued fields of characteristic 0 (and arbitrary residue field characteristic) in a language $\mathcal{L}$ expanding the language of valued fields. We say that $\mathcal{T}$ is ω-h^{eqc}-minimal if for every model $K \models \mathcal{T}$ the following holds: If the valuation ν_{eqc} on K is non-trivial, then the $\mathcal{L}_{eqc}$-theory of K, when considered as a valued field with the valuation ν_{eqc}, is ω-h-minimal.

In Corollary 6.1.11 of [8] it is proved that we can recover preparation results for ω-h^{eqc}-minimal theories in the base language $\mathcal{L}$:

Fact 4.4.1. Assume that $Th(K)$ is ω-h^{eqc}-minimal. For any $k > 0$ and any $\mathcal{L}$-definable set

$$W \subset K \times RV_\lambda^k$$

there exists a finite non-empty $\mathcal{L}$-definable set C and an integer $m \geq 1$ such that for every ball B which is $(\lambda + \nu(m))$-next to C, the fiber $W_x := \{\xi \in RV_\lambda^k \mid (x, \xi) \in W\}$ does not depend on x when x runs over B.

This is used in [8] to deduce the result which is of primary interest in our context: theories which are ω-h^{eqc}-minimal are T_r-tame for all r:

Fact 4.4.2. Suppose that $\mathcal{T}$ is ω-h^{eqc}-minimal and let K be a model of $\mathcal{T}$. Let $f : K \to K$ be an $\mathcal{L}$-definable function and let $r \in \mathbb{N}$ be given. Then there exists a finite $\mathcal{L}$-definable set C and an integer $m \geq 1$ such that for every ball B which is $\nu(m)$-next to C, f is $(r+1)$-fold differentiable on B, $\nu(f^{r+1})$ is constant on B, and we have:

$$|f(x) - T^{\leq r}_{f,x_0}(x)| \leq \left| \frac{1}{m} \cdot f^{(r+1)}(x_0) \cdot (x - x_0)^{r+1} \right|$$

for every $x_0, x \in B$.

Corollary 4.4.1. Suppose K above is also P-minimal. Then K is T_r-tame.

Proof. Assuming that we have a function $f : X \subset \mathcal{O} \to K$ where $|f|_r \leq 1$, we can obtain a set C and integer m such that the above inequality holds on every ball B which is $\nu(m)$-next to C. We know that such balls form a partition of $K - C$. We can take a cell decomposition of $X - C$ with angular components of degree $\geq \nu(m) + 1$. Then every maximal ball of a cell in such a decomposition is contained in a ball which is $\nu(m)$-next to C, hence f is well-approximated on such balls by its Taylor polynomial, as desired. □

Finally, citing corollary 7.1.7 of [8] we obtain that the familiar examples of nice P-minimal structures are in fact hensel-minimal, and so this notion generalizes the known examples of P-minimal structures which are T_r-tame:

Fact 4.4.3. Let K be a Henselian valued field of mixed characteristic in a language $\mathcal{L}$. Then in each of the following cases, $Th(K)$ is ω-h^{eqc}-minimal:

1. $\mathcal{L}$ is the pure valued field language.
2. K is a finite field extension of $\mathbb{Q}_p$ and $\mathcal{L}$ is the sub-analytic language.

Part II

Formalization of Macintyre's Theorem in Isabelle/HOL

Chapter 5

Macintyre's Theorem in Isabelle

Fact 2.1.2 is foundational to the definition of P-minimality, and allows us to understand the structure of definable sets in the language $\mathcal{L}_d$. In this section we will outline a machine-checkable formal proof of a version of this theorem using the proof assistant Isabelle. The formalisation draws on existing libraries of algebraic formalisations in Isabelle, both from the HOL-Algebra library which comes with the standard Isabelle distribution, as well as the archive of formal proofs, an online journal which acts as a central repository of Isabelle formalisations. All the Isabelle sessions and files referred to in Part 2 of this thesis can be found in a github repository [15].

5.1 Abstract Algebra in Isabelle-HOL

The standard Isabelle distribution includes the HOL-Algebra library [3], which is a library of definitions and theorems developing the basic notions of abstract algebra. Our formalisation of Macintyre's Theorem and its algebraic prerequisites builds on this library. Here we will give a general outline of how algebraic formalisations in HOL-Algebra work.

5.1.1 Records

A record type in Isabelle is a tuple-like data structure which stores data in named fields. Here we will outline the aspects of records needed for the purposes of our formalisation, but one can refer to Section 11.6 of the Isabelle/Isar Reference Manual [33] for more detailed information. The HOL-Algebra library uses records to define algebraic structures such as partially ordered sets, monoids and rings with record types. For example, the record type for a structure containing only the data of a set can be defined as follows:

```
record 'a partial_object =
    carrier :: "'a set"
```

Having defined this record, it can be extended to richer record types by adding fields for new data. For example, if a partial object is the data of a single set, a monoid structure can be viewed as a partial object augmented with the additional data of a binary operation and constant symbol:

```
record 'a monoid = "'a partial_object" +
    mult :: "['a, 'a] ⇒ 'a" (infixl "⊗ι" 70)
    one :: 'a ("1ι")
```

and again this can be extended to define a record structure for a ring:

```
record 'a ring = "'a monoid" +
    zero :: 'a ("0ι")
    add :: "['a, 'a] ⇒ 'a" (infixl "⊕ι" 65).
```

One can create a new instance of a record by providing the required data of each field. For a monoid for instance, one must provide a set `S` whose elements are of some fixed type `'a`, as well as a binary operation `mult` of function type `'a ⇒ 'a ⇒ 'a` and a unit element `one` of type `'a`. The resulting object, which we can call `M` will have type `'a monoid`. Since the ring record extends the monoid record, a ring will simultaneously be viewed as having type `'a ring` for some fixed type `'a`, but also is viewed as having type `'a monoid_scheme`, which essentially means "monoid with extra structure".

5.1.2 Locales

While records in Isabelle/HOL carry the raw data of a structure, they cannot hold axiomatic assumptions about them. As such, one can freely define a monoid record in Isabelle whose binary operation is not associative, or whose unit element is not a member of the carrier set. Management of axiomatic assumptions about algebraic structures can be handled in Isabelle through locales. A locale in Isabelle is a proof context which can be declared and reused at will to keep track of axiomatic assumptions and definitions. For a detailed explanation of how locales in Isabelle work, one can refer to [2]. One can declare a locale in Isabelle by listing objects which are assumed to be fixed by the locale and then listing logical assumptions one would like to make about these objects. For example, the declaration for the monoid locale is given below.:

```
locale monoid =
    fixes G (structure)
    assumes m_closed [intro, simp]:
        "[x ∈ carrier G; y ∈ carrier G ] ⇒ x ⊗ y ∈
  carrier G"
    and m_assoc:
        "[x ∈ carrier G; y ∈ carrier G; z ∈ carrier G]
        ⇒ (x ⊗ y) ⊗ z = x ⊗ (y ⊗ z)"
    and one_closed [intro, simp]:
        "1 ∈ carrier G"
    and l_one [simp]:
        "x ∈ carrier G ⇒ 1 ⊗ x = x"
    and r_one [simp]:
        "x ∈ carrier G ⇒ x ⊗ 1 = x".
```

Here one could explicitly declare the type of `G` to be that of `'a monoid`, but since this has not been done, Isabelle will infer that G is a record type which has the fields of a monoid record, and therefore can be assumed to have type `'a monoid_scheme` (leaving the possibility of extra structure being present). In this way, any ring can also be viewed as an instance of this locale, and all theorems proved about monoids will apply to the multiplicative monoid of a ring. One can now prove theorems and give definitions within this locale. This allows one to avoid tedious duplication of premises in theorem and lemma statements when one wants to constantly work within a possibly very specific and complex mathematical context. Once a locale has been declared, one can prove that a particular instance of the types of objects fixed within the locales satisfy the locale axioms and then these objects will inherit all theorems proved within it, through a process called locale interpretation. Again, one may consult [2] for more detailed information on locale interpretation.

We may explicitly define a new locale to extend an old one (for example the locale for rings should extend that of monoids). One can also interpret one locale as an instance of another in a post hoc way via the sublocale command. This is necessary when a certain locale does not inherit the structure of another locale by definition, but one can prove a theorem within it that the axioms of another locale are satisfied. For example, a locale for elliptic curves would not axiomatically declare that the curve is a group, but instead would prove a theorem within the locale that a group structure can be defined, at which point one could declare the elliptic curve locale to be a sublocale of the group locale.

5.2 Denef's Proof of Macintyre's Theorem

A formal proof in Isabelle requires every intermediate result in the proof to itself be formalized in Isabelle. As a result, careful attention has to be paid to underlying mathematical tools that a particular proof requires in order to formalize. Macintyre's original proof in [26] establishes quantifier elimination using the following criterion due to Shoenfield:

Theorem 5.2.1. *Suppose $\mathcal{L}$ is a first-order language containing a constant symbol and T is an $\mathcal{L}$-theory. Then T admits elimination of quantifiers if and only if the following holds: For all, M_1, M_2, A_1, A_2, f, if $\mathcal{M}_1$ is an $\mathcal{L}$-structure, and $\mathcal{M}_2$ is an $|\mathcal{M}|^+$-saturated $\mathcal{L}$-structure, A_i is a substructure of M_i for each i, and $f : A_1 \to A_2$ is an isomorphism of $\mathcal{L}$ structures, then f can be extended to an elementary embedding of M_1 into M_2.*

A formalisation of this theorem would require drawing on concepts in model theory and set theory such as first-order languages, models, saturated models, etc. While developing this criterion and its underlying principles in Isabelle-HOL is surely a worthwhile endeavour, most quantifier elimination results for algebraic structures can be expressed and proved as purely algebraic facts, which reduces the theoretical overhead required to produce a formal proof. In the case of the p-adics, Denef proved Macintyre's theorem in [19] in an entirely algebraic manner. For this reason we have chosen this paper as the primary reference material for a formalisation of the result. As is standard in model theory, to avoid any reference to formal languages, quantifier elimination can be expressed as the closure of a certain class of subsets of powers of $\mathbb{Q}_p^n$ under projections. Furthermore, the notion of definable functions and definable sets must be replaced by an algebraically expressible version of these notions. Denef's proof assumes that the base field is either $\mathbb{Q}_p$ or some finite extension of $\mathbb{Q}_p$ but for simplicity our formalisation explicitly assumes the base field is $\mathbb{Q}_p$. This allows us the luxury of avoiding extra developments regarding finite field extensions and their properties.

In the rest of this section we will list the definitions and theorems of [19] which are most pertinent to its formalisation. First, we have the definitions of semi-algebraic sets and functions:

Definition 5.2.1. A subset of $\mathbb{Q}_p^m$ is called semi-algebraic if it is a boolean combination of subsets of the form

$$\{x \in \mathbb{Q}_p^m \mid \exists y \in K : f(x) = y^n\}$$

where $f(x) \in \mathbb{Q}_p[x]$, $x = (x_1, \ldots, x_m)$, and $n \in \mathbb{N}$, $n \geq 2$.

Definition 5.2.2. A function $f : \mathbb{Q}_p^m \to \mathbb{Q}_p$ is semi-algebraic if for every semi-algebraic subset $S \subseteq \mathbb{Q}_p \times \mathbb{Q}_p^r$, the set

$$\{(x, y) \in \mathbb{Q}_p^{m+r} \mid (f(x), y) \in S\}$$

is semi-algebraic.

Denef infers Macintyre's Theorem from two cell decomposition theorems, which are proved by a joint induction. However, Denef's notion of cell is slightly different from the one we used in Section 1, in that it omits the presence of a condition stipulating membership in a multiplicative subgroup of the field. The precise definition is given in section 8.9.1. Below we outline Denef's two cell decomposition theorems.

Theorem 5.2.2 (Cell Decomposition Theorem I)**.** *Let t be one variable and $x = (x_1, \ldots, x_m)$. Let $f(x, t)$ be a polynomial in t with coefficients which are semi-algebraic functions of x. Then there exists a finite partition of $K^m \times K$ into cells A, such that each such cell A has a center $c(x)$ such that the following holds:*

If we write $f(x, t)$ as a polynomial in $t - c(x)$:

$$f(x, t) = a_0(x) + a_1(x)(t - c(x)) + \cdots + a_i(x)(t - c(x))^i + \ldots,$$

then

$$ordf(x, t) - Min_i ord[a_i(x)(t - c(x))^i]$$

is bounded on A.

Theorem 5.2.3 (Cell Decomposition Theorem II)**.** *Let t be one variable and $x = (x_1, \ldots, x_m)$. Let $f_i(x, t)$, for $i = 1, \ldots, r$ be polynomials in t with coefficients which are semi-algebraic functions of x. Let $n \in \mathbb{N}$, $n > 0$ be fixed. Then there exists a finite partition of $K^m \times K$ into cells A, such that each such cell has a center $c(x)$ such that for all $(x, t) \in A$ we have*

$$f_i(x, t) = u_i(x, t)^n h_i(x)(t - c(x))^{\nu_i}, \text{ for } i = 1, \ldots r$$

with $ord(u_i(x, t)) = 0$, $h_i(x)$ a semi-algebraic function of x, and $v_i \in \mathbb{N}$.

These theorems suggest several important notions that need to be formalized in order to express statements of this form in the Isabelle language. We briefly describe some of these challenges below.

First, we needed to provide a formalisation of the field $\mathbb{Q}_p$ for an arbitrary prime p. Importantly, the proof of Macintyre's theorem requires us to prove Hensel's Lemma (Fact 2.1.3) for the field $\mathbb{Q}_p$. This work is described in Chapter 7. We also need to be able to reason about tuples $(x_1, \ldots, x_m)$ over the field $\mathbb{Q}_p$. While an existing library for this was available in the standard Isabelle distribution, it was underdeveloped, and significant work had to be done on this, which is described in Section 6.4. We then need to be able to reason robustly about the semi-algebraic subsets of the powers of a field. This is both to express the notion of a semi-algebraic function, and also to define and reason about p-adic cells. The basic formalisations regarding semi-algebraic sets are exposited in Section 8, and formalisations regarding cells and cell decompositions are discussed in Section 8.9

Next, we notice that we need to be able express the concept of a polynomial in one variable, whose coefficients are a specific kind of function in m variables over a field. For Theorem 5.2.2, we need to be able to take Taylor expansions of these polynomials, where the center of expansion is a semi-algebraic function $c(x)$, which will result in new coefficient functions $a_i(x)$. To express Theorem 5.2.3, these polynomials must be able to be interpreted as functions in the variables $(x_1, \ldots, x_m, t)$, which can be factored by semi-algebraic functions over certain sets. This first required generic tools for reasoning about polynomials in one or several variables, with coefficients over an arbitrary ring, which is detailed in Section 6.3. We then we needed to formalize semi-algebraic functions in m variables as a ring, which is detailed in Section 8.6.

Chapter 6

General Algebraic Developments

This chapter outlines some generic algebra whose development was necessary for the formalisation of Denef's proof. These make no particular use of p-adics themselves, but mainly pertain to arbitrary commutative rings. We do not develop any particularly deep algebra, but instead focus on constructing basic properties of polynomials in one and several variables over a commutative ring, and the very basic notions of algebraic sets. These results will be applied later to the field $\mathbb{Q}_p$, the ring $\mathbb{Z}_p$, as well as to the ring of semi-algebraic functions $f : \mathbb{Q}_p^n \to \mathbb{Q}_p$. While some material on polynomials was already available in the standard HOL-Algbera library ([3]), the main tools that were found lacking were an explicit construction of single variable Taylor polynomials over a commutative ring (section 6.2) and construction of multivariable polynomials as a ring, with evaluation maps to interpret polynomials as functions (section 6.3). The material on rings of functions in Section 6.1 and cartesian powers of a ring in section 6.4 are new, with little previously available in HOL-Algebra.

6.1 Rings of Functions

A central construction that is used in several places in the formalisation is the notion of a ring of functions from some arbitrary set `S` to the carrier of a base ring `R`. In Isabelle's logic, a function `f` of type `'a⇒'b` must be a total function mapping values of type `'a` to those of type `'b`. However, one can explicitly leave certain values of a function unspecified using a special expression named `undefined`. This allows one to define partial functions by specifying values on some elements of the universe of a type, and explicitly specifying that the function takes value `undefined` on others. The expression `undefined` can be cast to any type, but essentially nothing can be proved about it, and it behaves like a generic element of the type. Since `undefined` is a well-typed expression, pure logical statements such as `undefined = undefined` can be proved in Isabelle, but statements which depend on a specifically underlying value of this expression can never be proved. In this way we can reason about functional identity of two partial functions. Such an approach to partial functions can be found in the theory `FuncSet` from `HOL-Library` in the standard Isabelle distribution, and serves as the basis for our construction of rings of functions in the style of `HOL-Algebra`.

In our theory `Function_Ring` in [14], we define extensional function operations pointwise in the obvious way. For example we can define multiplication with the following:

```
definition function_mult:: "'c set ⇒ ('a, 'b) ring_scheme
    ⇒ ('c ⇒ 'a) ⇒ ('c ⇒ 'a) ⇒ ('c ⇒ 'a)" where
"function_mult S R f g = (λx ∈ S. (f x) ⊗_R (g x))"
```

where the notation ($\lambda x \in$ `S. f x)` is shorthand for the piecewise function whose values equal those specified by the function `f` on the set `S` and are undefined elsewhere.

One point of note is that a function not being defined at a point is not the same as that function being explicitly specified as `undefined` there. This is why the extensional guard "$\lambda x \in$ `S`" is needed in front of every operation definition. For example, we define

```
definition function_zero::
"'c set ⇒ ('a, 'b) ring_scheme ⇒ ('c ⇒ 'a)" where
"function_zero S R = (λ x ∈ S. 0_R)".
```

If we were to take some set `S` whose elements have type `'c`, and an element `x` which lies outside of `S`, then in Isabelle we can prove that

```
function_zero S R x ⊗ function_zero S R x = undefined ⊗
   undefined
```

where the expression `undefined⊗undefined` is well-typed but unspecified. We can prove an identity such as `undefined⊗undefined = undefined⊗undefined` with no trouble since this follows from the basic laws of identity. However, we cannot in general prove that `undefined⊗undefined = undefined`, unless there was a general rule that allowed us to infer `x⊗x = x` for all values of `x`.

Defining all ring constants and operations in the style above, we can then define a general function ring over a base ring. In fact, we define it as an algebra rather than just a ring, as we can perform scalar multiplication on functions:

```
definition function_ring:: "'c set ⇒ ('a, 'b) ring_scheme
    ⇒ ( 'a, 'c ⇒ 'a) module" where
"function_ring S R = (|
   carrier = extensional_funcset S (carrier R),
   Group.monoid.mult = (function_mult S R),
   one = (function_one S R),
   zero = (function_zero S R),
   add = (function_add S R),
   smult = function_scalar_mult S R |) ".
```

With this construction, we can handle many constructions over rings in a uniform way, without having to re-specify pointwise operations each time. For example, the expression `function_ring (UNIV::nat set) R` gives us the algebra of sequences over a ring, and the expression `function_ring (carrier R) R` gives us the ring of functions from the carrier set of a ring to itself.

6.2 Evaluation and Taylor Expansions of Polynomials

The HOL-Algebra library includes a theory `UnivPoly` describing polynomials in one variable. Polynomials over base ring are equated with their coefficient maps, which maps a natural number degree to the coefficient of that degree. Constructions over polynomial rings are facilitated by a generic evaluation function `eval` for which a universal mapping property is proved:

```
definition
  eval :: "[('a, 'm) ring_scheme, ('b, 'n) ring_scheme,
           'a ⇒ 'b, 'b, nat ⇒ 'a] ⇒ 'b"
  where "eval R S phi s = (λp ∈ carrier (UP R).
    ⨁_S i ∈ {..deg R p}. phi (coeff (UP R) p i) ⊗_S s [^]_S i
   )".
```

Given rings R and S, a homomorphism $\phi : R \to S$ and an element $s \in S$, `eval` maps a polynomial $a_0 + a_1x + \cdots + a_nx^n$ over R to the element $\phi(a_0) + \phi(a_1)s + \cdots + \phi(a_n)s^n$ in S. It its proved that this is the unique homomorphism $R[x] \to S$ which maps $x \to s$ and restricts to ϕ on constant values. This allows for a uniform way of handling definitions such as polynomial composition and evaluation of polynomial as a function over the base ring. For example, in [14] we defined the composition of two polynomials as:

```
definition compose where
"compose R f g = eval R (UP R) (to_polynomial R) g f"
```

where `UP R` denotes the ring of univariate polynomials over ring R, and `to_polynomial R` is the morphism mapping an element of ring `R` to its associated constant value. For evaluating a polynomial as a function we only needed:

```
definition to_function  where
"to_function R f = (λ r ∈ carrier R. eval R R (λ x. x) r f
   ".
```

We can also prove that `to_function R` is a ring homomorphism from the polynomial ring `UP R` to `function_ring (carrier R) R`. This also made it particularly easy to define the Taylor expansion of a polynomial at a given point. The Taylor coefficients of a polynomial f at a point c are just given by the coefficients of the polynomial $f(x+c)$:

```
definition taylor_expansion where
"taylor_expansion R c p = compose R p ((X_poly R) ⊕UP R
   to_polynomial R c)".
```

This construction is important for the proof of Hensel's Lemma, which requires considering the degree 1 Taylor approximation of a polynomial. We prove the following lemma:

```
lemma(in UP_cring) Taylor_deg_1_expansion':
  assumes "f ∈ carrier (UP R)"
  assumes "a ∈ carrier R"
  assumes "x ∈ carrier R"
  shows "∃a ∈ carrier R. to_fun f x =
  (to_fun f c) ⊕ (deriv f a)⊗(x⊖c) ⊕ a⊗(x ⊖ c)[^]2".
```

6.3 Polynomials in Several Variables over a Commutative Ring

It is essential for expressing and formalizing the proof of Macintyre's Theorem that we have a formalisation of multivariable polynomials over a base ring. Some work on this has been done and is included in the `HOL-Algebra` library in the theory `Indexed_Polynomials`. A mulitivariable polynomial is a finitely supported function from monomials to ring elements, and monomials are formalized as (finite) multisets over a variable set `I`. The original purpose of this formalisation was to define the algebraic closure of a field [17], which requires considering multivariable polynomials with an infinite variable set indexed by single variable polynomials. The set of polynomials over a variable set is defined, as well a function for constructing constant polynomials, addition of two polynomials, and multiplication of a polynomial by a single variable.

