Writing a grammar and Parser for Blazon

Thomas Heyes

April 16, 2013

Abstract

Blazon is the semi-formal language of family crest and heraldry, dating back to the twelfth century. Using a well structured grammar to describe a coat of arms in a top down approach the language provides a robust yet flexible way to define a textual description of what is naturally a very graphical subject. Using a test-driven development model I have produced a project which is capable of parsing a large subset of Blazon. Encompassing several fields of Computer Science ranging from parsing to HTML5 graphics the application provides a platform that demonstrates how modern concepts and technologies can be used to represent a subject that pre-dates them by centuries.

Firstly giving a brief background and description of the language of Blazon, this report goes onto describes both how the Blazon parsing application works as well as and how the project was implemented and tested. From a simple shield of a single colour through partitioning into sections and sub partitioning as well as covering different line-types before heading onto geometric charges, honourable charges and the rule of tincture before finally discussing semi-formal charges.

After thoroughly describing the language I go onto discuss the implementation of the project using test driven development. Producing lots of iterations increasing the functionality of the project gradually and performing regression testing to ensure the soundness of my code base. Initially starting with a couple of Python script based prototypes I describe how the project was iteratively built into a fully fledged web application over the course of the academic year. Given more time I would attempt to expand the project further into parsing and drawing a larger subset of Blazon with features such as Counter Charging and combining shields with Quartering.

Contents

Chapter 1

Introduction

My project was to *Write a Grammar and Parser for Blazon.* Another student project by *Luke Torjussen* done the year before my own had attempted, very successfully, to produce an application that both parsed and drew representations of Blazoned coats of arms. This prior project used a parse generator to handle the grammar, lexing and parsing of the language. It was my objective to see if performing the parsing by hand in my own application could realise a more complete representation of Blazon.

Initially then my project was entirely based in tokenising, lexing, parsing and validating Blazon sentences. However my project grew later-on to also attempt to encompass producing a graphical representation of a parsed Blazon sentence as well as providing a plain English translation.

1.1 Aims

The aim of my project was to produce an application which upon receiving a Blazon sentence would firstly validate the sentence according to the rules of the Blazon language and then produce an English translation.

1.2 Achievable

1.3 Limitations

Chapter 2

What is Blazon?

Blazon is the language of heraldry and family crests, it dates back to the twelfth century and provides a strict set of rules about how to produce a coat of arms. There are several different versions of Blazon found around Europe however all follow the same behavioural rules regarding tinctures and charges. Most versions differ only in the set of tinctures and honourable ordinaries either having a more generous or conservative view on what is and isn't acceptable for example, African Blazon allows for an Orange coloured Tincture whilst English Blazon does not. For this project I focused on exclusively on a strict English Blazon.

Blazon is a powerful language for allowing limitless combinations and configurations of patterns and shapes to be *Blazoned* onto a shield in and recorded in a concise textual description.

What is impressive about the language is that it achieves this flexibility whilst reaming fairly formal and well defined, restricting the set of tinctures to total seven and maintaining a fairly low number of pre-defined honourable ordinaries.

Blazon, like any other language, has some unique terminology used to address certain aspects of heraldry. Anyone attempting utilize the language will need to familiarise themselves with this terminology.

2.1 The Seven Tinctures

The most fundamental elements of Blazon are the Tinctures. Tinctures are the set of colours allowed in coats of arms. English Blazon defines a set of seven tinctures and places them into to groups metals and non-metals.

Although at first this may seem overly restrictive Blazon overcomes having a limited set of valid tinctures by leaving interpretation of the tone and

The Seven Tinctures		
Tincture	Colour	Metal
Azure	Blue	Non-metal
Argent	Silver	Metal
Gules	Red	Non-metal
Or	Gold	Metal
Purpure	Purple	Non-metal
Sable	Black	Non-metal
Vert	Green	Non-metal

Table 2.1: Table of Tinctures found in English Blazon, the corresponding colours and whether each tincture is or isn't a metal.

shade of each Tincture's corresponding colour to the artist producing the shield. An Azure, blue, shield could be a light sky blue or a dark navy shade of blue depending upon the artists preference, as long as it is recognisably blue.