We extended the work on multivariable polynomials to include polynomial multiplication and scalar multiplication. Adopting the notation $\sum\limits_{\alpha} a_\alpha x^\alpha$ for a multivariable polynomial, where α is a multi-index, the product of two multivariable polynomials can be defined by the equation:

$$\left(\sum_{\alpha} a_\alpha x^\alpha\right)\left(\sum_{\beta} b_\beta x^\beta\right) = \sum_{\gamma}\left(\sum_{\alpha+\beta=\gamma} a_\alpha b_\beta\right) x^\gamma.$$

which we can state in the Isabelle formalism as:

```
definition P_ring_mult :: "('a, 'b) ring_scheme ⇒ ('a,'c)
    mvar_poly ⇒ ('a,'c) mvar_poly ⇒ 'c monomial ⇒ 'a"
   where

"P_ring_mult R P Q m =
    ⨁R x∈mset_factors m. (P x) ⊗R (Q (m - x))"
```

where `mset_factors m` denotes the set of all monomial factors of a monomial `m`. From this definition we can show that multiplication is associative and commutative. The proof of associativity is somewhat tedious. By definition we have:

$$\left(\sum_{\alpha} a_\alpha x^\alpha\right)\left(\left(\sum_{\beta} b_\beta x^\beta\right)\left(\sum_{\gamma} c_\gamma x^\gamma\right)\right) = \sum_{\epsilon}\left(\sum_{\alpha+\delta=\epsilon} a_\alpha\left(\sum_{\beta+\gamma=\delta} b_\beta c_\gamma\right)\right) x^\gamma.$$

$$\left(\left(\sum_{\alpha} a_\alpha x^\alpha\right)\left(\sum_{\beta} b_\beta x^\beta\right)\right)\left(\sum_{\gamma} c_\gamma x^\gamma\right) = \sum_{\epsilon}\left(\sum_{\delta+\gamma=\epsilon}\left(\sum_{\alpha+\beta=\delta} a_\alpha b_\beta\right) c_\gamma\right) x^\gamma.$$

So the challenge is to show that for any muliti-index ϵ, and coefficients $a_\alpha, b_\beta, c_\gamma$, we have that

$$\sum_{\alpha+\delta=\epsilon} a_\alpha\left(\sum_{\beta+\gamma=\delta} b_\beta c_\gamma\right) = \sum_{\delta+\gamma=\epsilon}\left(\sum_{\alpha+\beta=\delta} a_\alpha b_\beta\right) c_\gamma.$$

Our approach is to prove that both sides of this equation can be written as a sum over a single (product) index set, and then to show that these two single sums are equal.

Having constructed the basic algebraic operations on polynomials, we can define the algebra `Pring R I` of multivariable polynomials over a base ring `R` and variable set `I`, and prove that this structure satisfies the axioms of a commutative algebra (i.e. a commutative ring which is also a module over a base ring).

The approach to defining polynomial evaluation is incremental. In order, we:

1. Define evaluation of a monomial (i.e. a multiset) over variable set `I` to a ring element, given a function `f` mapping `I` to `carrier R`.
2. Use this to define the partial evaluation of a polynomial `f` over variables `I` on some subset `J⊆I`. The result is a polynomial over variables `I - J`.
3. Taking `J = I` above, we can define total evaluation of a polynomial in variables `I` over ring `R` to a constant polynomial. Taking the constant coefficient of this yields a value in `carrier R`.

Having constructed this, we now have a function `total_eval R g f` allowing for total evaluation of a polynomial `f` over ring `R` and a variable assignment g mapping the variables of `f` to `carrier R`. However, as is done for univariate polynomials, we would like a more general notion of evaluation which allows us to evaluate `f` given a variable assignment into some other ring `S` and a homomorphism φ from the base ring `R` to `S`. We can do this by first applying φ to the coefficients of `f`, then applying `total_eval`. We name this map "`indexed_poly_induced_morphism I S` φ `g`". We can then prove a universal mapping property for this function, which says that this is the unique morphism from $R[X_I]$ to S which respects the variable assignment $g : I \to S$ and extends the base morphism $\varphi : R \to S$:

```
lemma Pring_universal_prop:
  assumes a_cring: "cring S"
  assumes index_map: "g ∈ I → carrier S"
  assumes ring_hom: "ring_hom_ring R S φ"
  assumes "ψ = indexed_poly_induced_morphism I S φ g"
  shows "(ring_hom_ring (Pring R I) S ψ)"
        "(∀i ∈ I. ψ (mset_to_IP R {#i#}) = g i)"
        "(∀a ∈ carrier R. ψ (indexed_const a) = φ a)"
        "∀ ϱ. (ring_hom_ring (Pring R I) S ϱ) ∧
          (∀i ∈ I. ϱ (mset_to_IP R {#i#}) = g i) ∧
          (∀a ∈ carrier R. ϱ (indexed_const a) = φ a) →
          (∀x ∈ carrier (Pring R I). ϱ x = ψ x)".
```

We can use this for a number of useful constructions. For example, we can easily construct the isomorphism from $R[X_I] \to R[X_{I-\{i\}}][x_i]$ which views a polynomial in variables `I` as a univariate polynomial in the variable i over the polynomial ring in variables `I - i`. We can also construct, for disjoint variable sets I, J, the isomorphism $R[X_{(I \cup J)}] \to R[X_I][X_J]$, and show that such maps commute with evaluation of polynomials.

6.4 Powers of a Ring

One basic tool that was found lacking from the HOL-Algebra library is the general theory of finite cartesian powers of a ring and their subsets. The theory `Chinese_Remainder` from the standard distribution [16] does define the direct product of a finite list of rings, but little development is provided there that it well-suited to our purposes. The direct product of a list of rings `[R_1, ..., R_n]` is defined as the set of lists of length n whose i^{th} element lies in `carrier R_i`. Using this definition as a starting point, in the theory `Ring_Powers` we develop many of the basic properties of cartesian powers in the context of `HOL-Algebra`. Topics of main interest are defining inclusion and projection maps between powers of different dimensions, defining coordinate rings for affine spaces, defining affine algebraic sets, and showing that many basic subsets of a ring power R^n are algebraic.

6.4.1 Basics of Cartesian Powers

Since there is a pre-defined function `RDirProd_list` which maps a list of rings to their direct product ring, we can define a cartesian product first by defining a function which maps a ring `R` and a dimension `n` to the list of length `R` containing repeated copies of `R`, then passing to this function:

```
fun R_list ::
"nat ⇒ ('a, 'b) ring_scheme ⇒
    (('a, 'b) ring_scheme ) list" where
"R_list n R = map (λ_. R) (index_list n)"

definition cartesian_power ::
"('a, 'b) ring_scheme ⇒ nat ⇒ ('a list) ring"  where
"cartesian_power R n ≡ RDirProd_list (R_list n R)".
```

We also define notation $\texttt{R}^{\texttt{n}}$ as a shorthand for `cartesian_power R n`. Our formalism has the unfortunate consequence of having to distinguish between the ring `R` and its 1-dimensional cartesian product $\texttt{R}^{1}$, the latter consisting of singleton lists `[r]` for `r ∈ carrier R`.

The decision to view elements of R^n as lists rather than functions $f : \{0, ..., n-1\} \to R$ was motivated by the fact that much of the reasoning involved in model theoretic arguments about $\mathbb{Q}_p$ involve partitioning and permuting variables in sequential order. This is readily done with many of the existing functions and lemmas available in Isabelle for lists. For example, if we have points $a \in R^n$ and $b \in R^m$, we easily form the point $(a, b) \in R^{n+m}$ by list concatenation, whereas if we viewed these as functions the resulting definition would take more work.

6.4.2 Algebraic Sets

We can define the coordinate polynomial rings $R[x_0, ..., x_{n-1}]$ over R using the tools outlined in Section 6.3,

```
definition coord_ring ::
"('a, 'b) ring_scheme ⇒ nat ⇒
    ('a, ('a, nat) mvar_poly) module"
 ("_ [𝒳_]" 80) where
 "R[𝒳_n] ≡ Pring R {..< n::nat}".
```

We can then use the notation `R [𝒳_n]` to denote this ring. Since we have chosen to view points in R^n as lists rather than functions, we cannot directly evaluate a polynomial at a point using the `total_eval` function. Instead we define:

```
definition eval_at_point ::
"('a, 'b) ring_scheme ⇒ 'a list ⇒ ('a, nat) mvar_poly ⇒
   'a" where

"eval_at_point R as p ≡ total_eval R  ((!) as) p"
```

where "`(!) as i`" denotes the i^{th} element of the list `as`. Now we can define the zero set of a polynomial,

```
definition zero_set  :: "('a, 'b) ring_scheme ⇒ nat ⇒ ('a
   , nat) mvar_poly ⇒ 'a list set" where
"zero_set R n p =  {as ∈ carrier R^n. eval_at_point R as p
   = 0_R}"
```

and then the affine algebraic set corresponding to a finite set of polynomials,

```
definition affine_alg_set ::
"('a, 'b) ring_scheme ⇒ nat ⇒ ('a, nat) mvar_poly set ⇒
   'a list set" where
"affine_alg_set R n as =
    {a ∈ carrier R^n. ∀ b ∈ as. a ∈ (zero_set R n b)}"
```

and finally a predicate for algebraic sets,

```
definition is_algebraic :: "('a, 'b) ring_scheme ⇒ nat ⇒
   'a list set ⇒ bool" where

"is_algebraic R n S =
    (∃ps. finite ps ∧ ps ⊆ carrier (R[𝒳_n])
        ∧ S = affine_alg_set R n ps)".
```

A common theme that we see in these definitions and will see in many others is that the predicates and functions we would like to define for ring powers will need to take a parameter `n` for dimension of the power of `R`.

Our main interest in defining algebraic sets is that later we will prove that all algebraic sets are semi-algebraic, which will be needed in the proof of Macintyre's Theorem. It would be optimal to formulate algebraic sets in terms of polynomial ideals rather than the finite sets of generators, but for our purposes this definition suffices. A drawback of this approach is that it is not immediate that algebraic sets over domains are closed under finite unions. We can prove this by induction, with binary unions as the base case. This required defining a function which maps two finite sets of polynomials to the set of all pairwise products between them, and proving that this set is still finite.

We also included versions of set operations relativised to carriers of ring powers. For example, the function `evimage` takes the inverse image of a function, intersected with carrier R^{n},

```
definition evimage where
"evimage n f S = (f -` S) ∩ carrier R^n"
```

with notation $\mathrm{f}_{\mathrm{n}}^{-1}S$ as shorthand for `evimage n f S`. We also use the notation $\mathtt{Fun}_{\mathrm{n}}(\mathrm{R})$ as shorthand for the extensional function rings function_ring (carrier (R^{n})) R consisting of functions whose domain is a Cartesian power of the ring.

6.4.3 Polynomial Maps

It is useful to introduce definitions and lemmas about polynomial maps (i.e. tuples of polynomials). These will be important for reasoning about and constructing new examples of semi-algebraic functions. The definitions are straightforward,

```
definition poly_tuple_eval ::
"('a, nat) mvar_poly list ⇒ 'a list ⇒ 'a list" where
"poly_tuple_eval fs as =
    map (λ f. eval_at_poly R f as) fs ".
```

We also provide an extensional version of this function,

```
definition poly_map ::
"nat ⇒ ('a, nat) mvar_poly list ⇒ 'a list ⇒ 'a list"
   where
"poly_map n fs = (λ a ∈ carrier R^n. poly_tuple_eval fs a)".
```

This construction is useful because it can be shown that polynomial maps are simple examples of semi-algebraic maps, and hence preserve semi-algebraic sets under inverse images.

Chapter 7

Constructing $\mathbb{Z}_p$, $\mathbb{Q}_p$ and Hensel's Lemma

This section outlines the basic construction of the p-adic ring of integers $\mathbb{Z}_p$ and the p-adic field $\mathbb{Q}_p$, as well as proofs of Hensel's Lemma in both contexts. Work pertaining purely to the ring $\mathbb{Z}_p$ can be found in the AFP submission `padic_ints` [13], while work pertaining to $\mathbb{Q}_p$ is in a session titled `padic_fields`, which has been submitted for publication to the AFP as of writing.

7.1 Defining Zp

There are many options available for a formal construction of $\mathbb{Z}_p$ and $\mathbb{Q}_p$ in a proof assistant. For example, in [25], the author formalizes the proof of Hensel's Lemma in the Lean proof assistant. The approach here is to first define the p-adic absolute value on the field $\mathbb{Q}$, and then define the field $\mathbb{Q}_p$ as the completion of $\mathbb{Q}$ with respect to this absolute value. One can then define $\mathbb{Z}_p$ as the unit ball in $\mathbb{Q}_p$. We opt for a different approach, which is more compatible with the formalism of the HOL-Algebra library. The standard algebraic approach to defining $\mathbb{Z}_p$ as the inverse limit of the rings $\mathbb{Z}/p\mathbb{Z}$. Set theoretically, this is:

$$\begin{aligned}\mathbb{Z}_p &= \varprojlim_{i\in\mathbb{N}} \mathbb{Z}/p^i\mathbb{Z} \\ &= \left\{ f \in \prod_{i\in\mathbb{N}} \mathbb{Z}/p^i\mathbb{Z} \quad \middle| \quad \forall m < n.\ \ f(m) = f(n) \bmod p^m \right\}.\end{aligned}$$

This lends itself to a simple formalisation in Isabelle using existing tools from the standard distribution:

```
definition padic_set :: "int ⇒ padic_int set" where
    "padic_set p  =
        {f::nat ⇒ int .
               (∀ m::nat. f m ∈ carrier (residue_ring p^m))
            ∧  (∀ (n::nat) (m::nat).
                n > m ⟶ residue (p^m) (f n) = (f m))}".
```

Here the type `padic_int` is a synonym for the function type `nat ⇒ int`. Here the function `residue_ring` is pre-defined in the standard distribution so that `residue_ring n` is a ring-record object representing the ring $\mathbb{Z}/n\mathbb{Z}$. We can then define the standard ring operations for $\mathbb{Z}_p$ componentwise in the usual way. For example, addition can be defined in Isabelle as:

```
definition padic_add ::
        "int ⇒ padic_int ⇒ padic_int ⇒ padic_int"
        where "padic_add p f g n = (f n) ⊕_{residue_ring p^n} (g n)
   ".
```

Performing similar constructions to the above for the other usual ring constructions, we can then define the ring $\mathbb{Z}_p$ as a ring record object in the following way:

```
"padic_int p = (|carrier = padic_set p,
    mult = padic_mult p,
    one = padic_one p,
    zero = padic_zero p,
    add = padic_add p|)"}.
```

This must be accompanied by lemmas stating that for any prime p, the set `padic_set p` is closed under the operations `padic_add p` and `padic_mult p` and that these operations are associative, commutative, and obey the usual algebraic identities of ring operations.

We can bundle the data of our $\mathbb{Z}_p$ construction into a locale of its own, which we call `padic_int`. It fixes constants `Zp` for the ring $\mathbb{Z}_p$ as well as for the prime parameter p, and has as its only axiom that p is a prime.

For the sake of containing the scope of this project, these constructions have been performed specifically for the ring $\mathbb{Z}_p$. However, further improvements to this would allow one to define a general notion of inverse limit for sequences of rings, with $\mathbb{Z}_p$ only a special case of this. One could also provide a locale for discrete valuation rings (DVR), and then define the canonical completion of a DVR in terms of the inverse limit construction.

7.2 The Valuation on $\mathbb{Z}_p$

There are many options for how one can approach the problem of defining the p-adic valuation on `padic_set p`. One possible approach is to construct it as a real-valued absolute value, which maps to the pre-defined field of real numbers available in the standard HOL Library. Since our goal is the formalisation of results in p-adic model theory, this approach is not desirable, since it obscures the arithmetic of the value group, which plays an essential role in formalizing Macintyre's Theorem. Our approach instead occurs in two steps. First, we define an integer-valued valuation on $\mathbb{Z}_p \setminus 0$. We then define a new abstract type `eint` in Isabelle to represent the extended integers, with an infinite element added to represent the valuation of 0. We call the integer-valued valuation `ord_Zp` and the extended integer-valued valuation `val_Zp`. The main advantage of this two-step approach is that lemmas regarding `ord_Zp` can admit more proof automation, since they can avail themselves of Isabelle's built-in proof tactics such as linarith and presburger (described in [28]). One can then efficiently prove restricted versions of lemmas for `ord_Zp` applied to nonzero elements, and then reformulate them in general for `val_Zp` and transfer the proofs over, usually with a case distinction over the possibility of zero and nonzero elements.

7.2.1 The Integer-Valued Valuation on Nonzero Elements

The inverse limit definition of `padic_set p` means that each element of the set is literally the residue map of the element of the ring $\mathbb{Z}_p$ which it represents. This means that for every $x \in \mathbb{Z}_p$ is represented by an element `x` $\in$ `padic_set p`, such that for every natural number n, the value `x n` is just the element of $\mathbb{Z}/p^n\mathbb{Z}$ representing `x` mod p^n (technically `x n` is actually an integer between 0 and $p^n - 1$ which is a representative of the residue class). This allows for a straightforward definition of `ord_Zp x` for nonzero `x` $\in$ `padic_set p`. The definition below is for the function `padic_val`, which depends on the parameter p. Once we are working in the `padic_int` locale we can define `ord_Zp` to remove the dependence on p. The valuation of an element of $\mathbb{Z}_p$ is just the largest n for which its residue modulo p^n is zero. We arbitrarily assign the value -1 to 0 to make the function total on `padic_set p`.

```
definition padic_val :: "int ⇒ padic_int ⇒ int"  where
"padic_val p f ≡ if (f = padic_zero p) then -1
                 else int (LEAST k::nat. (f (Suc k)) ≠ 0".
```

Without too much difficulty we can then prove the basic properties of the valuation, such as multiplicativity:

```
lemma val_prod:
  assumes "prime p"
  assumes "f ∈ (padic_set p)"
  assumes "g ∈ (padic_set p)"
  assumes "f ≠ padic_zero p"
  assumes "g ≠ padic_zero p"
  shows
  "padic_val p (padic_mult p f g) = padic_val p f +
   padic_val p g".
```

From this it is easily inferred that $\mathbb{Z}_p$ is an integral domain:

```
lemma padic_int_is_domain:
   assumes "prime p"
   shows "domain (padic_int p)".
```

7.2.2 The Extended Integer-Valued Valuation

Our construction of the extended integers is directly modelled on a similar construction for the extended natural numbers from the standard Isabelle distribution [32]. The type `eint` is defined as an abstract type whose underlying set is just `int option`, i.e. a type whose elements either represent a unique integer, or a "None" type (which will be our infinity). We define a function

```
definition eint :: "int ⇒ eint" where
  "eint n = Abs_eint (Some n)"
```

which maps an integer to its associated extended integer. In this way we can easily define the `eint`-valued valuation:

```
definition(in padic_integers) val_Zp  where
"val_Zp x = (if (x = 0) then (∞::eint)
            else (eint (padic_val p x)))".
```

We then define addition and multiplication on the type `eint` in the obvious way on integer values, and stipulate (as is standard) that the sum or product of anything with ∞ is just ∞. We also define the ordering on `eint` in the usual way on integers, with ∞ as our infinite element.

7.3 Angular Components and Division

In the proof of Hensel's Lemma and elsewhere, it will be necessary to have a notion of restricted division of two p-adic integers. Since we have defined $\mathbb{Z}_p$ prior to defining

$\mathbb{Q}_p$, simply performing division of two elements of $\mathbb{Z}_p$ as their quotient in $\mathbb{Q}_p$ is not desirable. There is a pre-defined function for inverses of ring units in HOL-Algebra, but this can't be directly applied since we would like to be able to form the quotient x/y whenever $\mathrm{val}(x) \geq \mathrm{val}(y)$, even in the case that y may not be an actual unit. A convenient way to do this internally to $\mathbb{Z}_p$ is with angular components. We can define a function `ac_Zp` which maps a nonzero element $x \in \mathbb{Z}_p$ to its normalized value $p^{-\mathrm{val}(x)}x$. This will always have valuation 0, hence will always be a unit. The residues of this function implement the angular component maps which are also described in Part 1 of the thesis in Definition 2.3.3, and this construction is extended to the fields $\mathbb{Q}_p$, as is outlined in Section 7.5.3. We then define an ad-hoc division function on $\mathbb{Z}_p$ as:

```
definition divide where
"divide x y =
    (if x = 0 then 0 else
    (p[^](nat (ord_Zp x - ord_Zp y)) ⊗ ac_Zp x ⊗ (inv
  ac_Zp y)))".
```

Which corresponds to the mapping:

$$x, y \mapsto p^{val(x)-val(y)}\frac{p^{-\mathrm{val}(x)}x}{p^{-\mathrm{val}(y)}y}$$

when $\mathrm{val}(x) \geq \mathrm{val}(y)$, in which case it will simply evaluate to the p-adic integer x/y.

7.4 Hensel's Lemma

7.4.1 Cauchy Sequences and Completeness

The particular formalism we have chosen makes development of basic toplogical properties of $\mathbb{Z}_p$ relatively straightforward. The standard definition of a cauchy sequence for a valued field (as in section 2.4 of [20], for example) can be stated for $\mathbb{Z}_p$ with no difficulty. For the purposes of developing the necessary topology for proving Hensel's Lemma, we use the characterization of Cauchy sequences over $\mathbb{Z}_p$ which says that a sequences is Cauchy if and only if it's residues are eventually constant:

Lemma 7.4.1. *Suppose (s_n) is a sequence in $\mathbb{Z}_p$. Then (s_n) is Cauchy if and only if for all $k \in \mathbb{N}$, there exists some $N \in \mathbb{N}$ such that for all $n_0, n_1 > N$,*

$$s_{n_0} \mod p^k = s_{n_1} \mod p^k.$$

In our Isabelle formalism this can be expressed as a distinct rule for each direction of the bi-implication:

```
lemma is_Zp_cauchyI:
  assumes "s ∈ closed_seqs Zp"
  assumes "⋀ n.  (∃N. (∀ n0 n1. n0 > N ∧ n1 > N ⟹ (s n0
   ) n = (s n1) n))"
  shows "is_Zp_cauchy s"
```

```
lemma is_Zp_cauchy_imp_res_eventually_const:
  assumes "is_Zp_cauchy s"
  fixes n::nat
  obtains N r where "r ∈ carrier (Zp_res_ring n)" and "⋀ m
   . m > N ⇒ s m) n = r".
```

One useful aspect of this characterization in our formalism is that completeness of $\mathbb{Z}_p$ is almost immediate: an element of $\mathbb{Z}_p$ is literally a coherent residue map, and every Cauchy sequence induces such a map by looking at its eventually constant residue values for each residue degree n.

```
definition res_lim :: "padic_int_seq ⇒ padic_int" where
"res_lim s k = (THE r. (∃N.  (∀ m.
      m > N ⟹ (s m) k = r) ) ) ".
```

We can easily prove that the residue map defined in this way is an element of $\mathbb{Z}_p$, and that it is the limit of the Cauchy sequence:

```
lemma is_Zp_cauchy_imp_has_limit:
  assumes "is_Zp_cauchy s"
  assumes "a = res_lim s"
  shows "Zp_converges_to s a".
```

Since $\mathbb{Z}_p$ is a compact metric space, a function $f : \mathbb{Z}_p \to \mathbb{Z}_p$ is continuous if and only if it carries a Cauchy sequence (s_n) to a Cauchy sequence $(f(s_n))$. We directly adopt this as our definition of continuity, and prove that all polynomials are continuous (which is the only kind of continuous function we need to consider for the purposes of proving Hensel's Lemma). Some extra work is needed to actually infer that $f(\lim s_n) = \lim(f(s_n))$ holds for a continuous f and a Cauchy sequence s_n, but this is relatively straightforward:

```
lemma continuous_limit:
  assumes "is_Zp_continuous f"
  assumes "is_Zp_cauchy s"
  shows "Zp_converges_to (f ∘ s) (f (res_lim s))".
```

Although it is not needed for anything else in the project, we are also able to prove sequential compactness of $\mathbb{Z}_p$ in this formalism.

7.4.2 Proof of Hensel's Lemma

As in the formal proof in Lean by Robert Y. Lewis from [25], we used a proof of Hensel's Lemma using Newton's method exposited by Keith Conrad in [12] as an outline. The constants and hypothesis in the statement of the theorem are declared in a locale which we call `hensel` to reduce the need for repetition of hypotheses in lemmas:

```
locale hensel = padic_integers+
  fixes f::padic_int_poly
  fixes a::padic_int
  assumes f_closed: "f ∈ carrier Zp_x"
  assumes a_closed: "a ∈ carrier Zp"
  assumes fa_nonzero: "f·a ≠ 0"
  assumes hensel_hypothesis: "val_Zp (f·a) > 2*val_Zp ((
   pderiv f)·a)".
```

In the above we have defined abbreviated notation for application of a polynomial function. We can write `f·a` to denote application of a polynomial `f` to a number `a` in $\mathbb{Z}_p$. We have added the assumption that $f(a) \neq 0$ to remove unnecessary case distinctions from the proof, and this will be removed in the final statement of the theorem.

The proof of Hensel's lemma involves starting from an approximate root a, defining the Newton sequence:

$$a_0 = a$$

$$a_{n+1} = a_n - \frac{f(a_n)}{f'(a_n)}$$

and showing that this sequence is Cauchy and converges to a root of f of the desired form. We can easily define this sequence in Isabelle using our restricted p-adic division function. First we define the function which maps a_n to a_{n+1} in the Newton sequence, then define the Newton sequence itself as a recursive function.

```
definition newton_step :: "padic_int ⇒ padic_int" where
"newton_step x = x ⊖ (divide (f·x) (f'·x))"
```

```
fun newton_seq :: "padic_int_seq" ("ns") where
```

```
"newton_seq 0 = a"|
"newton_seq (Suc n) = newton_step (newton_seq n)".
```

In this definition we are using f' as an abbreviation for the polynomial derivative of f. We follow the same inductive pattern as Conrad in proving by induction on n that the Newton sequence increases in valuation as the index n increases:

```
lemma newton_seq_props_induct:
shows "⋀k. k ≤ n ⟹ ns k ∈ carrier Zp
    ∧ val_Zp (f'·(ns k)) = val_Zp ((f'·a))
    ∧ val_Zp (f·(ns k)) ≥ 2*(val_Zp (f'·a)) + (2^k)*t"
```

where t is a predefined valuative constant of positive valuation:

```
definition hensel_factor ("t") where
"hensel_factor = val_Zp (f·a) - 2*(val_Zp (f'·a))"

lemma t_pos:
"t > 0".
```

The main challenge in proving the inductive lemma above is performing the algebraic calculations in HOL-Algebra. Many applications of basic operations such as commutativity and associativity, as well as cancellation of quotients, must be applied manually after checking certain closure conditions.

We can finally prove Hensel's Lemma in two parts: first showing existence of the root, then showing uniqueness in a separate lemma:

```
lemma hensels_lemma:
  assumes "f ∈ carrier Zp_x"
  assumes "a ∈ carrier Zp"
  assumes "val_Zp (f·a) > 2*val_Zp ((pderiv f)·a)"
  shows "∃!α ∈ carrier Zp.
    f·α = 0 ∧ val_Zp (a ⊖ α) > val_Zp ((pderiv f)·a)"
```

7.5 Constructing $\mathbb{Q}_p$ and Importing Results

7.5.1 Fields of Fractions

Building on an existing AFP entry by Anthony Bordg [4] formalizing the general notion of the localization of a commutative ring at a multiplicative set , we defined the basic properties of the field of fractions over a domain in an Isabelle theory

`Fraction_Field` in [14]. We defined a locale `domain_frac` for proving lemmas about fields of fractions over a domain. Within this locale, we define the field of fractions `Frac R` for a domain `R`, and have an inclusion map ι for `R` into `Frac R`. We can then define choice functions `numer` and `denom` from `Frac R` to `R` satisfying

$$\texttt{x} = \iota\ \texttt{(numer x)}\ \otimes_{\texttt{Frac R}}\ \texttt{inv}_{\texttt{Frac R}}\iota\texttt{(denom x)}$$

for all $\texttt{x} \in$ `carrier (Frac R)` where $\texttt{inv}_{\texttt{Frac R}}$ is the multiplicative inverse function. With this in place the definition of $\mathbb{Q}_p$ is straightforward. In `Fraction_Field` we define explicit choice functions `numer` and `denom` which map elements of a fraction field `Frac R` over a domain `R` to witnesses in `R`. To work with a field $\mathbb{Q}_p$, we created a locale `padic_fields` which has named constants for the prime p, the ring $\mathbb{Z}_p$, the field $\mathbb{Q}_p$, and the inclusion map $\iota : \mathbb{Z}_p \to \mathbb{Q}_p$. Here we define $\mathbb{Z}_p$ exactly as in the previous section, and define $\mathbb{Q}_p$ simply by $\texttt{Q}_\texttt{p}$ = `Frac` $\texttt{Z}_p$. There is also a constant $\mathcal{O}_p$ for embedded image of $\texttt{Z}_p$ in $\texttt{Q}_\texttt{p}$ under ι. $\mathcal{O}_p$ is defined only as a set rather than a full ring structure, though closure properties are still proved. We also can define a partial inverse for ι

```
definition to_Zp where
"to_Zp a = (
 if (a ∈ 𝒪_p) then (SOME x. x ∈ carrier Z_p ∧ ι x = a)
             else 0_Z_p )".
```

The locale `padic_fields` can inherit the lemmas from `padic_integers` via the sublocale command:

```
sublocale padic_fields < padic_integers Z_p
apply (simp add: padic_integers_def prime)
using Z_p_def by auto
```

which allows us to efficiently import defintitions and facts from $\texttt{Z}_p$ to $\texttt{Q}_\texttt{p}$.

7.5.2 The p-adic Valuation

As in the case of the p-adic integers, we define the valuation on $\mathbb{Q}_p$ twice over: once with integer values for efficient reasoning, and once over the extended integers. Within the locale `padic_fields`, we name these `ord` and `val` respectively, and define `val` in terms of `ord`. We can define `ord` using the existing definition of `ord_Zp`:

```
definition ord where
"ord x = (ord_Zp (numer x)) - (ord_Zp (denom x))"

definition val where
```

```
"val x = (if x = 0 then (∞::eint) else eint (ord x))".
```

From this the basic properties of the valuation are easily proved from their analogues for $\mathbb{Z}_p$ and the properties of `numer` and `denom`.

7.5.3 Angular Components

We can define angular component functions on $\mathbb{Q}_p$ using their analogous definitions for $\mathbb{Z}_p$ as well as `numer` and `denom`:

```
definition angular_component where
"angular_component a = ac_Zp (numer a)⊗Zp(invZp ac_Zp (
   denom a))".
```

The angular component takes its values in `carrier` Z$_p$, which makes it straightforward to define versions which take values in the residue rings:

```
definition ac :: "nat ⇒ padic_number ⇒ int" where
"ac n x = (if x = 0 then (0::int)
                    else (angular_component x) n )"
```

and to prove that these maps are homomorphisms (since they are obtained by composition with the already defined residue maps on Z$_p$).

7.5.4 Hensel's Lemma for $\mathbb{Q}_p$

We can state and Hensel's Lemma for $\mathbb{Q}_p$ using the version proved for $\mathbb{Z}_p$, as well as our canonical maps between Z$_p$ and Q$_\text{p}$. A useful tool for this is development of the gauss norm for polynomials in $\mathbb{Q}_p[x]$. This construction is outlined for general valued fields in Section 2.2 of [20]. The gauss norm of a polynomial f is simply the minimum of the valuations of its coefficients, which can be easily stated in Isabelle:

```
definition gauss_norm where
"gauss_norm g = Min (val ‘ g ‘ {..degree g})"
```

where the function `Min` denotes the minimum element function for an ordered set, the notation "`f ‘ X`" for a function `f` defined on a set `X` denotes the image set, and `.n..n` denotes the set of natural numbers ranging from 0 up to and including `n`. We can prove that on this definition, a polynomial in $\mathbb{Q}_p[X]$ has positive gauss norm if and only if its coefficients all lie in $\mathcal{O}_p$:

```
lemma positive_gauss_norm_valuation_ring_coeffs:
  assumes "g ∈ carrier (UP Qp)"
  assumes "gauss_norm g ≥ 0"
  shows "g n ∈ Z_p"

lemma val_ring_cfs_imp_nonneg_gauss_norm:
  assumes "g ∈ carrier (UP Q_p)"
  assumes "⋀ n. g n ∈ Z_p"
  shows "gauss_norm g ≥ 0".
```

We then define maps between the polynomial rings $\mathbb{Q}_p[X]$ and $\mathbb{Z}_p[X]$ induced by the maps ι and `to_Zp` defined on their coefficients, and prove that these maps commute with evaluation of polynomials as functions and taking polynomial derivatives. Finally, we prove a version of Hensel's Lemma for $\mathbb{Q}_p$:

```
theorem hensels_lemma:
  assumes "f ∈ carrier (UP Q_p)"
  assumes "a ∈ 𝒪_p"
  assumes "gauss_norm f ≥ 0"
  assumes "val (f·a) > 2*val ((pderiv f)·a)"
  shows "∃!α ∈ 𝒪_p. f·α = 0 ∧ val (a ⊖ α) > val ((pderiv f)·a
   )".
```

As a sample application of this, we can prove the following common criterion for the existence of n^{th} roots in the field $\mathbb{Q}_p$:

```
lemma nth_root_poly_root:
  assumes "(n::nat) > 1"
  assumes "a ∈ 𝒪_p"
  assumes "val (a ⊖ 1) > 2*val ([n]·1)"
  shows "∃!b ∈ 𝒪_p. b[^]n = a ∧ val (b ⊖ 1) > val ([n]·1)".
```

In the above, the notation "`[n]·1`" refers to the inclusion of the natural number n in the field $\mathbb{Q}_p$.

Chapter 8

Sets and Functions on Powers of $\mathbb{Q}_p$

Our account of semi-algebraic sets closely follows Denef's in [19]. This requires bringing together content from section 6.4 and 7.5. In addition, since definition 5.2.1 requires appeal to boolean combinations, we need to include a formalisation of boolean algebras of sets generated by some basic set of generators. This will be especially important in the proof of Macintyre's theorem to keep track of repeated partitions of semi-algebraic sets into successively smaller ones.

8.1 Generated Boolean Algebras

We can define the boolean algebra generated by generator sets `B` over a universe set `S` as an inductive set in Isabelle. That is,

```
inductive_set gen_boolean_algebra
  for S and B  where
    universe: "S ∈ gen_boolean_algebra S B"
  | generator:  "A ∈ B ⟹ A ∩ S ∈ gen_boolean_algebra S B
   "
  | union:
    "[| A ∈ gen_boolean_algebra S B;
        C ∈ gen_boolean_algebra S B|]
        ⟹ A ∪ C ∈ gen_boolean_algebra S B"
  | complement: "A ∈ gen_boolean_algebra S B
        ⟹ S - A ∈ gen_boolean_algebra S B".
```

This defines the expression `gen_boolean_algebra S B` as the minimal collection of subsets of the set `S` which contains intersections of the generator sets in the collection `B` with `S`, contains `S` itself, and is closed under unions and complements. Technically this allows for the possibility that the generators B are not subsets of S, but in almost all use-cases they will be. Closure of gen_boolean_algebra S B under intersections and set differences are easy consequences of this defintion.

We will frequently be interested in looking at boolean algebras generated by a finite collection of generators. For example, this will be useful for partitioning sets according to the zero sets of some collection of polynomials, so that each polynomial is either always zero or always nonzero on each element of the partition. Given a finite set of generating sets Xs, we know that the atoms (i.e. the nonempty elements minimal with respect to inclusion) of the generated boolean algebra will always be given by an intersection of elements of Xs and their complements. We therefore give the definition,

```
definition subset_to_atom where
"subset_to_atom Xs As = ⋂ As - ⋃ (Xs - As)".
```

For a given subset As of the generators Xs, either subset_to_atom Xs As will be an atom of Xs, or it will be empty, which justifies the definition,

```
definition atoms_of where
"atoms_of Xs = (subset_to_atom Xs ` ((Pow Xs) - {{}})) -
   {{}}"
```

with Pow Xs denoting the power set of Xs. This definition makes it straightforward to establish that a finite collection of sets has only finitely many atoms. We can also easily prove that for every atom of Xs, and each element X∈Xs, either the atom is contained in X or disjoint from it,

```
lemma atoms_are_minimal:
  assumes "A ∈ atoms_of Xs"
  assumes "X ∈ Xs"
  shows "X ∩ A = {} ∨ A ⊆ X"
```

We can then recharacterize the atoms of a collection of generators as induced by points from the union of the generators. We name these types due to the conceptual similarity to types from model theory:

```
definition point_to_type where
"point_to_type Xs x = {X ∈ Xs. x ∈ X}"

lemma point_in_atom_of_type:
  assumes "x ∈ ⋃ Xs"
  shows "x ∈ subset_to_atom Xs (point_to_type Xs x)".
```

We can finally characterize finitely generated boolean algebras as the collection of all possible unions of the atoms of its generators,

```
lemma gen_boolean_algebra_generated_by_atoms:
  assumes "finite Xs"
  assumes "S = ⋃ Xs"
  shows "gen_boolean_algebra S Xs = ⋃ ` (Pow (atoms_of Xs)
   )"
```

which in turn means that finitely generated boolean algebras are themselves finite,

```
lemma fin_gens_imp_fin_algebra:
  assumes "finite Xs"
  assumes "S = ⋃ Xs"
  shows "finite (gen_boolean_algebra S Xs)".
```

Furthermore, we can prove that the atoms generated by a collection of generators `Xs` are the same as those generated by the whole boolean algebra itself,

```
lemma atoms_of_sets_eq_atoms_of_algebra:
  assumes "finite Xs"
  assumes "S = ⋃ Xs"
  shows "atoms_of Xs = atoms_of (gen_boolean_algebra S Xs)
   ".
```

In cell decomposition arguments, we often would like to take multiple partitions of some set and amalgamate these into one partition which is a refinement of all of them. This is can be easily expressed in the formalism we have developed above,

```
definition family_intersect where
"family_intersect parts = atoms_of (⋃ parts)"

lemma family_intersect_partitions:
  assumes "⋀ Ps. Ps ∈ parts ⟹ s partitions A"
  assumes "⋀Ps. Ps ∈ parts ⟹ finite Ps"
  assumes "finite parts"
  assumes "parts ≠ {}"
  shows "family_intersect parts partitions A".
```

8.2 Basic semi-algebraic Sets

The basic semi-algebraic generators defined in Definition 5.2.1 are easy to describe,

```
definition basic_semialg_set where
"basic_semialg_set (m::nat) (n::nat) P =
  {q ∈ carrier (Q_p^m). ∃y ∈ carrier Q_p. Qp_ev P q = (y[^]n)}"
```

using Qp_ev P q as shorthand for the evaluation of the polynomial P at the point q∈carrier$(\mathtt{Q}_\mathtt{p}^\mathtt{m})$. We can then define a predicate for basic semi-algebraic sets,

```
definition is_basic_semialg ::
"nat ⇒ ((nat ⇒ int) × (nat ⇒ int)) set list set ⇒ bool"
    where
"is_basic_semialg m S ≡
∃ (n::nat) ≠ 0.
    (∃ P ∈ carrier (Q_p^m). S = basic_semialg_set m n P)"
```

and define the class of semi-algebraic sets as the boolean alegbra generated by these

```
definition semialg_sets where
"semialg_sets m =
    gen_boolean_algebra (carrier (Q_p^m)) (basic_semialgs m)".
```

We define a predicate version is_semialgebraic m S which identifies a semi-algebraic subset S of carrier $\mathtt{Q}_\mathtt{p}^\mathtt{m}$, as well as an alternate version is_univ_semialgebraic S for semi-algebraic subsets S of carrier $(\mathtt{Q}_\mathtt{p})$ rather than the isomorphic carrier $\mathtt{Q}_\mathtt{p}^1$. From here we can begin to show that various familiar sets and constructions are in fact semi-algebraic. For example, we show that the valuation relation

$$\{(x, y) \in \mathbb{Q}_p^2 \mid \mathrm{val}(x) \leq \mathrm{val}(y)\}$$

is a basic semi-algebraic set. This requires proving the algebraic fact that $\mathrm{val}(y) \leq \mathrm{val}(x)$ holds for $x, y \in \mathbb{Q}_p$ (for any prime p) if and only if $y^4 + p^3x^4$ is a square, a proof which uses Hensel's lemma. We can then give the name Qp_val_poly to the polynomial $f(x, y) = y^4 + p^3x^4$, and prove the lemma:

```
lemma Qp_val_semialg:
  assumes "a ∈ carrier Q_p"
  assumes "b ∈ carrier Q_p"
  shows "val b ≤ val a ↔
        [a,b] ∈ basic_semialg_set 2 2 Qp_val_poly".
```

Giving the name val_relation_set to the subset of $\mathbb{Q}_p^2$ given by

$$\{(x, y) \mid \mathrm{val}(y) \leq \mathrm{val}(x)\}$$

, it is now trivial to infer that this is semi-algebraic:

```
lemma val_relation_is_semialgebraic:
"is_semialgebraic 2 val_relation_set".
```

In addition, it is important to state and prove the contents of Denef's Lemma 2.1 from [19], which addresses the various ways one can define new semi-algebraic subsets of `carrier` $(\mathtt{Q}_\mathtt{p}^\mathtt{m})$ from polynomials in $\mathtt{Q}_\mathtt{p}[\mathcal{X}_\mathtt{n}]$. One main tool we use for this is the lemma which shows that the inverse image of a semi-algebraic set under a polynomial map is again semi-algebraic,

```
lemma pullback_is_semialg:
  assumes "is_poly_tuple n fs"
  assumes "length fs = k"
  assumes "is_semialgebraic k S"
  shows "is_semialgebraic n (poly_map n fs)^{-1}_n S".
```

This lemma follows easily from the fact that the composition of a polynomial with a polynomial map produces a new polynomial.

8.3 Semi-Algebraic Functions

To define semi-algebraic functions as in definition 5.2.2, we create some new explicit functions to represent the modified version of inverse image being used,

```
definition partial_image where
"partial_image m f xs = (f (take m xs))#(drop m xs)"

definition partial_pullback where
"partial_pullback m f l S = (partial_image m f)^{-1}_{m+l} S"

definition is_semialg_function where
"is_semialg_function m f =
  f ∈ carrier (Q_p^m) → carrier (Q_p)  ∧
  (∀l ≥ 0. ∀S ∈ semialg_sets (1 + l).
    is_semialgebraic (m + l) (partial_pullback m f l S)
   )".
```

A priority is to verify the contents of Denef's Remark 1.5, namely to show that semi-algebraic functions have semi-algebraic graphs, and are closed under compostiton, addition, multiplication, and (multiplicative) inverses. This will then allow us to define the rings of semi-algebraic functions over $\mathtt{Q}_\mathtt{p}^\mathtt{n}$ for each `n`. In fact, once we have shown that semialgebraic functions are closed under composition, closure under the basic algebraic operations will be simple, due to the fact that the basic ring operations either are polynomial functions or can be defined in terms of them. We can therefore simply note that the sums, products, etc of functions are compositions of semi-algebraic functions with another semi-algebraic function.

8.4 Closure of Semi-Algebraic Functions Under Composition

The informal proof that semi-algebraic functions are closed under composition is given below.

Lemma 8.4.1. *Suppose $R \subseteq \mathbb{Q}_p^{n+n+k}$ is semi-algebraic. Then the set*

$$R' = \{(x, y) \in \mathbb{Q}_p^{n+k} \mid (x, x, y) \in R\}$$

is also semi-algebraic.

Proof. The map $(x, y) \to (x, x, y)$ is a polynomial map from $\mathbb{Q}_p^{n+k} \to \mathbb{Q}_p^{n+n+k}$. Then we see that R' is the inverse image of the set R under this mapping, hence it is also semi-algebraic. □

Lemma 8.4.2. *Suppose $f_i : \mathbb{Q}_p^m \to \mathbb{Q}_p$ are semi-algebraic, for $i = 1, ..., k$, and $S \subseteq \mathbb{Q}_p^{k+n}$ is semi-algebraic. Then the set*

$$\{(x, y) \in \mathbb{Q}_p^{m+n} \mid (f_1(x), \dots, f_k(x), y) \in S\}$$

is also semi-algebraic.

Proof. If $k = 1$ then this follows straightforwardly from the definition of semi-algebraic functions. Proceeding by induction on k, suppose we know this fact for k, and would like to prove it for $k + 1$. Take $S \subseteq \mathbb{Q}_p^{k+1+n}$ semi-algebraic. By induction, we know that the set,

$$S_0 := \{(x, t, y) \in \mathbb{Q}_p^{m+1+n} \mid (f_1(x), \dots, f_k(x), t, y) \in S\}$$

is semi-algebraic. By the definition of semi-algebraic functions (up to permutation of indices), it follows that the set

$$S_1 := \{(x, x', y) \in \mathbb{Q}_p^{m+m+n} \mid (x, f_{k+1}(x'), y) \in S_0\}$$

is semi-algebraic. Applying Lemma 8.4.1 we get that

$$S_2 := \{(x, y) \in \mathbb{Q}_p^{m+n} \mid (x, x, y) \in S_1\}$$

is semi-algebraic. The set S_2 is precisely the set

$$\{(x, y) \in \mathbb{Q}_p^{m+n} \mid (f_1(x), \dots, f_k(x), y) \in S\}$$

which completes the proof. □

Finally we can prove the desired result:

Proposition 8.4.3. Suppose $f_i : \mathbb{Q}_p^m \to \mathbb{Q}_p$ are semi-algebraic, for $i = 1, ..., k$, and $F : \mathbb{Q}_p^k \to \mathbb{Q}_p$ is semialgebraic. Then $F(f_1(x), \dots, f_k(x)) : \mathbb{Q}_p^m \to \mathbb{Q}_p$ is semi-algebraic.

Proof. It suffices to fix a semi-algebraic set $S \subseteq \mathbb{Q}_p^{1+n}$ for an arbitrary n, and show that the set $S_0 := \{(x, y) \in \mathbb{Q}_p^{m+n} \mid (F(f_1(x), \dots, f_k(x)), y) \in S\}$ is semi-algebraic. However, this set is of the form

$$\{(x, y) \in \mathbb{Q}_p^{m+n} \mid (f_1(x), \dots, f_k(x), y) \in S_1\}$$

where $S_1 := \{(x, y) \in \mathbb{Q}_p^{k+n} \mid (F(x), y) \in S\}$. S_1 is semi-algebraic by the definition of a semi-algebraic function, and therefore S_0 is by the previous lemma. □

After defining the notion of a semi-algebraic map to match the map

$$x \mapsto (f_1(x), \dots, f_k(x))$$

in the above lemma, this can be stated and proved in Isabelle:

```
lemma semialg_function_comp_closed:
  assumes "is_semialg_function m f"
  assumes "is_semialg_map k m g"
  shows "is_semialg_function k (f ∘ g)".
```

8.5 Inversion is Semi-Algebraic

Using the fact that semi-algebraic functions are closed under function composition, we only need to show that the function $x \to 1/x$ is semi-algebraic (arbitrarily sending $0 \to 0$ so as to be total). This requires showing that

$$S := \{(x, y) \in \mathbb{Q}_p^{1+k} \mid \exists t((x = 0 \land f(0, y) = t^n) \lor f(x^{-1}, y) = t^n)\}$$

are semi-algebraic for any $n > 0$ and polynomial f. The approach is straightforward. For any n, there is a polynomial $g(x, y)$ such that $x^{nm}f(x^{-1}, y) = g(x, y)$, where m is the degree of f in the variable x. Giving the map $f(x) \mapsto 1/f(x)$ the admitedly awkward name `one_over_fun n f` (for $x \in \mathbb{Q}_p^n$), we can prove the lemma in Isabelle:

```
definition one_over_fun where
"one_over_fun n f = inv_(SA n)(to_fun_unit n f)"

lemma one_over_fun_closed:
  assumes "f ∈ carrier (SA n)"
  shows "one_over_fun n f ∈ carrier (SA n)".
```

8.6 Rings of Semi-algebraic Functions

Denef's paper requires us to consider various arithmetic operations on semi-algberaic functions, as well as to consider polynomials with semi-algebraic coefficients. For this reason we chose to construct, for each n, the ring of semi-algebraic functions `f: carrier Q`$_{\mathtt{p}}^{\mathtt{n}}$ $\rightarrow$ `carrier Q`$_{\mathtt{p}}$, which we denote by `SA n`. We can then, for example, write `UP (SA n)` to refer to the ring of single-variable polynomials with semi-algebraic coefficients in n variables. These rings are defined as subrings of the rings of all possible functions `f: carrier Q`$_{\mathtt{p}}^{\mathtt{n}}$ $\rightarrow$ `carrier Q`$_{\mathtt{p}}$, which have already been constructed. Without doing this, we would be able to perform operations such as forming the sum of two functions, but this approach allows us to efficiently perform constructions such as taking the Taylor expansion of a semi-algebraic polynomial centred at a semi-algebraic function, and we will know that the resulting coefficients are again semi-algebraic. This material is outlined in the theory `padic_semialgebraic_function_ring`.

8.7 Piecewise Semi-Algebraic Functions

One crucial property of semi-algebraic functions is that they are closed under piecewise definitions, provided that the domain components are themselves semi-algebraic, and there are only finitely many pieces. This is easily shown from the definition of semi-algebraic functions. We provide a special function for (binary) piecewise definitions,

```
definition fun_glue where
"fun_glue n S f g = (λx ∈ carrier Q_p^n. if x ∈ S then f x
   else g x)"
```

and prove a closure lemma,

```
lemma fun_glue_closed:
  assumes "f ∈ carrier (SA n)"
  assumes "g ∈ carrier (SA n)"
  assumes "is_semialgebraic n S"
  shows "fun_glue n S f g ∈ carrier (SA n)".
```

We also define a parametric version of this function for piecewise definitions with finite but arbitrarily large collections of domain pieces. It requires a finite collection `Xs` of semi-algebraic sets (which are expected to be disjoint and cover all of `carrier` $\mathtt{Q}_{\mathtt{p}}^{\mathtt{n}}$) and a function `fs` mapping the sets in `Xs` to the semi-algebraic function we would like to take the values of on that piece. In Isabelle this can be defined using the definite description operator,

```
definition parametric_fun_glue where
"parametric_fun_glue n Xs fs =
(λ x ∈ carrier (Q_p^n). let S = (THE S. S ∈ Xs ∧ x ∈ S) in (
   fs S x))".
```

This characterization of this function makes the output of the function transparent and simple to prove. However, to show that this function is semi-algebraic, we can prove that the result of gluing functions $f_0, \ldots, f_n$ along semi-algebraic sets $S_0, \ldots, S_n$ can be written as the finite sum $f_0\chi_{S_0}+\cdots+f_n\chi_{S_n}$ where χ_{S_i} denotes the characteristic function on S_i. That χ_S is semi-algebraic follows from the fact that it can be expressed as `fun_glue n S` $1_{\mathtt{(SA\ n)}}$ $0_{\mathtt{(SA\ n)}}$.

8.8 Semi-Algebraic Units and Division

Since we have proved that multiplicative inversion is semi-algebraic, it is simple to show that $1/f$ is semi-algebraic whenever f is. This statement, however, is somewhat ambiguous when f is a function which may sometimes take the value zero. In practice (and in Denef's paper), the meaning of $1/f$ for such functions is often left ambiguous, with the understanding that this function will not be applied at points x where $f(x) = 0$, and could be arbitrarily redefined at such points to yield a function with a multiplicative inverse everywhere. Isabelle requires us to make a concrete decision on this point, so we chose to define a function `to_fun_unit n f` which uses `fun_glue` to glue `f` to the function $1_{\mathtt{SA\ n}}$ along the set of nonzero values of `f` (which is semi-algebraic as long as f is. This leaves us with the awkward but workable solution of using the value `inv`$_{\mathtt{(SA\ n)}}$ `(to_fun_unit n f)` as a stand-in for $1/f$.

8.9 Cells and Cell Decompositions

8.9.1 Defining Cells

Denef's notion of a cell differs slightly from the one we give in definition 2.2.2 in that it does not include a condition for membership in a certain multiplicative coset of $\mathbb{Q}_p$.

Definition 8.9.1. A cell in $\mathbb{Q}_p^m \times \mathbb{Q}_p$ is a set of the form

$$A = \{(x,t) \in \mathbb{Q}_p^m \times \mathbb{Q}_p \mid x \in C \text{ and } \mathrm{ord}(a_1(x)) \square_1 \mathrm{ord}(t - c(x)) \square_2 \mathrm{ord}(a_2(x))\},$$

where C is a semi-algebraic subset of $\mathbb{Q}_p^m$, and $a_1(x)$, $a_2(x)$, $c(x)$ are semi-algebraic functions from $\mathbb{Q}_p^m \to \mathbb{Q}_p$, and $\square_i$ is either $\leq$, $<$, or no condition. We call $c(x)$ a center of the cell A.

There are a few choices that need to be made in deciding how to translate this definition in Isabelle. First, one needs to decide how to specify the boundary constraints $\square_i$. This is done with the notion of a convex condition over the value group.

To begin, we define a predicate which can identity a subset of the value group which is convex in the obvious way,

```
definition is_convex :: "eint set ⇒ bool" where
"is_convex A = (∀ x ∈ A. ∀y ∈ A. ∀c. x ≤ c ∧ c ≤ y → c ∈
   A)".
```

We can then define four special classes of convex sets which will be sufficient to characterize all possible convex subsets of the extended integers,

```
definition closed_interval :: "eint ⇒ eint ⇒ eint set"
   where
  "closed_interval α β = {a. α ≤ a ∧ a ≤ β}"

definition left_closed_interval :: "eint ⇒ eint ⇒ eint
   set" where
"left_closed_interval α β = {a. α ≤ a ∧ a < β}"

definition closed_ray :: "eint ⇒ eint ⇒ eint set" where
"closed_ray α β  = {a. a ≤ β}"

definition open_ray :: "eint ⇒ eint ⇒ eint set" where
"open_ray α β  = {a. a < β}".
```

From these definitions, we can then disjunctively define a convex condition as a set defined by one of the above four binary functions,

```
definition is_convex_condition :: "(eint ⇒ eint ⇒ eint
   set) ⇒ bool"
  where "is_convex_condition I ≡
              I = closed_interval ∨ I =
   left_closed_interval ∨
              I = closed_ray ∨ I = open_ray".
```

With this definition in hand, we can replace the choice of two boundary conditions $\square_i$ in the definition of cell with a choice of one of the four possible convex condition.

```
definition cell :: "nat ⇒ padic_tuple set ⇒
   padic_nary_function ⇒ padic_nary_function ⇒
   padic_nary_function ⇒ (eint ⇒ eint ⇒ eint set)" where
  "cell m C c a1 a2 I  =
        {as ∈ carrier (Q_p^{Suc m}). tl as ∈ C ∧
             val (hd as ⊖ (c (tl as)))
                 ∈ I (val (a1 (tl as))) (val (a2 (tl as)))
   }"
```

where `tl` and `hd` are the tail and head functions on linked lists. Using these functions for reasoning about cells is very useful because of the privileged status of the "t" coordinate in an element (x, t) of a cell. For this reason the coordinates of our cells are ordered oppositely to Denef's. The definition of cell as a set will however be inadequate for our purposes. This is because one set can be defined by various different choices of defining parameters, and thus the parameters C, c, a_1, a_2, I cannot be recovered from the data of the set `cell m C c a1 a2 I` alone. For this reason we define a special datatype called a *cell condition*, which abstractly carries the defining data of a cell,

```
datatype cell_condition = Cond nat
                               "padic_tuple set"
                               "padic_nary_function"
                               "padic_nary_function"
                               "padic_nary_function"
                               "eint ⇒ eint ⇒ eint set".
```

Given a cell condition `Cond m C c a1 a2 I`, we have named functions `arity`, `fibre_set`, `center`, `l_bound`, `u_bound`, `boundary_condition` which explicitly return the parameters `m, C, c, a1, a2, I` respectively. We can also define a function `condition_to_set` which maps an abstract cell condition to the underlying set which it corresponds to. We could provide well-typed parameters to the `Cond` constructor which would produce “junk” cells which we do not want to consider. For example, the parameter `C` might fail to be a semi-algebraic set, or the boundary condition `I` could fail to map to a convex set. Since it will be possible to construct malformed cell conditions which are well-typed but do not define legitimate cells, we also provide a predicate on cell conditions Skolemwhich pick out the desired ones,

```
primrec is_cell_condition :: "cell_condition ⇒ bool"
   where
"is_cell_condition (Cond n C c a1 a2 I) =
is_semialgebraic n C ∧ c ∈ carrier (SA n) ∧ a1 ∈ carrier (
   SA n) ∧ a2 ∈ carrier (SA n) ∧ is_convex_condition I".
```

8.9.2 Cells are Semi-Algebraic

Since we have proved that the valuation relation is semi-algebraic, we can show that a cell defined by a cell condition is semi-algebraic. In this section we outline how this proof is formalized to give an idea of how such arguments proceed in our formalism. The argument is very elementary, but like many of the results we have formalized, the details can become quite tedious. The final result is stated in the lemma,

```
lemma condition_to_set_is_semialg:
  assumes "is_cell_condition 𝒞"
  assumes "arity 𝒞 = m"
  shows "is_semialgebraic (Suc m) (condition_to_set 𝒞)".
```

First, we show, for any $n > 0$, that a set of the form

$$\{x \in \mathbb{Q}_p^n \mid \mathrm{ord}(f(x))\square\mathrm{ord}(g(x))\}$$

is semi-algebraic if f and g are both semi-algebraic (where $\square$ may be $\leq$, $<$,or no condition). This is easy to show since the valuation relation is semi-algebraic, and this set is therefore inverse image of a semi-algebraic set under the semi-algebraic map $(f(x), g(x)) : \mathbb{Q}_p^n \to \mathbb{Q}_p^2$. Since semi-algebraic sets are closed under intersection, we get that $\{x \in \mathbb{Q}_p^n \mid \mathrm{ord}(a_1(x))\square\mathrm{ord}(f(x))\square\mathrm{ord}(a_2(x))\}$ is also semi-algebraic provided that a_1, f, and a_2 are.

Next, we note that if $f(x)$ is a semi-algebraic function $\mathbb{Q}_p^m \to \mathbb{Q}_p$, then the function $(t, x) \mapsto f(x)$ is a semi-algebraic function $\mathbb{Q}_p^{m+1} \to \mathbb{Q}_p$, since this is the composition of f with the semi-algebraic map $(t, x) \mapsto x$. We define a generic operation for functions of this kind which we call `drop_apply`,

```
definition drop_apply where
"drop_apply m n f = restrict (f ∘ drop n) (carrier (Q_p^m))"
```

where the function `drop n` is the function which maps a list to a new list obtained by removing the first `n` elements. We can prove that for any $k \geq n$, the function `drop n` is a semi-algebraic map from $\mathbb{Q}_p^k \to \mathbb{Q}_p^{k-n}$. This requires showing that it is induced by evaluating the polynomial map $(x_1, \ldots, x_k) \mapsto (x_{n+1}, x_{n+2} \ldots, x_k)$. It will then follows that `drop_apply k n f` will always be a semi-algebraic function from $\mathbb{Q}_p^k \to \mathbb{Q}_p$ as long as $f : \mathbb{Q}_p^n \to \mathbb{Q}_p$ is, and $k \geq n$. We can express the map $\mathbb{Q}_p^{1+m} \to \mathbb{Q}_p$ defined by $(t, x) \mapsto f(x)$ as `drop_apply (Suc m) m f`. If $c(x)$ is semi-algebraic $\mathbb{Q}_p^m \to \mathbb{Q}_p$, then the function $(t, x) \mapsto t - c(x)$ also semi-algebraic since this is the composition of the polynomial $(x_1, x_2) \mapsto x_1 + x_2$ with the semi-algebraic map $(t, x) \to (t, c(x))$. From these observations, and the previous paragraph (applied to $n = m + 1$), that for any m, the set

$$\{(t, x) \in \mathbb{Q}_p \times \mathbb{Q}_p^m \mid \mathrm{ord}(a_1(x)) \Box_1 \mathrm{ord}(t - c(x)) \Box_2 \mathrm{ord}(a_2(x))\}$$

is semi-algebraic.

Finally, if $C \subseteq \mathbb{Q}_p^m$ is semi-algebraic, we know that the cartesian product $\mathbb{Q}_p \times C \subseteq \mathbb{Q}_p^{m+1}$ is also semi-algebraic. Intersecting this with the set from the last paragraph, we get that

$$\{(t, x) \in \mathbb{Q}_p \times \mathbb{Q}_p^m \mid x \in C \wedge \mathrm{ord}(a_1(x)) \Box_1 \mathrm{ord}(t - c(x)) \Box_2 \mathrm{ord}(a_2(x))\}$$

is semi-algebraic, as desired.

8.9.3 Cell Decompositions

Having defined cells, we can now define cell decompositions. For reasons discussed previously, we would like a cell decomposition to be a collection of cell conditions, rather than a collection of sets, because this allows us to preserve parameter information that would otherwise be lost. We define a boolean valued function `is_cell_decomp n S`, where `n` is an arity, `S` is a set to be decomposed, which identitfies when a set of cells `A` is a valid cell decomposition:

```
definition is_cell_decomp :: "arity ⇒ cell_condition set
   ⇒ padic_tuple set ⇒ bool" where
"is_cell_decomp n S A ≡ finite S ∧
        (∀s ∈ S. is_cell_condition s ∧ arity s = n) ∧
        ((condition_to_set ‘ S) partitions A) ∧
        A ⊆ carrier (Q_p^(Suc n)) ∧
        (∀ s ∈ S. ∀s’ ∈ S. s ≠ s’ →
            condition_to_set s ∩ condition_to_set s’ = {})
   ".
```

This just says that a cell decomposition of a set $A \subseteq \mathbb{Q}_p^{n+1}$ is a finite collection of cell conditions whose underlying sets partition A. The last condition in the conjunction, that distinct cells `s, s'` will produce disjoint sets, may look redundant but is in fact needed. This is because two distinct cells could produce identical underlying sets, which would mean that they would contribute the same member to the collection of underlying sets.

A basic technique for producing new cell decompositions of a set $A \subseteq \mathbb{Q}_p^{1+m}$ is to take an existing partition of A and, for each set in this partition, producing a cell decomposition of this set where each new cell satisfies a certain property. This is so ubiquitous in the proof of Macintyre's Theorem that we include it as a pair of lemmas. The first lemma below assumes the initial partition of A is induced by a cell decomposition, and the second only assumes it is a partition by sets:

```
lemma refine_each_cell:
  assumes "is_cell_decomp m S A"
  assumes "⋀ C. C ∈ S ⇒
        ∃S'. is_cell_decomp m S' (condition_to_set C) ∧
            (∀B ∈ S'. P B)"
  shows "∃S'. is_cell_decomp m S' A ∧ (∀B ∈ S'. P B)"

lemma refine_each_cell':
  assumes "A ⊆ carrier (Q_p^{Suc m})"
  assumes "As partitions A"
  assumes "finite As"
  assumes "⋀C. C ∈ As ⟹
        ∃S. is_cell_decomp m S C ∧
        (∀B ∈ S. P B)"
  shows "∃S'. is_cell_decomp m S' A ∧ (∀B ∈ S'. P B)".
```

Most of our formalized proofs from Denef's paper which construct cell decompositions by repeated refinement of cell decompositions are structured by repeatedly using `refine_each_cell` on successively smaller cells until the desired property holds.

8.10 Algebraic Properties of Cells

8.10.1 Cells With a Common Center Generate a Boolean Algebra

A common theme in this project was that many of Denef's innocuous remarks which are made without proof turned out to be some of the more tedious assertions to verify in Isabelle. One of these comes at the beginning of Denef's proof of cell decomposition II: "We will often use without mentioning the trivial fact that a boolean combination of cells with the same center $c(x)$ can be partitioned into a finite number of cells with the same center $c(x)$".

The first step in verifying this was to exhaustively classify the results of taking set differences and intersections between the four different types of convex conditions, and showing that the results is always a disjoint union of new convex conditions. This requires explicitly identifying the endpoints of the new convex sets, in terms of the endpoints of the old ones. For example,

```
lemma open_ray_minus_closed_interval:
  assumes "d ≠ ∞"
  assumes "I = open_ray a b"
  assumes "J = closed_interval c d"
  shows "I - J =
    open_ray a (min b c) ∪ left_closed_interval (d + 1) b
   ".
```

We then need to verify that when the endpoints of such sets are given by the valuations of semi-algebraic functions, the endpoints of the new sets also are. For example, in the above example, we would need to verify that if a, b, c, d are all semi-algebraic, then there are new semi-algebraic functions α, β satisfying $val(\alpha(x)) = min((val(b(x))), val(c(x))))$ for all x and $val(\beta(x)) = val(d(x)) + 1$ for all x. These constructions can be easily performed using piecewise functions and scalar multiplication, but the need to exhaustively cover every case requires many lines of proof to produce.

To complete the proof we define a predicate `is_c_decomposable m c C` which holds if and only if a set $C \subseteq \mathbb{Q}_p^{1+m}$ can be decomposed into a finite disjoint union of cells with center c. We then define a function `c_cells m c C` which denotes the class of cells with center `c` which are subsets of the given set `C`. We also require a function `c_decomposables m c C` which picks out those subsets of the given set `C` which can be partitioned into cells with center `c`. The formalisation of Denef's remark can be encapsulated in the statement that the class `c_decomposables m c C` is the same as the boolean algebra generated over `C` by the generators `c_cells m c C`. This is the content of the following lemma,

```
theorem c_decomposable_is_gen_boolean_algebra:
  assumes "is_c_decomposable m c C"
  shows "c_decomposables m c C =
            gen_boolean_algebra C (c_cells m c C)".
```

Given that these form a boolean algebra, we can prove lemmas such as the following, which are necessary for proving the cell decomposition theorems. It says that if a subset of a cell with center c has a decomposition into cells of center c, then its complement within the larger cell also admits such a decomposition:

```
      lemma cell_decomp_same_center:
  assumes "is_cell_condition 𝒞
  assumes "𝒞= Cond m C c a1 a2 I"
  assumes "B ⊆ condition_to_set 𝒞
  assumes "∃ S. is_cell_decomp m S B ∧ (∀ A ∈ S. center A
   = c)"
  shows "∃ S'. is_cell_decomp m S' (condition_to_set 𝒞 - B
   ) ∧
  (∀A ∈ S'. center A = c)".
```

8.10.2 Algebras of Cells with One Boundary Point

It is also useful to consider a subclass of the class `c_decomposables m c C` which also enjoys convenient closure properties under boolean operations. This is the class of sets which admit decompositions by cells of the form `Cond m C c a1 a1 closed_interval`, i.e. cells like,

$$\{(t, x) \mid x \in C \text{ and } \mathrm{ord}(t - c(x)) = \mathrm{ord}(a_1(x))\}$$

These sets are also closed under operations such as set differences, unions, and intersections. However, verifying this poses a challenge because they do not form a generated boolean algebra. This is because there is no sensible choice of "universe" set to situate these set in such they are closed under complements relative to the universe. For this reason, we need to posit a slightly different kind of set algebra to classify these, which we call a `cell_algebra`:

```
inductive_set cell_algebra
  for Cells  where
    empty: "{} ∈ cell_algebra Cells"
  | generator:  "A ∈ Cells  ⟹ A ∈ cell_algebra Cells"
  | disjoint_union:      "[|A ∈ cell_algebra Cells ; C ∈
   cell_algebra Cells; A ∩ C = {}|]  ⟹ A ∪ C ∈
   cell_algebra Cells".
```

This says that a cell algebra generated over some collection of generator `Cells` is the smallest class of sets which contains the empty set, all generators, and is closed under disjoint union. In the case that the cells algebra is closed under intersections and set differences of generating cells, we can show that the resulting cell algebra will be closed under finite (possibly non-disjoint) unions of generators:

```
lemma cell_algebra_finite_union:
  assumes "⋀ A C. [|A ∈ Cells ; C ∈ Cells|] ⟹
        A ∩ C ∈ cell_algebra Cells"
  assumes "⋀ A C. [|A ∈ Cells ; C ∈ Cells|] ⟹
        A - C ∈ cell_algebra Cells"
  shows "⋀S. finite S ∧ S ⊆ Cells ⟹ ⋃ S ∈ cell_algebra
   Cells".
```

We give the name `one_val_point_c_cells m c C` for the class of such cells whose underlying set is contained in the set `C`. We then give the name `one_val_point_c_decomposables m c C` for the class of subsets of `C` which admit decompositions by cells in `one_val_point_c_cells m c C`. It is clear that `one_val_point_c_decomposables m c C` is the cell algebra generated by `one_val_point_c_cells m c C`. One therefore only needs to show the above lemma for the set `one_val_point_c_cells m c C` to conclude that such sets are closed under finite unions. This way, if a certain set is covered by these cells, we can automatically conclude that it admits a cell decomposition by them.

8.10.3 Endpoint Algebras

There is another class of cell algebra we would like to consider, which we refer to as "Endpoint Algebras". Given a finite set `Fs` of semialgebraic functions, and a cell condition $\mathcal{C}$ `= Cond m C c a1 a2 I`, we would like to decompose the underlying set of $\mathcal{C}$ into finitely many cells `S` which are compatible with `Fs` in the sense that for any cell $\mathcal{D} \in$ `S`, and any cell $\mathcal{B}$ `= Cond m C c f g J` where `f, g ∈ Fs` and `J` is arbitrary, either `condition_to_set` $\mathcal{D} \subseteq$ `condition_to_set` $\mathcal{B}$ or they are disjoint. This will be useful in the proof of Denef's Theorem II. In particular, it will be crucial for establishing equation (3) from Denef's proof of cell decompostion theorem II_d. This proof will use the results from the previous section that sets which can be decomposed into cells with a common center form a boolean algebra. We can proceed as follows:

1. Form the collection of underlying sets of cells $\mathcal{B}$ `= Cond m C c f g J` where $f, g \in Fs \cup \{a1, a2\}$, which are contained in `condition_to_set` $\mathcal{C}$. These generate a subalgebra of c-decomposable sets contained in `condition_to_set` $\mathcal{C}$. This subalgebra is what we call

   `endpoint_c_algebra` in the theory `Algebras_of_Cells`.

2. Since there are finitely many sets above, we can obtain the atoms of the subalgebra they generate. Since c-decomposable sets form a boolean algebra, these atoms themselves will be c-decomposable. Thus each atom can be further decomposed into a finite disjoint union of cells centered at c. Collecting the cells in each of these decompositions will give us our desired decomposition of $\mathcal{C}$. We can thus prove the following lemma:

```
lemma semialg_boundary_cell_decomp:
  assumes "finite Fs"
  assumes "Fs ⊆ carrier (SA m)"
  assumes "is_cell_condition 𝒞"
  assumes "𝒞 = Cond m B c a1 a2 I"
  shows
  "∃S. is_cell_decomp m S (condition_to_set (Cond m B c a1
    a2 I))
  ∧ (∀C ∈ S. center C = c
  ∧ (∀f ∈ Fs. ∀g ∈ Fs. ∀I. is_convex_condition I ⟶
    condition_to_set C ⊆ condition_to_set (Cond m B c f g
   I)
    ∨ condition_to_set C ∩ condition_to_set (Cond m B c f
   g I) = {}))"
```

which says that given any finite set of endpoints, there is a finite collection of cells which are either totally contained in, or totally disjoint from, any other cell defined with the same fibres and center, and with boundary endpoints draws from the set Fs.

Chapter 9

Proving Denef's Cell Decomposition Theorems

Denef's proof is by a joint induction on degrees of the polynomials mentioned in theorems 5.2.2 and 5.2.3. The names I_d and II_d are given to the statements in theorems 5.2.2 and 5.2.3, restricted to polynomials of degree $\leq d$. The inductive argument proceeds by proving:

1. I_0
2. II_0
3. I_d and II_d implies I_{d+1}
4. I_{d+1} implies II_{d+1}

There are also two lemmas on the construction of semi-algebraic functions which fit into this inductive argument,

Lemma 9.0.1 (Lemma 2.3 from [19]). *Assume that cell decomposition theorem II_d holds. Let t be one variable and $x = (x_1, \ldots, x_m)$. Let $g(x, t)$ be a polynomial in t of degree $\leq d + 1$ with coefficients which are semialgebraic functions of x taking values in $\mathcal{O}_p$. Let $e \in \mathbb{N}$, $\kappa \in \mathcal{O}_p$ be fixed. Suppose that $\xi(x)$ is a function from $\mathbb{Q}_p^m$ to $\mathcal{O}_p$ such that for all $x \in \mathbb{Q}_p^m$:*

1. $g(x, \xi(x)) = 0$
2. $\xi(x) \equiv \kappa \ mod\ p^{e+1}$
3. $ord\ g'(x, \xi(x)) \leq e$,

where g' denotes the derivative of g with respect to t. Then $\xi(x)$ is a semialgebraic function.

Lemma 9.0.2 (Lemma 2.4 from [19]). *Assume cell decomposition II_d. Let $\xi(x)$ be a semialgebraic function from $\mathbb{Q}_p^m \to \mathbb{Q}_p$, $x = (x_1, \ldots, x_m)$. Let $k \in \mathbb{N}$, $k \geq 2$, $k \leq d+1$. Suppose for every $x \in \mathbb{Q}_p^m$ that $\xi(x) \neq 0$, and that ord $(\xi(x))$ is a multiple of k. Then there exists a semialgebraic function $\eta(x)$ from $\mathbb{Q}_p^m \to \mathbb{Q}_p$ such that*

$$\mathit{ord}(\eta(x)) = \frac{1}{k}\mathit{ord}(\theta(x)), \text{ for all } x \in \mathbb{Q}_p^m$$

The proof of I_d and II_d is structured with a series of locales to keep track of the complex parameters and assumptions involved. These are laid out in the theory `Locales_For_Macintyre`.

9.1 The Conclusions of the Decomposition Theorems

The conclusion of Theorem 5.2.2 is a statement about the boundedness of the expression,

$$\mathrm{ord} f(x,t) - Min_i \mathrm{ord}[a_i(x)(t - c(x))^i]$$

on some set A. In a footnote, Denef clarifies this to mean that there exists some $e \in \mathbb{N}$ such that

$$\mathrm{ord} f(x,t) \leq Min_i \mathrm{ord}[a_i(x)(t - c(x))^i] + e$$

for all values $(t, x) \in A$. We make this relationship precise in the locale which we name `SA_poly_ubounded`:

```
locale SA_poly_ubounded = padic_fields +
  fixes n P c A N
  assumes P_closed: "P ∈ carrier (UP (SA n))"
  assumes c_closed: "c ∈ carrier (SA n)"
  assumes A_closed: "A ⊆ carrier (Q_p^{Suc n})"
  assumes ubounded: "⋀ x t. t#x ∈ A ⟹
        val ((SA_poly_to_Qp_poly n x P)· t) ≤
        val ((UPQ.taylor_term (c x)
                (SA_poly_to_Qp_poly n x P) i)·t) + eint N".
```

We can then express the conclusion of the theorem for a particular semi-algebraic polynomial f by the expression "`SA_poly_ubounded p n f c A e`" for the particular integer bound `e` which we have found.

The conclusion of theorem 5.2.3 asserts that we can perform a certain factorization of polynomials,

$$f(x,t) = u(x,t)^n h(x)(t - c(x))^\nu$$

which again can be encoded into a locale:

```
locale SA_poly_factors  = padic_fields +
  fixes n::nat and  m::nat and  P::
   padic_nary_function_poly and
        c:: padic_nary_function and  A::"padic_tuple set"
   and
        u:: padic_nary_function and  h::
   padic_nary_function  and  k::nat
  assumes h_closed: "h ∈ carrier (SA n)"
  assumes c_closed: "c ∈ carrier (SA n)"
  assumes u_closed: "u ∈ (carrier (Q_p^{Suc n}) → carrier Q_p)"
  assumes A_closed: "A ⊆ carrier (Q_p^{Suc n})"
  assumes u_val:
    "⋀ x t. [| x ∈ carrier (Q_p^n); t ∈ carrier Q_p; t#x ∈ A|]
    ⟹
    val (u (t#x)) = 0"
  assumes factors:
    "⋀ x t. [| x ∈ carrier (Q_p^n); t ∈ carrier Q_p; t#x ∈ A|]
    ⟹
    (SA_poly_to_Qp_poly n x P)· t = ((u (t#x))[^] m)⊗(h x)
   ⊗ (t ⊖ (c x))[^] k".
```

9.2 Expressing Theorems I and II With Locales

We can now use the locales of the conclusion to express the propositions for our inductive argument in locales as well:

```
locale denef_I = padic_fields +
  fixes d::nat
  assumes cell_decomp:
    "⋀m P.[| P ∈ carrier (UP (SA m)); deg (SA m) P ≤ d |]
    ⟹
    ∃ S. (is_cell_decomp m S (carrier (Q_p^{Suc m})) ∧
    (∀A ∈ S. ∃N. SA_poly_ubounded p m P (center A) (
   condition_to_set A) N))"
```

```
locale denef_II = padic_fields +
  fixes d::nat
  assumes cell_decomp:
  "⋀n m Ps.
    [| finite Ps ;
    (∀P ∈ Ps. P ∈ carrier (UP (SA n)) ∧ deg (SA n) P ≤ d);
    m > 0 |]
    ⟹
        (∃ S. (is_cell_decomp n S (carrier (Q_p^{Suc n})) ∧
        (∀A ∈ S. ∀P ∈ Ps. ∃u h k. SA_poly_factors p n m P
            (center A) (condition_to_set A) u h k)))".
```

9.3 Constructing Semialgebraic Functions

There are two important lemmas which Denef uses to infer Macintyre's Theorem, which pertain to the construction of semi-algebraic functions from functions which are roots of polynomials. These are Lemmas 2.3 and 2.4 of [19]. Both of these Lemmas fit into the inductive scheme for proving cell decomposition theorems I and II, and hence they assume theorem II for polynomials of degree $\leq d$. Both Lemmas 2.3 and 2.4 use a substantially similar procedure for proving that function is semi-algebraic, based on euclidean division of semi-algebraic polynomials. In Lemma 2.3, we have a function which is a root of the semi-algebraic polynomial $g(x,t)$, and in Lemma 2.4, the proof proceeds by constructing a semi-algebraic function which is a k^{th} root of another semi-algebraic function. In either case, we have some function $\xi(x)$ and a semi-algebraic polynomial $g(x,t)$ such that we know $g(x,\xi(x)) = 0$ always holds.

To show that $\xi(x)$ is semi-algebraic, we can fix a semi-algebraic set S and show that the set,

$$\{(y,x) \in \mathbb{Q}_p^{r+m} \mid (y,\xi(x)) \in S\}$$

is semi-algebraic. In fact, it is fine to assume that S is a basic semi-algebraic set, of the form

$$S = \{(y,t) \in \mathbb{Q}_p^{r+1} \mid \exists z \in \mathbb{Q}_p . f(y,t) = z^n\}$$

for some semi-algebraic polynomial $f(y,t)$ and some $n > 0$. The goal is therefore to show that the set

$$\{(y,x) \in \mathbb{Q}_p^{r+m} \mid \exists z \in \mathbb{Q}_p . f(y,\xi(x)) = z^n\}$$

is semi-algebraic.

We can view $f(y,t)$ and $g(y,t)$ as polynomials in one variable t, whose coefficients are semialgebraic functions in the variables (x,y). In this way, we perform polynomial division of f by g to get an equation:

$$f(y,t) = q(x,y,t)g(x,t) + f_1(x,y,t)$$

with the degree of f_1 in the variable t being $\leq d-1$. so that we know that for all $(x,y) \in \mathbb{Q}_p^{r+m}$, $f(y,\xi(x)) = f_1(x,y,\xi(x))$. We therefore see that to show that ξ is semi-algebraic is suffices to show that the set

$$\{(y,x) \in \mathbb{Q}_p^{r+m} \mid \exists z \in \mathbb{Q}_p . f(x,y,\xi(x)) = z^n\}$$

is, for f of degree $\leq d$ and arbitrary n, semi-algebraic. Having reduced to the case where f has degree $\leq d$, we may use cell decomposition `II` to verify this. We can then prove, in the locale `denef_II`, the following theorem:

```
theorem(in denef_II) SA_fun_test:
  assumes g_deg_bound:"deg (SA m) g ≤ Suc d"
  assumes g_deg_pos:"deg (SA m) g > 0"
  assumes g_closed:"g ∈ carrier (UP (SA m))"
  assumes ξ_fun:"ξ ∈ carrier (Fun_m Q_p)"
  assumes g_ltrm_Unit:
    "UP_ring.lcf (SA m) g ∈ Units (SA m)"
  assumes ξ_root:
    "∀x ∈ carrier (Q_p^m).
        (SA_poly_to_SA_fun m g) ((ξ x)#x) = 0"
  assumes val_leq_inv_im:
    "⋀ k c a.
    [|c ∈ carrier (SA (m+k)); a ∈ carrier (SA (m+k))|]
    ⟹
    is_semialgebraic (m+k)
    {x ∈ carrier (Q_p^(m+k)). val (ξ (take m x) ⊖ c x)
                              ≤ val (a x)}"
  assumes pow_res_inv_im:
  "⋀ k c α n.
  [|c ∈ carrier (SA (m+k)); α ∈ carrier (Q_p) ; n > 0 |]
                              ⟹
    is_semialgebraic (m+k)
    {x ∈ carrier (Q_p^(m+k)).(ξ (take m x) ⊖ c x) ∈ pow_res n α}"

  shows "ξ ∈carrier (SA m)".
```

This will be the proof method called in the formal proofs of both lemma 2.3 and 2.4 and formally captures the logic of the statement Denef makes in the proof of Lemma 2.4 “As in the proof of Lemma 2.3, after Euclidean division, [...] we see that if suffices to prove that the relations [...] are semi-algebraic...” [19].

The assumption `g_ltrm_Unit` in the above lemma states that the leading term of the polynomial g is a semi-algebraic unit, i.e. that it only takes nonzero values. This is technically not needed, but is helpful as it guarantees that the degree of the polynomials over $\mathbb{Q}_p$ which are obtained by evaluating the coefficients will always have the same degree as the semi-algebraic polynomial.

9.4 Lemmas 2.3 and 2.4

Lemma 2.3 (our lemma 9.0.1) shows that under certain circumstances, a semi-algebraic function which is a root of a semi-algebraic polynomial is again semi-algebraic. Since the lemma contains many parameters and assumptions, and assumes cell decomposition theorem `II_d`, it is natural to organize these assumptions into a locale which extends the locale `denef_II`,

```
locale denef_lemma_2_3 = denef_II +
  fixes g ξ e l m
  assumes DL_2_3_1:"deg (SA m) g ≤ Suc d"
  assumes DL_2_3_2:"deg (SA m) g ≥ 0"
  assumes DL_2_3_3:"g ∈ carrier (UP (SA m))"
  assumes DL_2_3_4:"⋀j. g j ∈ carrier  (Q_p^m) → 𝒪_p"
  assumes DL_2_3_5:"ξ ∈ carrier (Fun_m Q_p) ∧ ξ ∈ carrier (Q_p^m)
    → 𝒪_p"
  assumes DL_2_3_6:"UP_ring.lcf (SA m) g ∈ Units (SA m)"
  assumes DL_2_3_7:"∀x ∈ carrier (Q_p^m). (SA_poly_to_SA_fun m
    g) (ξ x#x) = 0"
  assumes DL_2_3_8:"∀x ∈ carrier (Q_p^m). to_Zp (ξ x) (Suc e)
    = l"
  assumes DL_2_3_9:"∀x ∈ carrier (Q_p^m). val (
   SA_poly_to_SA_fun m (UP_cring.pderiv (SA m) g) (ξ x#x))
   ≤ e"
  assumes DL_2_3_10: "m ≥ 0".
```

Within this locale, we can then state the conclusion of the lemma succinctly:

```
lemma denef_lemma_2_3:
"ξ ∈ carrier (SA m)"
```

and the proof can be automatically structured by invoking the generic lemma `SA_fun_test` as described in section 9.3. To simplify the structure of the proof, the goals generated by the invocation of this rule are proved within the `denef_lemma_2_3` locale prior to the proof:

```
lemma pow_res_inv_im:
  assumes "c ∈ carrier (SA (m+k))"
  assumes "b ∈ carrier Q_p"
  assumes "n > 0"
  shows "is_semialgebraic (m+k) {x ∈ carrier (Q_p^{m+k}). (ξ (
   take m x) ⊖ c x) ∈  pow_res n b}"

lemma val_leq_inv_im:
  assumes "c ∈ carrier (SA (m+k))"
  assumes "a ∈ carrier (SA (m+k))"
  shows "is_semialgebraic (m+k) {x ∈ carrier (Q_p^{m+k}). val (ξ
   (take m x) ⊖ c x) ≤ val (a x)}".
```

The proof of Lemma 2.4 (our lemma 9.0.2) is essentially structured in exactly the same way as Lemma 2.3. The generic semi-algebraicity test from section 9.3 can be used, which allows us to divide the proof into lemmas showing two particularly simple sets are semi-algebraic. The proof is also performed within a special locale fixing relevant parameters.

Chapter 10

Proof Sketches for Cell Decomposition Theorems and Macintyre's Theorem

10.1 Sketching the Proof of Decomposition Theorem I_{d+1} from Theorems I_d and II_d

In this section we will give an enumerated outline of the proof of theorem I_{d+1}, which we will refer back to in the next section in our exposition of how this is formalized in Isabelle. In some cases, the choices to parse one part of the proof into distinct steps reflects the logic of our formalisation. First, we will introduce some shorthand for ease of exposition.

Definition 10.1.1. Let $f(x,t)$ be a semi-algebraic polynomial in variable t, with $x = (x_1, \ldots, x_m)$ and C a cell in $\mathbb{Q}_p^m \times \mathbb{Q}_p$ with center $c(x)$. We say that f is uniformly bounded on C if there exists an $N \in \mathbb{Z}$ such that for all $(x,t) \in C$:

$$\mathrm{ord} f(x,t) \leq \mathrm{Min}_i \ \mathrm{ord}[a_i(x)(t - c(x))^i] + N$$

where $a_i(x)$ denotes the i^{th} Taylor coefficient of f, expanded at $c(x)$.

Then we can summarize theorem I_d as saying that for any polynomial $f(x,t)$ of degree $\leq d$, there exists a cell decomposition S of $\mathbb{Q}_p^m$ such that f is uniformly bounded on each $A \in S$. Notice that if f is uniformly bounded on a cell A, then it is automatically uniformly bounded on any cell $D \subseteq A$, provided D has the same center.

Then, the proof proceeds as follows: assuming that we have fixed some polynomial $f(x,t)$ of degree $\leq d+1$.

Step 1. (Obtain Initial Decomposition) We can apply theorem I_d to f' to get a cell decomposition where f' is uniformly bounded on each cell A in our decomposition.

Step 2. By further refinement of the cells, we can assume that the Taylor coefficients $a_i(x)$ of $f(x,t)$ expanded at $c(x)$ are either identically zero or never zero at all points $(x,t) \in A$, for any cell A in our decomposition.

Step 3. For a fixed cell A with center $c(x)$ in the prior decomposition, define A_0 to be

$$A_0 := \{(x,t) \in A \mid \mathrm{ord}(t - c(x)) \neq \frac{1}{i-j}\mathrm{ord}\left(\frac{a_j(x)}{a_i(x)}\right) \text{ for all } i \neq j, \text{ with } i,j \in I\}$$

where I is the set of indices at which $a_i(x)$ is not identically zero on A. Since $A_0 \subseteq A$, to finish the proof we can show that both A_0 and $A \setminus A_0$ can be decomposed into cells on which f is uniformly bounded.

Step 4 (Obtain an initial decomposition of $A - A_0$)**.** Inspecting the definition of A_0 and using Lemma 2.3, we can infer that $A - A_0$ can be decomposed into cells of the form

$$\{(x,t) \mid x \in C \text{ and } \mathrm{ord}(t - c(x)) = \theta(x)\}$$

where $C \subseteq \mathbb{Q}_p^m$ and $\theta(x)$ is a semi-algebraic unit.

Step 5 (Decomposing A_0)**.** By the prior step, we can also decompose A_0 into cells with center $c(x)$, since A is such a cell, and A_0 can be decomposed into these cells. By definition of A, it follows that f is uniformly bounded on each cell in this decomposition.

At this point, all that remains is to refine the decomposition from Step 4 into smaller cells on which f is uniformly bounded. We fix a single cell B from this partition, which is of the form,

$$B := \{(x,t) \mid x \in C \text{ and } \mathrm{ord}(t - c(x)) = \theta(x)\}$$

Step 6 (Partition B to obtain a uniform minimal coefficient)**.** After partitioning B into finer cells by partitioning its fibre set C, we may assume that for all $x \in C$, there is one $i_0 \in \mathbb{N}$ such that for all $x \in C$:

$$\mathrm{ord} a_{i_0}(x)\theta(x)^{i_0} = \mathrm{Min}_i \ \mathrm{ord} a_i(x)\theta(x)^i$$

Step 7 (Reduce to the case where $i_0 > 0$)**.** After partitioning again, we can assume without loss of generality that either $i_0 = 0$ and for all $j > 0$,

$$\text{ord}a_0(x) < \text{ord}a_j(x)\theta(x)^j$$

or $i_0 > 0$. In the case of $i_0 = 0$, the conclusion of theorem I_{d+1} can be established easily. The case $i_0 > 0$ requires more work and is the focus of the rest of the proof.

Step 8 (Change variables)**.** By setting $u = \frac{t-c(x)}{\theta(x)}$, and

$$g(x,u) = \frac{f(x,t)}{a_{i_0}\theta(x)^{i_0}}$$

We may replace the cell B from Step 5 with:

$$B' := \{(x,u) \in \mathbb{Q}_p^m \times \mathbb{Q}_p \mid x \in C \text{ and } \text{ord}(u) = 0\}$$

and we have that $g = b_0(x) + b_1(x)u + \cdots + b_i(x)u^i + \ldots$, with $b_{i0}(x) = 1$, and every coefficient b_i taking values in the valuation ring. The proof can now be completed by decomposing B' into finer cells which satisfy I_{d+1} for the polynomial g.

Step 9 (Disjunction over Residue Ring)**.** The final step requires partitioning B' further. First, we obtain a parameter e such that $0 \leq \text{ord}(g'(x,u)) \leq e$ for all $(x,u) \in B'$. Now we can decompose B' according to the residue classes of the parameters u and $g(x,u)$ to obtain the final desired decomposition.

10.2 Sketching the Proof of Decomposition Theorem II_{d+1} from I_{d+1}

The statement of theorem II_{d+1} applies to a finite collection $f_1, \ldots, f_r$ of polynomials. The proof proceeds by induction on the number r of polynomials, and we sketch the structure in this section. Steps enumerated here will be referred back to in the exposition of the corresponding formalisation.

10.2.1 The Base Case $r = 1$

In this case we assume that there is only one polynomial f_1 of degree $\leq d+1$, which we would like to factor on each cell in a cell decomposition. As in the previous proof, it proceeds by successive refinements of an initial cell decomposition until the desired property is attained. The power $n \geq 2$ in the factorization is fixed in the background, and we also fix a parameter $\lambda \geq 1$ such that for all $x \in \mathcal{O}$, $x \equiv 1 \mod p^\lambda$ implies that x is an n^{th} power.

Base Case Step 1. Apply theorem I_{d+1} to the polynomial $f := f_1$ to obtain a cell decomposition such that for each cell $\mathcal{C}$ in the decomposition, and all $(x,t) \in \mathcal{C}$:

$$\text{ord} f(x,t) \leq \text{Min}_i \ \text{ord}[a_i(x)(t-c(x))^i] + N$$

where $c(x)$ is the center of $\mathcal{C}$ and $a_i(x)$ are the Taylor coefficients of $f(x)$ at $c(x)$.

Base Case Step 2. By further refinement of the cells, we can assume that the Taylor coefficients $a_i(x)$ of $f(x,t)$ expanded at $c(x)$ are either identically zero or never zero at all points $(x,t) \in A$, for any cell A in our decomposition.

Base Case Step 3. Further partition each cell into finer cells so that on each cell $\mathcal{C}$ there is a uniform $i_0 \in \mathbb{N}$ such that for all $(x,t) \in \mathcal{C}$:

$$\text{ord}[a_{i_0}(x)(t-c(x))^{i_0}] = \text{Min}_i \ \text{ord}[a_i(x)(t-c(x))^i]$$

Base Case Step 4. For each cell A of our current decomposition, with center $c(x)$ and Taylor coefficients $a_i(x)$ for $f(x,t)$ at $c(x)$, define a subset

$$A_0 := \{(x,t) \in A \mid \lambda + \text{ord}[a_{i_0}(x)(t-c(x))^{i_0}] \leq \text{Min}_{i \neq i_0} \ \text{ord}[a_i(x)(t-c(x))^i]\}$$

of A. We now decompose both $A \setminus A_0$ and A_0 separately into cells of the desired form.

Base Case Step 5. For each cell A with center $c(x)$ we can decompose $A \setminus A_0$ into cells of the form

$$\{(x,t) \mid x \in C, \text{ord}(t-c(x)) = \text{ord}\theta(x)\}$$

For a semialgebraic function $\theta(x)$.

Base Case Step 6. Using that boolean combinations of cells with a fixed center $c(x)$ admit decompositions into cells with center $c(x)$, we can use the previous step to decompose each A_0 into cells centered at the center $c(x)$ of the cell A. The definition of A_0 makes it so that on each of these cells our desired conclusion holds

All that remains is to show that a cell of the form

$$B = \{(x,t) \mid x \in C, \text{ord}(t-c(x)) = \text{ord}\theta(x)\}$$

can be decomposed into cells of the desired form. We assume there is a fixed such cell.

Base Case Step 7. Change variables as in the proof of I_{d+1} to $u = \frac{t-c(x)}{\theta(x)}$. This transforms B into the cell

$$B' = \{(x,u) \mid x \in C, \mathrm{ord}(u) = 0\}$$

and if we replace f with

$$g(x,u) := \frac{f(x,t)}{a_{i_0}(x)\theta(x)^{i_0}}$$

and let $b_i(x)$ denote the Taylor coefficients of g at 0 (in the variable u), then g satisfies the conclusion of the theorem on B' if and only if f does on B. Therefore, we can shift our attention to g and B', noting that $b_{i_0} = 1$ and that $b_i(x) \in \mathcal{O}$ for all $xinC$.

Base Case Step 8. The boundedness of f established in Step 1 implies that we have $\mathrm{ord}(g(x,u)) \leq \kappa$ for some $\kappa \in \mathbb{Z}$ and all $(x,u) \in B'$. Partition the cell B' into finer cells so that the residues of u and $b_i(x)$ are constant for each $(x,u) \in B'$ and $i \leq deg(g)$, modulo $p^{\kappa+\lambda}$. We now can argue that on each of these new cells the desired property holds for g, completing the proof.

10.2.2 The Induction Step

We now want to decompose $\mathbb{Q}_p^{m+1}$ so that an entire collection $f_1, \ldots, f_r$ of polynomials factors appropriately on each cell.

Inductive Step 1 (Apply Base Case)**.** We can apply the base case to each f_i separately to obtain a cell decomposition on which f_i has the desired property. The issue then is that these cells may be different for each i. Intersecting all of these cells with one another, we obtain a semi-algebraic partition of our domain such that each partition element is an intersection of cells $A_1, \ldots A_r$, and each f_i factors appropriately on A_i. Some of the intersected cells may have the same center, and some may not.

Inductive Step 2 (Reduce the Number of Centers)**.** We have a set A which is the intersection of cells. These cells have distinct centers $c_1, \ldots, c_s$. If $s = 1$, then we can repartition A as desired (since boolean comnbinations of cells with the same center admit cell decompositions with the same center). The goal then becomes iteratively partitioning A into finer sets which themselves are the intersections of cells prepared for each f_i, such that the number of distinct centers for each intersection is just 1. If we can show how to reduce s by 1, then this will be accomplished.

10.3 Sketching Denef's Proof of Macintyre's Theorem

As in our accounts of the proofs of the cell decomposition theorems, we can start our account of Macintyre's Theorem by factorizing the informal proof into enumerated steps, so that we may refer back to those steps in our exposition of our formal proof:

1. First, we formulate the precise version of the theorem we aim to prove. We fix a natural number m, a semi-algebraic set $S \subseteq \mathbb{Q}_p^{m+1}$. The goal is to show that the set
$$P = \{x \in \mathbb{Q}_p^m \mid \exists t \in \mathbb{Q}_p : (x,t) \in S\}$$
is semi-algebraic.

2. Second, using the defintition of semi-algebraic sets, we may assume that S is a disjunction of relations of the form
$$\text{ord}(a_1(x))\Box_1\text{ord}(t-c(x))\Box_2\text{ord}(a_2(x)) \text{ and } x \in C \text{ and}$$
$$h_i(x)(t-c(x))^{\nu_i} \text{ is (is not) an } n_i\text{-th power, for } i = 1, \dots, k$$
with a_1, a_2, c, h all semi-algebraic functions, and $\Box_i$ either $\leq, <$, or no condition. Since projections commute with finite unions, we may even assume that S consists of just one set in the above form.

3. We make a disjunction over n^{th} power residues of $h_i(x)$ and $t - c(x)$ so that without loss of generality S is of the form:
$$\text{ord}(a_1(x)) \leq \text{ord}(t-c(x)) \leq \text{ord}(a_2(x)) \text{ and } x \in C \text{ and}$$
$$t - c(x) = \varrho \cdot (\text{ nonzero n-th power}),$$
for some $\varrho \neq 0$.

4. For a fixed x, there exists a t such that (x,t) satisfies the above relations if and only if
$$\exists l \in \mathbb{Z} : \text{ord}(a_1(x)) \leq l \leq \text{ord}(a_2(x)) \text{ and } l \equiv \text{ord}(\varrho) \mod n$$
Clearly $l = \text{ord}(t - c(x))$ would be such an l for a given (t,x), and conversely, if such an l exists then we can set $t = c(x) + \varrho \cdot p^{l-\text{ord}(\varrho)}$. So we may assume that S is defined by a relation of the above form, which can then be re-expressed as

$$\exists l \in \mathbb{Z} : \frac{\operatorname{ord}(a_1(x)\varrho^{-1})}{n} \leq l \leq \frac{\operatorname{ord}(a_2(x)\varrho^{-1})}{n}$$

5. We perform a disjunction over the possible values of $\operatorname{ord}(a_1(x)\varrho^{-1}) \mod n$ to reduce P to a set of the form

$$\{x \in \mathbb{Q}_p^m \mid \operatorname{ord}(a_1(x)\varrho^{-1}) \leq \operatorname{ord}(a_2(x)\varrho^{-1})\} \cap \\ \{x \in \mathbb{Q}_p^m \mid \operatorname{ord}(a_1(x)\varrho^{-1}) \mod n = 0\}$$

or

$$\{x \in \mathbb{Q}_p^m \mid \operatorname{ord}(a_1(x)\varrho^{-1}) + n - \gamma \leq \operatorname{ord}(a_2(x)\varrho^{-1})\} \cap \\ \{x \in \mathbb{Q}_p^m \mid \operatorname{ord}(a_1(x)\varrho^{-1}) \mod n = \gamma\}$$

for some $0 < \gamma < n$. In both cases the set is semi-algebraic, which completes the proof.

Chapter 11

Formalizing the Cell Decomposition Proofs

The proofs of the decomposition theorems are structured by a series of locales which reflect the successive reductions made in their proofs. They are also structured in such a way that certain locales can be re-used between the two main proofs to avoid redundancy. The two main proofs are that theorem I_d and II_d imply theorem I_{d+1}, and that theorem I_{d+1} implies II_{d+1}. For simplicity, we will refer to these as the proofs of theorem I_{d+1} and II_{d+1} respectively, with the background assumptions being implied. In addition, based on the inductive structure of the final proofs, we can also add I_d and II_d as background assumptions for the proof of II_{d+1}. The theory `Cell_Decomp_Helper_Lemmas` sets up a common library of results and context that are needed in both of the proofs. To start we create a common locale for both proofs which we call `common_decomp_proof_context`, which can be extended as needed for the more specific contexts we will need later:

```
locale common_decomp_proof_context = denef_I + denef_II
```

11.1 Refining for Unit Taylor Coefficients

Step 2 of the proof of I_{d+1} and step 2 of the base case of the proof of II_{d+1} both involve taking a cell decomposition for a single polynomial of degree $\leq d+1$ and further decomposing so that the Taylor coefficients of f expanded at cell centres are either always zero or never zero. We can therefore create a common locale for both of these situations, which again can be extended and interpreted in either context as needed:

```
locale common_refinement_locale =
   common_decomp_proof_context +
  fixes 𝒞 c a1 a2 I f m
  assumes f_closed: "f ∈ carrier (UP (SA m))"
  assumes f_deg: " deg (SA m) f ≤ (Suc d)"
  assumes 𝒞_def: "𝒞 = Cond m A c a1 a2 I"
  assumes 𝒞_cond: "is_cell_condition 𝒞
  assumes f_taylor_cfs:
    "⋀ i. (taylor_expansion (SA m) c f i = 0SA m) ∨
          (taylor_expansion (SA m) c f i ∈ Units (SA m))"
```

We make the stronger assumption here that the Taylor coefficients are not just uniformly zero or nonzero on the cell in question, but that this holds globally for f so that it's coefficients are either the zero function or are semi-algebraic units. This simplifies the proof since we can divide by semi-algebraic units without restriction, and the zero function evaluates to 0 provably without having to worry about whether the points of evaluation are in the correct subset of $\mathbb{Q}_p^m$. To match the notation of the source paper, we define a new polynomial $a :=$ `taylor_expansion (SA m) c f`, so that the i^{th} term in the Taylor expansion of f, evaluated at a point $(t, x) \in \mathbb{Q}_p^{1+m}$ will be given (in HOL-Algebra's notation) by the expression `(a i x)⊗(t⊖c x)[^]i`.

11.2 Fixing the Order Type of Taylor Expansion Terms

There is some detail missing in Denef's exposition of what we call Step 3 in the base case of the proof of theorem I_{d+1}. The fact that we can perform this finer cell decomposition so that there is some degree i_0 whose term in the Taylor expansion is always minimal with respect to the valuation is simply said to follow from Lemmas 2.4 and 2.1, without further explanation. While we do not know the specific proof that the author had in mind, we can achieve this result using the specified lemmas, borrowing techniques from the proofs of steps 3 to 6 in the proof of theorem I_{d+1}. While a plain english paper has the luxury of alluding to similar processes, in a formal proof we must make these similarities explicit. To avoid the inelegant of copy-pasting of one proof and making the necessary changes to fit both cases, we instead formalize this construction in `common_refinement_locale`, so that it can be used in both proofs without repetition. This forms the bulk of the development within the theory `Cell_Decomp_Theorem_Helpers`. The basic outline of this theory is as follows:

1. One works within `common_refinement_locale`. In this context, we may partition the set `condition_to_set𝒞` according to the set `A`$_0$:

```
definition A_0 where
"A_0 = {x ∈ condition_to_set 𝒞.
    (∀ i ∈ inds. (∀ j ∈ inds. i < j
    ⟶
        val (a i (tl x)) ≠
        val (a j (tl x) ⊗ (hd x ⊖ c (tl x))[^](j-i)))
  }"
```

which corresponds to the set defined in Step 3 of the proof outline for theorem I_{d+1}. Note that we have moved some terms around in the definition since our valuation does not allow division. The set `inds` here refers to those degrees `i` for which the Taylor expansion of f at c is a semi-algebraic unit (as opposed to being identically zero). We can prove that this set is semi-algebraic since it is an intersection of semi-algebraic sets.

2. We can obtain a decomposition as described in Step 4 of `condition_to_set` $\mathcal{C} \setminus$ `A`$_0$. To do this, we construct a function $\Theta(i,j)$ such that for all $i,j \in$ `inds`, with $i < j$, the value $\Theta(i,j)$ is a semi-algebraic unit such that for all $(t,x) \in$ `condition_to_set` $\mathcal{C}$,

$$\mathrm{val}(a_i(x)) \neq \mathrm{val}(a_j(x)(t-c(x))^{j-i}) \iff \mathrm{val}(t-c(x)) = \mathrm{val}(\Theta(i,j)(x)).$$

It is clear that we can then cover (`condition_to_set` $\mathcal{C} \setminus$ `A`$_0$) with finitely many cells of the appropriate form. We can then use results described in Section 8.10 to construct a fixed cell decomposition, which we name `A`$_0$`_comp_decomp` which satisfies the following lemma:

```
lemma A_0_comp_decomp:
"(is_cell_decomp m A_0_comp_decomp (condition_to_set 𝒞 \ A_0) ∧
 (∀B ∈ A_0_comp_decomp.
       (∃ ϕ. ϕ ∈ Units (SA m) ∧
             center B = c ∧ l_bound B = ϕ ∧ u_bound B
    = ϕ ∧
             boundary_condition B = closed_interval))
  )"
```

which establishes the goal of Step 4 in the proof of Theorem I_{d+1}.

3. Using the lemma `cell_decomp_same_center` described in Section 8.10, we can

immediately obtain a cell decomposition of the set A$_0$, where all constituent cells have the same center $c(x)$, using that both the cell $\mathcal{C}$ itself and the complement (condition_to_set $\mathcal{C} \setminus$ A$_0$) admit such a decomposition:

```
lemma A0_decomp,:
assumes "inds ≠ {}"
shows "∃S. is_cell_decomp m S A0 ∧
            (∀B ∈ S. center B = c ∧
            (∃N. SA_poly_ubounded p m f (center B) (
 condition_to_set B) N))"
```

where the assumption that the set inds is nonempty is only present because otherwise the set A$_0$ would be the entire cell $\mathcal{C}$.

We can now refine the decompositions we obtained above to also satisfy the requirements of the proof of Step 3 in the base case of theorem II_{d+1}. To simplify proofs, we develop a predicate for cells which have this property:

```
      definition has_minimal_i where
      "has_minimal_i 𝓑 = (∃i0 . (∀j. ∀t. ∀x. t#x ∈
 condition_to_set 𝓑
         ⟶ val ((a i0 x)⊗(t ⊖ c x)[^]i0) ≤ val ((a j
  x)⊗(t ⊖ c x)[^]j)))".
```

4. For the cells in A$_0$_comp_decomp, this is relatively easy because on each cell, there is a semi-algebraic unit $\varphi(x)$ such that for each i for which $a_i(x)$ is not identically zero, the valuation of $a_i(x)(t - c(x))^i$ is equal to the valuation of $a_i(x)\varphi(x)^i$, which is a semi-algebraic function in the $\mathbb{Q}_p^m$-variable x. We therefore only need to decompose each cell in A$_0$_comp_decomp further by decomposing its fibre set sufficiently finely so that the ordering of the valuations of each $a_i(x)\varphi(x)^i$ is fixed on each piece. This is codified in the following lemma:

```
      lemma A0_comp_minimal_i_decomp:
assumes "inds ≠ {}"
shows "∃ S. is_cell_decomp m S (condition_to_set 𝓒 \ A0)∧
            (∀ 𝓑 ∈ S. has_minimal_i 𝓑 ∧
                      (∃ ϕ i0. ϕ ∈ Units (SA m) ∧
 center 𝓑 = c ∧ l_bound 𝓑 = ϕ  ∧
                                      u_bound 𝓑 = ϕ ∧
 boundary_condition 𝓑 = closed_interval ∧
```

```
                                          (∀j. ∀t. ∀x.
                                          t#x ∈
   condition_to_set 𝓑 ⟶
                                          val ((a i_0 x)⊗(ϕ x)
   [^]i_0) ≤ val ((a j x)⊗(ϕ x)[^]j))))".
```

This also satisfies the requirements of Step 6 of Theorem I_{d+1}.

5. Refining the cells in a decomposition of $\mathtt{A}_0$ so that `has_minimal_i` $\mathcal{B}$ holds for each $\mathcal{B}$ in the decomposition is slightly more involved, and draws on tools again related to algebras of cells discussed in Section 8.10.3.

6. We can now prove the lemma that will be needed in the proof of Theorem II_{d+1} about the existence of a unique index i_0, within the locale `common_refinement_locale`:

```
        lemma 𝒞_comp_minimal_i_decomp:
 shows "∃ S. is_cell_decomp m S (condition_to_set 𝒞)
 ∧ (∀ 𝓑 ∈ S. center 𝓑 = c ∧ has_minimal_i 𝓑)".
```

11.3 Cell Normalization

The proofs of both Theorems I_{d+1} and II_{d+1} (at Step 8 and Base Case Step 7, respectively) involve taking a cell of the form

$$\mathcal{B} = \{(x,t) \mid x \in C, \mathrm{ord}(t - c(x)) = \mathrm{ord}\theta(x)\}$$

and performing the change of variables $u = \frac{t-c(x)}{\theta(x)}$ to obtain a new cell in the variables (u, x) of the form:

$$\mathcal{B}' = \{(x,u) \mid x \in C, \mathrm{ord}(u) = 0\}.$$

One then defines the polynomial

$$g(x,u) := \frac{f(x,t)}{a_{i_0}(x)\theta(x)^{i_0}}$$

where i_0 has the valuative minimality property discussed in the previous section on all of the cell $\mathcal{B}$. Many of the results needed in these proofs are common to both, and we gather them in a theory named `Cell_Normalization`. In this theory, we first create a locale establishing the underlying context of a cell $\mathcal{B}$ as above, with a polynomial f expanded at the center c of the cell:

```
locale one_val_point_cell = padic_fields +
  fixes 𝒞 a b c φ m f
  assumes f_closed: "f ∈ carrier (UP (SA m))"
  assumes φ_Unit: "φ ∈ Units (SA m)"
  assumes 𝒞_cell: "is_cell_condition 𝒞"
  assumes 𝒞_eq: "𝒞 = Cond m b c φ φ closed_interval"
  assumes a_def: "a = taylor_expansion (SA m) c f"
  assumes b_nonempty: "b ≠ {}".
```

We can further extend this locale to reflect the context where the Taylor coefficients of f at c are either semi-algebraic units or constantly zero:

```
locale refined_one_val_point_cell = one_val_point_cell +
  fixes inds :: "nat set"
  assumes inds_memE: "⋀ j. j ∈ inds ⟹ a j ∈ Units (SA m)
   "
  assumes inds_non_memE: "⋀ j. j ∉ inds ⟹ a j = 0_{SA m}
  assumes inds_nonempty: "inds ≠ {}"
```

and finally we can extend further again by adding the minimality assumption on the index i_0:

```
locale cell_normalization = refined_one_val_point_cell +
  fixes i_0 :: nat
  assumes i_0_min: "⋀ j t x. t#x ∈ condition_to_set 𝒞
                              ⟹ val ((a i_0 x)⊗(t ⊖ c x)[^]
   i_0) ≤ val ((a j x)⊗(t ⊖ c x)[^]j)".
```

Within `cell_normalization`, we can define the polynomial g:

```
definition α where α_def:
"α = inv_{SA m}a i_0 ⊗_{SA m} φ _SA m - int i_0"

definition g where g_def:
"g = α ⊙_{UP (SA m)}compose (SA m) a (up_ring.monom (UP (SA m))
   φ 1)".
```

In the above, the constant α is just the semi-algebraic function $\frac{a_{i_0}(x)}{\varphi(x)^{i_0}}$, the term `up_ring.monom (UP (SA m))`$\varphi$ `1` denotes the monomial $\phi(x)t$ in the variable t, and the function `compose (SA m)` represents polynomial composition. We can also define a semi-algebraic function `u` which corresponds to the change of variables, and prove the lemma characterizing its evaluation:

```
lemma u_eval:
  assumes "t#x ∈ condition_to_set 𝒞"
  shows "u(t#x) = inv (φ x) ⊗ (t ⊖ c x)".
```

We similarly define an inverse function `u_inv` for `u` and prove that they satisfy the inverse relationship:

```
lemma u_inv_u:
  assumes "t#x ∈ condition_to_set 𝒞"
  shows "u_inv(u(t#x)#x) = t" .
```

Given a set of cell fibres `b` we also define a function which maps this to the cell of the form $\mathcal{B}'$ above:

```
definition normal_cell where
"normal_cell m b =  Cond m b (1_{SA m}) (1_{SA m}) (0_{SA m})
   closed_interval".
```

For the purposes of this formalisation, we will also need a way to map cells which are contained in the cell $\mathcal{B}$ to cells which are contained in $\mathcal{B}'$, such that applying this mapping to a cell decomposition of $\mathcal{B}$ will product a cell decomposition of $\mathcal{B}'$, and the desired properties of each cell in the statements of Theorems I_{d+1} and II_{d+1} will transfer under this mapping as well (for the polynomial f for cells contained in $\mathcal{B}$ and for g for cells contained in $\mathcal{B}'$). To this end, we define a transformation on cell conditions, along with an inverse for this function:

```
fun normalize_cell where
"normalize_cell (Cond n C d a1 a2 I) =
    Cond n C ((inv_{SA m}φ)⊗_{SA m}(d ⊖_{SA m}c))
                  ((inv_{SA m}φ) ⊗_{SA m} a1) ((inv_{SA m}φ) ⊗_{SA m} a2)
    I"

fun normalize_cell_inv where
"normalize_cell_inv (Cond n C d a1 a2 I) =
      Cond n C ((φ ⊗_{SA m} d) ⊕_{SA m}c) (φ ⊗_{SA m} a1) (φ ⊗_{SA m} a2) I
   ".
```

The following lemmas (proved within the locale `cell_normalization` justify why we can prove the conclusions of theorems I_{d+1} and II_{d+1} on cells of the form $\{(x,t) \mid x \in$

$C, \mathrm{ord}(t - c(x)) = \mathrm{ord}\theta(x)\}$ by proving these instead on normalized cells. First, a cell decomposition of a normalized cell maps to a decomposition of the original cell:

```
lemma transfer_cell_decomp:
  assumes "is_cell_decomp m S (condition_to_set (
   normal_cell m b))"
  shows "is_cell_decomp m (normalize_cell_inv ‘ S) (
   condition_to_set 𝒞)".
```

Secondly, if a cell contained in a normalized cell satisfies the axioms of the `SA_poly_ubounded` locale for the polynomial `g`, then that cell mapped under `normalize_cell_inv` will satisfy this for the polynomial `f`:

```
lemma transfer_SA_poly_ubounded1:
  assumes "is_cell_condition ℬ"
  assumes "arity ℬ = m"
  assumes "SA_poly_ubounded p m g (center ℬ) (
   condition_to_set ℬ) N"
  shows "SA_poly_ubounded p m f (center (
   normalize_cell_inv ℬ))
                                  (condition_to_set (
   normalize_cell_inv ℬ)) N".
```

Finally, we have a similar result for the locale `SA_poly_factors`:

```
lemma transfer_SA_poly_factors1:
  assumes "is_cell_condition ℬ"
  assumes "arity ℬ = m"
  assumes "∃u h k. SA_poly_factors p m n g (center ℬ) (
   condition_to_set ℬ) u h k"
  shows "∃u h k. SA_poly_factors p m n f (center (
   normalize_cell_inv ℬ))
                                   (condition_to_set (
   normalize_cell_inv ℬ)) u h k".
```

These in turn easily translate into more user-friendly lemmas about the existence of the desired cell decompositions:

```
lemma transfer_decomp_ubounded:
  assumes "∃S. is_cell_decomp m S (condition_to_set (
   normal_cell m b)) ∧
                  (∀ C ∈ S. ∃N. SA_poly_ubounded p m g (
   center C)
```

```
                                    (condition_to_set C) N)"
  shows "∃S. is_cell_decomp m S (condition_to_set 𝒞) ∧
                  (∀ C ∈ S. ∃N. SA_poly_ubounded p m f (
   center C)
                                    (condition_to_set C) N)"

lemma transfer_decomp_poly_factors:
  assumes "is_cell_condition ℬ"
  assumes "arity ℬ = m"
  assumes "∃S. is_cell_decomp m S (condition_to_set (
   normal_cell m b)) ∧
                  (∀ C ∈ S. ∃u h k. SA_poly_factors p m n g
   (center C) (condition_to_set C) u h k)"
  shows "∃S. is_cell_decomp m S (condition_to_set 𝒞) ∧
                  (∀ C ∈ S. ∃u h k. SA_poly_factors p m n f
   (center C) (condition_to_set C) u h k)".
```

11.4 Proving Theorem I_{d+1}

11.4.1 Obtaining the Conclusion in Stricter Locales

The developments in the theory `Cell_Decomp_Theorem_Helpers` outline the major steps needed for proving I_{d+1} up to 6. From Step 7 onward, we only need to obtain a decomposition of the set A_0 into cells of the appropriate form. Toward that end, we extend the locale `common_refinement_locale` to one which reflects this specialized context:

```
locale A0_comp_refinement = common_refinement_locale +
  fixes B b φ i H
  assumes φ_Unit: "φ ∈ Units (SA m)"
  assumes B_cell: "is_cell_condition B"
  assumes B_eq: "B = Cond m b c φ φ closed_interval"
  assumes b_subset: "condition_to_set B ⊆ condition_to_set
    𝒞 - A0"
  assumes i_inds: "i ∈ inds"
  assumes H_def: "H = (λi. a i ⊗SA mφ[^]SA m i)"
  assumes H_ineq: "⋀j x. j ∈ inds ⟹ x ∈ b ⟹ val (H i x)
    ≤ val (H j x)"
  assumes b_nonempty: "b ≠ {}"
  assumes static: "static_order_type (H ‘ inds) b".
```

In the above, we are using the constant name i in place of the constant i_0 from Denef's paper. Our main goal in this locale is to decompose the underlying set of the cell `B` into the desired form. The locale axiom `H_ineq` expressed the statement that the constant i is minimal in the sense of Step 6. The axiom `static` expresses an even stronger property than this, essentially saying that the ordering relations between $val(a_i(x)\phi(x)^i)$ and $val(a_j(x)\phi^j)$ for any two natural numbers i, j are fixed for all possible $x \in b$ (b being the fibre set of the cell B). This stronger property is needed for the reduction in Step 7 to the case that $i > 0$.

The existence of the minimal constant `i` in `A₀_comp_refinement` is guaranteed by theorem `A₀_comp_minimal_i_decomp` described in the previous section, but it may not be unique. The axiom `static` guarantees that if there is some other j and a point $x \in b$ such that $val(a_i(x)\phi(x)^i) = val(a_j(x)\phi^j)$ at x, then we could replace i with j and the axioms of the locale would still be satisfied. This means that if we ended up in a situation where $i =$, we could either replace i with a nonzero value, or it would be the case that $val(a_0(x)) < val(a_j(x)\phi^j)$ holds for all $j > 0$ and all $x \in b$. This dichotomy allows us to further refine `A₀_comp_refinement` into two locales to deal with the positivity of the index i:

```
locale A₀_comp_refinement_i_pos = A₀_comp_refinement +
  assumes deriv_bounded: "∃N. SA_poly_ubounded p m (UPSA.
   pderiv m f) c (condition_to_set 𝒞) N"
  assumes i_nonzero: "i ≠ 0"

locale A₀_comp_refinement_i_zero = A₀_comp_refinement +
  assumes deriv_bounded: "∃N. SA_poly_ubounded p m (UPSA.
   pderiv m f) c (condition_to_set 𝒞) N"
  assumes i_zero: "i = 0"
  assumes i_unique_min: "⋀j x. j ∈ inds ⟹ j > 0 ⟹ x ∈ b
    ⟹ val (H 0 x) < val (H j x)".
```

In the above we have also added the assumption that the desired boundedness property has already been established on the cell $\mathcal{C}$, which can be arranged for in the final proof of theorem I_{d+1} by induction. We now can prove that our desired decomposition of B can be performed separately in each of these locales. In `A₀_comp_refinement_i_zero` we prove that no further decomposition of the cell `B` defined in the axioms of `A₀_comp_refinement` is needed:

```
lemma ubounded: "SA_poly_ubounded p m f c (
   condition_to_set B) 0".
```

In A$_0$_comp_refinement_i_pos the proof is more involved, and requires us to invoke the tools from Cell_Normalization to obtain an appropriate decomposition of the underlying set of the cell B by obtaining one for the corresponding cell normal_cell m b. To access the necessary lemmas from the locale cell_normaliztion in the context of A$_0$_comp_refinement_i_pos, we can perform a locale interpretation:

```
cell_normalization p B a b c φ m f inds i
```

which binds the constants of A$_0$_comp_refinement_i_pos to corresponding constants of cell_normaliztion. We can then construct the desired decomposition of normal_cell m b (with some work, following the construction in Denef's paper):

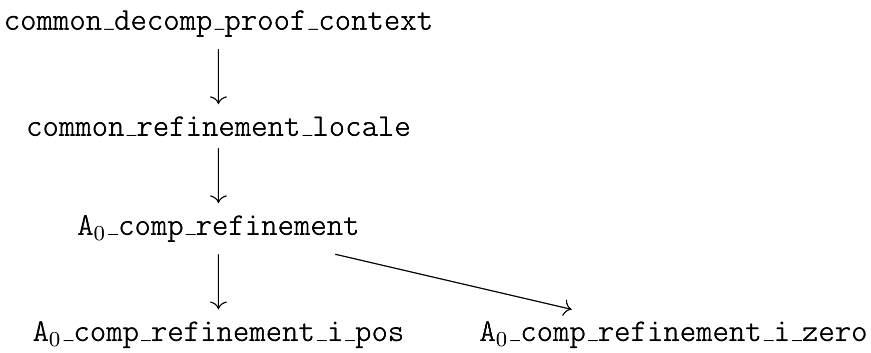
```
lemma normal_cell_final_decomp:
"∃ Cs. is_cell_decomp m Cs (condition_to_set (normal_cell
   m b)) ∧
         (∀C∈ Cs. (∃N. SA_poly_ubounded p m g (center C (
   condition_to_set C N ) )"
```

which we can then transfer to a cell decomposition of the cell B, by invocation of the inference rule transfer_decomp_ubounded which was stated in Section 11.3:

```
lemma B_final_decomp:
"∃ Cs. is_cell_decomp m Cs (condition_to_set B) ∧
         (∀C∈ Cs. ∃N. SA_poly_ubounded p m f (center C (
   condition_to_set C) N  )".
```

11.4.2 Pulling Results back to the General Proof Context

Up until now, we have only shown how to perform our desired decomposition on the underlying sets of certain cells which we have axiomatically assumed to satisfy certain properties in locales. To obtain the final desired decomposition of the actual set $\mathbb{Q}_p^{m+1}$, we can pull the conclusions of Theorem I_{d+1} backwards from the stronger locales to the weaker ones. We first provide a summary of the locale hierarchy up until this point for the purposes of proof:


common_decomp_proof_context
common_refinement_locale
A$_0$_comp_refinement
A$_0$_comp_refinement_i_pos
A$_0$_comp_refinement_i_zero

Following Denef, we would like to prove

```
theorem denef_I:
"denef_I p (Suc d)"
```

in the locale common_decomp_proof_context. We globally structure this proof by first defining a predicate for a set that can be decomposed as in the conclusion of the theorem:

```
definition denef_I_property where
"denef_I_property m f A = (∃ S. (is_cell_decomp m S A ∧
    (∀𝒞 ∈S. ∃N. SA_poly_ubounded p m f (center 𝒞) (
   condition_to_set 𝒞) N)))"
```

and prove a simple lemma that allows one to iteratively prove denef_I_property m f A for a set by successive cell decomposition

```
lemma denef_I_property_refine:
  assumes "is_cell_decomp m S A ∧ (∀ 𝒞 ∈S.
   denef_I_property m f (condition_to_set 𝒞))"
  shows "denef_I_property m f A"
```

and finally creating a simple lemma that can be invoked to conclude the theorem based on denef_I_property:

```
lemma denef_I_proof_by_property:
  assumes "⋀m f.[|f∈carrier (UP (SA m)); deg (SA m) f ≤ d
    |] ⟹
                  denef_I_property m f (carrier (Q_p^(Suc m))"
  shows "denef_I p d"
```

To prove the theorem, we can invoke the rule denef_I_proof_by_property, produce an initial cell decomposition of $\mathbb{Q}_p^{m+1}$ using denef_I d, and then iteratively refine this decomposition to satisfy the finer and finer locales, until we find ourselves working in a locale where we can directly prove `denef_property_I` on the current cell.

11.5 Proving the Base Case of Theorem II_{d+1}

The proof of Theorem II_{d+1} for the case that we only have one polynomial is structured very similarly to the proof Theorem I_{d+1}, successively refining a proof context until we have imposed sufficiently strong assumptions to prove the conclusions of the theorem, before lifting those results back up to the more general context. Since the proof will make use of Theorem I_{d+1}, we extend `common_refinement_locale` by adding the assumption that we are working with a cell where the conclusion of Theorem I_{d+1} is already satisfied, and we have already refined sufficiently so that there is a minimial index in the sense of Step 3. In addition, we add some assumptions that guarantee neither the polynomial f nor the cell $\mathcal{C}$ are trivial:

```
locale denef_II_base = common_refinement_locale +
  fixes n
  assumes n_pos: "(n::nat) > 0"
  assumes inds_nonempty: "inds ≠ {}"
  assumes ubounded: "∃N. SA_poly_ubounded p m f c (
   condition_to_set 𝒞) N"
  assumes min_taylor_term: "has_minimal_i 𝒞"
  assumes nontrivial: "condition_to_set 𝒞 ≠ {}".
```

The axiom `min_taylor_term` allows us to define a witness to this minimality, which we call i_0 (following the naming conventions of the source paper):

```
definition i₀ where i₀_def: "i₀ = (SOME i₀. (∀j. ∀t. ∀x. t#x
    ∈condition_to_set 𝒞
          ⟶ val ((a i₀ x)⊗(t ⊖ c x)[^]i₀) ≤ val ((a j x)
   ⊗(t ⊖ c x)[^]j)))"

lemma i₀_fact:
  "⋀ t x j. t#x ∈condition_to_set 𝒞 ⟹
                      val ((a i₀ x)⊗(t ⊖ c x)[^]i₀) ≤ val
   ((a j x)⊗(t ⊖ c x)[^]j)".
```

We can also use the choice operator to define a concrete parameter witnessing the axiom `ubounded`, which Denef calls κ and we call `N0`:

```
definition N0 where N0: "(N0::nat) = (SOME N. ∀x t i.
           t # x ∈ condition_to_set 𝒞 ⟶
```

```
            val (SA_poly_to_Qp_poly m x f · t)
            ≤ val (UPQ.taylor_term (c x) (
   SA_poly_to_Qp_poly m x f) i · t) + eint (int N))"
```

and we can do the same for the value λ which is described in our base case outline. Since λ is a special symbol in Isabelle, we call this value N1 instead:

```
definition N1 where N1: "N1 = (SOME (N::nat). N > 1 ∧
                                 (∀u∈carrier Q_p. ac N u =
   1 ∧ val u = 0 ⟶ u ∈ P_set n))"
```

and for later use, we also define a parameter which is larger than both of these, a value N which is defined to be N0 + N1 + 1.

The next step in the proof is to define the set which Denef calls A_0, and is described in Step 4 of our proof outline. Since this name is already used in a definition within common_refinement_locale, we give this set the name A_1 instead:

```
definition A_1 where A_1_def:
      "A_1 = {xs ∈condition_to_set 𝒞.
            (∀j ∈inds. j ≠ i_0 ⟶ val (a i_0 (tl xs) ⊗ ((hd
   xs) ⊖ c (tl xs))[^]i_0) + N ≤
                                      val (a j (tl xs) ⊗ ((hd
    xs) ⊖ c (tl xs))[^]j))}".
```

Without too much difficulty we can use this definition to show that the complement can be decomposed into cells with center c:

```
lemma A_1_comp_decomp:
  assumes "i_0 ∈inds"
  assumes "inds ≠ {i_0}"
  shows "∃S. is_cell_decomp m S (condition_to_set 𝒟- A_1)∧
                                        (∀𝒟∈S. center 𝒟 ∧
   boundary_condition 𝒟= closed_interval ∧
                                            (∃φ ∈Units (SA m)
   . u_bound 𝒟 =  φ ∧ l_bound 𝒟 = φ))"
```

which allows us to get a decomposition of A_1 as well, again using the lemma cell_decomp_same_center described in Section 8.10. This decomposition is already of the desired form in the conclusion of Theorem II_{d+1} without the need for further refinement, which completes the proof up to Step 6 of our outline:

```
lemma A_1_factored_decomp:
  assumes "i_0 ∈ inds"
  assumes "inds ≠ {i_0}"
  shows "∃S. is_cell_decomp m S A_1 ∧ (∀ 𝒟 ∈S. ∃ u h k.
   SA_poly_factors p m n f (center 𝒟) (condition_to_set 𝒟)
    u h k)".
```

The decomposition of the complement takes the majority of the work, and again will require cell normalization techniques as in the proof of Theorem I_{d+1}. Since we are interested in decomposing a single cell contained in the complement of $\mathtt{A}_1$, we can make a locale to reflect this context:

```
locale normal_cell_transformation =
  denef_II_base +
  fixes ℬ b φ H
  assumes subset: "condition_to_set ℬ ⊆ condition_to_set 𝒞
    - A_1"
  assumes φ_Unit: "φ ∈ Units (SA m)"
  assumes B_cell: "is_cell_condition ℬ"
  assumes B_eq: "ℬ = Cond m b c φ φ closed_interval"
  assumes b_nonempty: "b ≠ {}"
```

and then note that this interprets the locale `cell_normalization`

```
lemma cell_normalization_axioms_hold:
"cell_normalization p ℬ a b c φ m f inds i_0"

interpretation cell_normalization _ _ _ _ ℬ a b c φ _ _
   inds i_0
  using cell_normalization_axioms_hold by auto
```

to gain access to the lemmas and definitions of that locale which was described in Section 11.3. The polynomial g described in Section 11.3 is again the same one from Denef's proof of Theorem II_{d+1}. This proof follows Denef's closely. We can then perform the final decomposition needed in this locale:

```
lemma cell_decomp:
"∃S. is_cell_decomp m S (condition_to_set ℬ) ∧
      (∀ 𝒞 ∈ S. ∃ u h k. SA_poly_factors p m n f (center 𝒞
   ) (condition_to_set 𝒞) u h k)"
```

and transfer the results to the locale `denef_II_base` via locale interpretation and successive cell decomposition refinements:

```
lemma denef_II_base_cell_decomp:
```

```
"∃S. is_cell_decomp m S (condition_to_set 𝒞) ∧
     (∀ 𝒞 ∈ S. ∃u h k. SA_poly_factors p m n f (center 𝒞)
   (condition_to_set 𝒞) u h k)".
```

11.6 The Inductive Step of Theorem II_{d+1}

In the proof of the inductive step, a somewhat complicated setup is described. Denef's proof proceeeds as we described in Inductive Step 1. We work in the locale `common_decomp_proof_context`, which we recall consists of the amalgamation of the locales `denef_I` and `denef_II`. We need to define two predicates which are essential to the structure of our formalisation. First, we have a predicate `is_r_prepared`, which describes a set which is an intersection of cells of a certain kind:

```
definition is_r_prepared where
"is_r_prepared m n r Fs A  ≡
    finite Fs ∧

    Fs ⊆ carrier (UP (SA m)) ∧ (∃Cs 𝒞. 𝒞 ∈ Fs → Cs ∧ card
    (center ‘ 𝒞 ‘ Fs) ≤ r ∧

    A = (⋂f ∈ Fs. condition_to_set (𝒞 f)) ∧

    (∀ f ∈ Fs. is_cell_condition (𝒞 f) ∧ arity (𝒞 f) = m ∧
        (∃u h k. SA_poly_factors p m n f (center (𝒞 f)) (
   condition_to_set (𝒞 f)) u h k))) ".
```

We can break this definition down into simpler terms:

1. The set `Fs` is a finite set of semi-algebraic polynomials $f(t, x)$, with x being a variable with arity m.

2. We are asserting the existence of a set of cells `Cs`, and a function $\mathcal{C}$ which maps polynomials in `Fs` to cells in `Cs`. The image of this function is a set of cells with no more than `r` distinct centers.

3. The set `A` is the intersection of the cells in the image of the function $\mathcal{C}$.

4. For each `f` ∈ `Fs`, the polynomial `f` factors on the cell ($\mathcal{C}$ `f`) as in the conclusion of Theorem II_{d+1}.

We also need a predicate to describe a set which can be partitioned into sets of the above kind, which we call `is_r_preparable`,

```
definition is_r_preparable where
"is_r_preparable m n r Fs A = (∃Ps. finite Ps ∧ Ps
   partitions A ∧ (∀S ∈ Ps. is_r_prepared m n r Fs S ))".
```

As in the proof of Theorem II_{d+1}, we aim to show that $\mathbb{Q}_p^{m+1}$ is r-preparable with $r = 1$, for any finite set `Fs` of polynomials. From this we can infer Theorem II_{d+1} itself using the lemma `cell_decomp_same_center` which allows us to refine boolean combinations of cells with the same center into cell decompositions with the same center. Using the fact that we have proved the base case of Theorem II_{d+1}, we can prove that $\mathbb{Q}_p^{m+1}$ is r-preparable with $r = 1$, in the case that `Fs` is a singleton:

```
lemma is_1_preparable_singelton:
  assumes closed: "f ∈ carrier (UP (SA m))"
  assumes deg: "deg (SA m) f ≤ Suc d"
  assumes "n > 0"
  shows "is_r_preparable m n 1 {f} (carrier (Q_p^{Suc m})".
```

We can then easily show that if sets A, B can be r and k-prepared relative to sets Fs and Gs, respectively, then $A \cap B$ can be $r + k$ prepared relative to $Fs \cup Gs$:

```
lemma is_r_preparable_intersect:
  assumes "is_r_preparable m n r Fs A"
  assumes "Fs ≠ {}"
  assumes "is_r_preparable m n k Gs B"
  assumes "Gs ≠ {}"
  assumes "Fs ∩ Gs = {}"
  shows "is_r_preparable m n (r+k) (Fs ∪ Gs) (A ∩ B)".
```

These previous two basic lemmas are enough to show that $\mathbb{Q}_p^{m+1}$ is `r`-preparable relative to a set `Fs` of size `r`,

```
lemma Qp_is_r_preparable:
  assumes "n > 0"
  assumes "⋀f. f ∈ Fs ⟹ deg (SA m) f ≤ Suc d"
  assumes "Fs ⊆ carrier (UP (SA m))"
  assumes "finite Fs"
  assumes "Fs ≠ {}"
  shows "is_r_preparable m n (card Fs) Fs (carrier (Q_p^{Suc m})
   ".
```

From this, we only need to prove a lemma which shows that allows us to decrease the value of `r` in the previous lemma, provided that we known that $r \geq 2$:

```
lemma is_r_preparable_reduce:
  assumes "is_r_preparable m n (Suc (Suc r)) Fs A"
  assumes "⋀f. f ∈ Fs ⟹ deg (SA m) f ≤ Suc d"
  assumes "n > 0"
  shows "is_r_preparable m n (Suc r) Fs A".
```

The lemmas `Qp_is_r_preparable` and `is_r_preparable_reduce` can then be used in an inductive argument to show:

```
lemma Qp_is_1_preparable:
  assumes "n > 0"
  assumes "⋀f. f ∈ Fs ⟹ deg (SA m) f ≤ Suc d"
  assumes "Fs ⊆ carrier (UP (SA m))"
  assumes "finite Fs"
  assumes "Fs ≠ {}"
  shows "is_r_preparable m n 1 Fs (carrier (Q_p^Suc m))".
```

The bulk of the work in executing this proof strategy goes into showing the lemma `is_r_preparable_reduce`. As in Denef's proof, we can easily infer this result from the case of $r = 2$, so our work is focussed on this. In particular, we can infer the lemma `is_r_preparable_reduce` by induction from the following:

```
lemma is_2_prepared_reduce:
  assumes "fs ⊆ carrier (UP (SA m))"
  assumes "⋀f. f ∈ fs ⟹ deg (SA m) f ≤ Suc d"
  assumes "gs ⊆ carrier (UP (SA m))"
  assumes "⋀g. g ∈ gs ⟹ deg (SA m) g ≤ Suc d"
  assumes "n > 0"
  assumes "is_r_prepared m n 1 fs A"
  assumes "is_r_prepared m n 1 gs B"
  assumes "fs ∩ gs = {}"
  shows "is_r_preparable m n 1 (fs ∪ gs) (A ∩ B)".
```

Finally, once we have shown that $\mathbb{Q}_p^{m+1}$ is 1-preparable, we can express it as a disjoint union of intersections of cells, where each intersection of cells shares a common center. This means that the intersection can be re-expressed as the underlying set of a single cell, from which we can infer Theorem II_{d+1}:

```
lemma denef_cell_decomp_II_induct:
  shows "denef_II p (Suc d)".
```

11.7 Proving the Cell Decomposition Theorems for all Degrees

To prove cell decomposition theorems I and II unconditionally in the locale `padic_fields`, we do an induction on degree. We only need to prove a base case for Theorem II:

```
lemma denef_cell_decomp_II_base:
"denef_II p 0"
```

from which we can use induction to prove, for an arbitrary parameter d:

```
theorem denef_cell_decomp_I:
  "denef_I p d"

theorem denef_cell_decomp_II:
"denef_II p d".
```

Chapter 12

Proving Macintyre's Theorem

The final formalisation is for Macintyre's Theorem itself, and is contained within the theory file `Macintyre_Theorem.thy`. As in the proofs of the other theorems, the result is obtained iteratively by showing it for successively simpler sets. The main iterative tool is the lemma `macintyre_finite_union`, which states that a projection of a set is semi-algebraic if the set is a finite union of sets whose projections are semi-algebraic,

```
lemma macintyre_finite_union:
  assumes "⋀a. a ∈ A ⟹ is_semialgebraic m {x ∈ carrier
   (Q_p^m). (∃t ∈ carrier Q_p) (t#x) ∈ a)}"
  assumes "finite A"
  shows "is_semialgebraic m {x ∈ carrier (Q_p^m. (∃t ∈
   carrier Q_p (t#x) ∈ (⋃ A))}".
```

The proof uses 4 locales which successively refine one another to describe the generic construction. To get a reduction to a set as described in Step 2 of our proof sketch, we need to do a few things. First we can use the basic properties of generated Boolean algebras to decompose a semi-algebraic set S into a finite union of intersections of basic semi-algebraic sets and their complements. Our first locale reflects this:

```
locale macintyre_reduction_i = padic_fields +
  fixes Bs m
  assumes Bs_sub: "Bs ⊆ basic_semialgs (Suc m)"
  assumes Bs_finite: "finite Bs"
  assumes Bs_un: "carrier (Q_p^Suc m = ⋃ Bs"
  assumes Bs_nonempty: "Bs ≠ {}"
```

where the set Bs reflects the collection of sets in our intersection. For each b ∈ Bs we can use the choice operator to select a defining polynomial and exponent for this set:

```
definition F where F_def: "F = (λb. (SOME f. f ∈ carrier (
   Q_p^{Suc m} ∧ (∃(N::nat). N ≠ 0 ∧
                                                b = basic_semialg_set (
   Suc m) N f)))"
```

```
definition N where N_def: "N = (λb. (SOME N. N ≠ 0 ∧ b =
   basic_semialg_set (Suc m) N (F b)))"
```

and prove a lemma characterizing these functions:

```
lemma F_N_eval: "⋀ b. b ∈ Bs ⟹  b = basic_semialg_set (
   Suc m) (N b) (F b)".
```

In order to complete the obligations of this step, we must also apply theorem *iid* to obtain a cell decomposition of $\mathbb{Q}_p^{m+1}$, which will allow us to replace the polynomials from the definition of basic semi-algebraic sets with expressions of the form $h(x)(t - c(x))^\nu$ instead. This can be done because if we factor a polynomial $f(x,t)$ into the form

$$f(x,t) = u(x,t)^n h(x)(t - c(x))^\nu$$

then we know that f will be an n^{th} power iff $h(x)(t-c(x))^\nu$ is. Since the corresponding exponents N b for polynomials F b depend on the choice of b ∈ Bs, we must replace these with their least common multiple so that this replacement can be performed in a way compatible with each F b simultaneously. Assuming such a procedure has already been done, we can work in a finer locale now which also provides fixed constants for a background cell,

```
locale macintyre_reduction_ii = macintyre_reduction_i +
  fixes ν:: "padic_tuple set ⇒ nat"
  fixes  Xs 𝒞 h H
  assumes 𝒞_cell: "is_cell_condition 𝒞"
  assumes 𝒞_arity: "arity 𝒞 = m"
  assumes Xs_def: "Xs ⊆ Bs"
  assumes h_closed: "⋀ b. b ∈ Bs ⟹ h b ∈ carrier (SA m)
   "
  assumes H_Xs:
      "⋀b. b ∈ Xs ⟹
          H b = {xs ∈ carrier (Q_p^{Suc m}). (h b) (tl xs) ⊗ (hd
   xs ⊖ center 𝒞 tl xs)) [^] (ν b) ∈ P_pows (N b)}"
  assumes H_notXs:
      "⋀b. b ∉ Xs  ⟹
          H b = {xs ∈ carrier (Q_p^{Suc m}). (h b) (tl xs) ⊗ (hd
   xs ⊖ center𝒞 tl xs)) [^] (ν b) ∉ P_pows (N b)}".
```

In this locale the intention is that the sets `H b` will represent the conjuncts of the form "$h_i(x)(t-c(x))^{\nu_i}$ is (is not) an n_i-th power" which are alluded to in Step 2. The goal is to prove in this locale the following lemma:

```
  lemma reduction_ii:
  "is_semialgebraic m {x ∈ carrier (Q_p^{Suc m}). ∃t∈carrier Q_p.
   t # x ∈ (⋂x∈Bs. H x ∩ condition_to_set 𝒞)".
```

Steps 4 and 5 require showing that certain cell-like sets where the value group interval endpoints are possible rational numbers rather than integers are still semi-algebraic sets. In particular, we would like to prove that a set of the form

$$\left\{x \in \mathbb{Q}_p^m \mid \exists l \in \mathbb{Z} : \frac{\mathrm{ord}(a_1(x)\varrho^{-1})}{n} \leq l \leq \frac{\mathrm{ord}(a_2(x)\varrho^{-1})}{n}\right\}$$

is semi-algebraic, provided that a_1, a_2 are semi-algebraic functions, and $\varrho \in \mathbb{Q}_p$ is a unit. This is done in a dedicated locale which is specified below:

```
locale rational_cell_interval = padic_fields +
  fixes W and n and b1 and b2 and m and I
  assumes convex: "is_convex_condition I"
  assumes W_def: "W = {x ∈ carrier (Q_p^m). ∃l::int. l*n ∈ I
   (val (b1 x)) (val (b2 x))}"
  assumes n_pos: "(n::nat) > 0"
  assumes b1_closed: "b1 ∈ carrier (SA m)"
  assumes b2_closed: "b2 ∈ carrier (SA m)".
```

The main goal of this locale is then to prove that a set of the form

$$\{x \in \mathbb{Q}_p^m \mid \exists l \in \mathbb{Z} : \mathrm{ord}(b_1(x))\square_1 ln\square_2\mathrm{ord}(b_2(x))\} \tag{12.1}$$

is semi-algebraic. The proof of this lemma is facilitated by slightly sharpened versions of Denef's Lemma 2.4 (see 9.0.2). While Denef assumes the function $\xi(x)$ from this lemma always has a valuation which is a multiple of the modulus k, we can instead show that for any semi-algebraic unit (i.e. never takes nonzero values) $\xi(x)$, and any $k \geq 1$, there are semi-algebraic units $\eta_1(x), \eta_2(x)$ such that for all $x \in \mathbb{Q}_p^m$,

$$\mathrm{ord}(\eta_1(x)) = \frac{1}{k}(\mathrm{ord}(\xi(x) - (\mathrm{ord}(\xi(x) \mod k)))$$

$$\mathrm{ord}(\eta_2(x)) = \frac{1}{k}(\mathrm{ord}(\xi(x) + (k - \mathrm{ord}(\xi(x) \mod k))).$$

We can then define functions which map a semi-algebraic unit $\xi(x)$ and a modulus k to these η_1, η_2, which we name `floor_fun` and `ceiling_fun` respectively, and can then prove the following lemmas which will allow us to reduce inequality statements about a semi-algebraic unit φ`(x)` to those about `ceiling_fun` and `floor_fun`:

```
lemma ceiling_fun_equiv:
  assumes "x ∈ carrier (Q_p^m)"
  assumes "k ≥ 1"
  assumes "φ ∈ Units (SA m)"
  assumes "ord (φ x) mod int k ≠ 0"
  shows
    "(val (φ x) > j*int k) = (val (ceiling_fun m k φ x) >
   j)"
    "(val (φ x) ≥ j*k) = (val (ceiling_fun m k φ x) > j)"
```

```
lemma floor_fun_equiv:
  assumes "x ∈ carrier (Q_p^m)"
  assumes "k ≥ 1"
  assumes "φ ∈ Units (SA m)"
  assumes "ord (φ x) mod int k ≠ 0"
  shows
    "(val (φ x) < j* int k) =
        (val (floor_fun m k φ x) < j)"
    "(val (φ x) ≤ j* int k) =
        (val (floor_fun m k φ x) < j)".
```

The remainder of the work in this locale amounts to separating the possible values of the convex condition `I`, and separating edge cases where the boundary values `b_1, b_2` are or are not equal to 0 modulo `n`, which culminates in the lemma:

```
lemma W_semialg: "is_semialgebraic m W"
```

where `W` is specified in the locale definition. Finally, our project culminates with the statement and proof of the main result:

```
theorem macintyre_theorem:
  assumes "is_semialgebraic (Suc m) A"
  shows "is_semialgebraic m
    {x ∈ carrier (Q_p^m. (∃t ∈ carrier Q_p. (t#x) ∈ A)}".
```

Bibliography

[1] Avigad, J., and Harrison, J. Formally verified mathematics. *Commun. ACM 57*, 4 (apr 2014), 66–75.

[2] Ballarin, C. Tutorial to locales and locale interpretation. *Contribuciones científicas en honor de Mirian Andrés Gómez, 2010-01-01, ISBN 978-84-96487-50-5, pags. 123-140* (01 2010).

[3] Ballarin, C., et al. Session hol-algebra.

[4] Bordg, A. The localization of a commutative ring. *Archive of Formal Proofs* (June 2014). `https://www.isa-afp.org/entries/Localization_Ring.html`, Formal proof development.

[5] Cluckers, R. Presburger sets and p-minimal fields. *J. Symbolic Logic 68*, 1 (03 2003), 153–162.

[6] Cluckers, R., Comte, G., and Loeser, F. Lipschitz continuity properties for p-adic semi-algebraic and subanalytic functions. *Geometric and Functional Analysis 20* (06 2010), 68–87.

[7] Cluckers, R., Comte, G., and Loeser, F. Non-archimedean yomdin–gromov parametrizations and points of bounded height. *Forum of Mathematics, Pi 3* (2015), e5.

[8] Cluckers, R., Halupczok, I., and Rideau-Kikuchi, S. Hensel minimality i. *Forum of Mathematics, Pi 10* (2022), e11.

[9] Cluckers, R., and Leenknegt, E. A version of p-adic minimality. *The Journal of Symbolic Logic 77*, 2 (2012), 621–630.

[10] Cluckers, R., and Loeser, F. B-minimality. *J. Math. Log. 7* (2007).

[11] Cohen, P. J. Decision procedures for real and p-adic fields. *Communications on Pure and Applied Mathematics 22* (1969), 131–151.

[12] Conrad, K. Hensel's lemma. Accessed at the author's personal webpage: `https://kconrad.math.uconn.edu/blurbs/gradnumthy/hensel.pdf`.

[13] Crighton, A. Hensel's lemma for the p-adic integers. *Archive of Formal Proofs* (June 2014). `https://www.isa-afp.org/entries/Padic_Ints.html`, Formal proof development.

[14] Crighton, A. Hensel's lemma for the p-adic integers. *Archive of Formal Proofs* (Mar. 2021). `https://isa-afp.org/entries/Padic_Ints.html`, Formal proof development.

[15] Crighton, A. Macintyre's theorem in isabelle, 11 2022.

[16] de Vilhena, P. E. Hol/algebra/chinese_remainder.thy. Accessed at `https://isabelle.in.tum.de/library/HOL/HOL-Algebra/Chinese_Remainder.html`, Formal proof development.

[17] de Vilhena, P. E., and Paulson, L. C. Algebraically closed fields in isabelle/hol. In *Automated Reasoning* (Cham, 2020), N. Peltier and V. Sofronie-Stokkermans, Eds., Springer International Publishing, pp. 204–220.

[18] Denef, J. The rationality of the poincaré series associated to thep-adic points on a variety. *Inventiones mathematicae 77* (1984), 1–23.

[19] Denef, J. p-adic semi-algebraic sets and cell decomposition. *Journal für die reine und angewandte Mathematik 369* (1986), 154–166.

[20] Engler, A., and Prestel, A. *Valued Fields*. Springer Monographs in Mathematics. Springer Berlin Heidelberg, 2005.

[21] Haskell, D., and Macpherson, D. A version of o-minimality for the p-adics. *J. Symbolic Logic 62*, 4 (12 1997), 1075–1092.

[22] Hensel, K. Neue grundlagen der arithmetik. *Journal für die reine und angewandte Mathematik 127* (1904), 51–84.

[23] Kovacsics, P. C., and Nguyen, K. H. A p-minimal structure without definable skolem functions. *The Journal of Symbolic Logic 82*, 2 (2017), 778–786.

[24] Kuijpers, T., and Leenknegt, E. Differentiation in p-minimal structures and a p-adic local monotonicity theorem. *The Journal of Symbolic Logic 79*, 4 (2014), 1133–1147.

[25] Lewis, R. Y. A formal proof of hensel's lemma over the p-adic integers. In *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs* (New York, NY, USA, 2019), CPP 2019, Association for Computing Machinery, p. 15–26.

[26] MacIntyre, A. On definable subsets of p-adic fields. *The Journal of Symbolic Logic 41*, 3 (1976), 605–610.

[27] Mourgues, M.-H. Cell decomposition for p-minimal fields. *Mathematical Logic Quarterly 55*, 5 (2009), 487–492.

[28] Nipkow, T. Linear quantifier elimination. *Journal of Automated Reasoning 45*, 2 (July 2010), 189–212.

[29] Prestel, A., and Roquette, P. *Formally p-adic Fields*, vol. 1050 of *Lecture Notes in Mathematics.* Springer Berlin Heidelberg, Berlin, Heidelberg, 1984.

[30] Schikhof, W. H. *Ultrametric Calculus: An Introduction to p-Adic Analysis.* Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1985.

[31] Tarski, A. A decision method for elementary algebra and geometry. In *Quantifier Elimination and Cylindrical Algebraic Decomposition* (Vienna, 1998), B. F. Caviness and J. R. Johnson, Eds., Springer Vienna, pp. 24–84.

[32] von Oheimb, D., and Haftmann, F. Hol/library/extended_nat.thy. Accessed at `https://isabelle.in.tum.de/library/HOL/HOL-Library/Extended_Nat.html`.

[33] Wenzel, M. The isabelle/isar reference manual. Accessed at `https://isabelle.in.tum.de/dist/Isabelle2021-1/doc/isar-ref.pdf`.