The Freedom of Interpretation is a great strength of Blazon vastly increases the different graphical variations of Blazoned shields without radically increasing the size of the language by exhaustively listing a set of valid tones and shades.

With regard to the two metallic Tinctures Or, gold, and Argent, silver, having freedom of interpretation allows for matt, non-metallic, colours to be placed onto a shield, although they will still behave as metal Tinctures. It is not uncommon to see yellow instead of gold and white instead of silver on many coats of arms.

2.2 Furs

Blazon incorporates several furs commonly used in coats of arms into the language. These behave in much the same way as tinctures but don't have the same metal or non-metal property.

There are several pre-defined patterns for furs that are used directly instead of actual fur on a shield. These patters generally consist of a repeating pattern on-top of a background colour. Each fur has a unique pattern although several are very similar, some simply being the inverse of another.

The colours of Furs are pre-defined but can be explicitly stated as differing from the normal colours by stating the Tinctures to be used instead. For example for example Ermine coloured silver and black would be defined as *"Ermine Argent and Sable"*

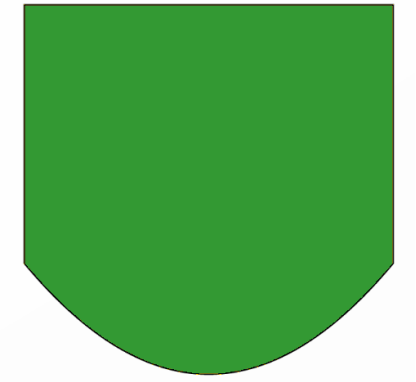
2.3 Fields

A Field is simply an area on a shield. Initially a shield has one implicit Field, which is the entire area of that shield. Blazon allows for two operations on fields, a field can be Tinctured, with a Tincture or a Fur, or a field can be Partitioned. Tincturing a Field dictates that the whole area incorporated in that Field be filled in the colour of that Tincture. Partitioning I will cover in much more detail later on.

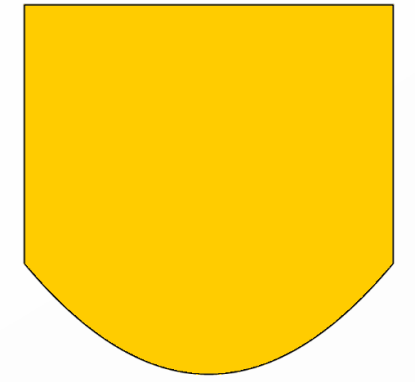
2.4 How to Blazon a Coat of Arms: Part One

I have now defined a large enough subset of Blazon for a couple of basic examples.

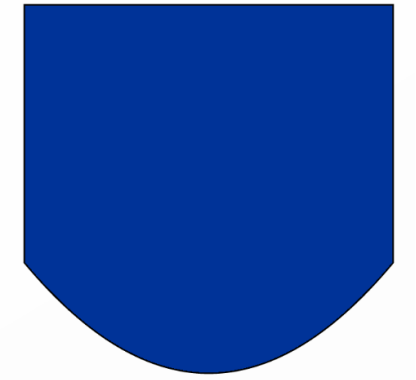
In each example we are implicitly provided with a single Field which encompasses the entire body of the shield. Then we Tincture that Field by providing a Tincture and we have successfully Blazoned a shield.



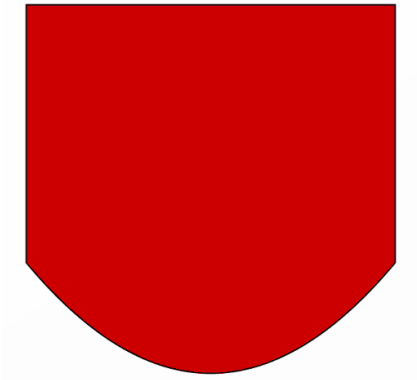
(a) *"Vert"*



(b) *"Or"*



(c) *"Azure"*



(d) *"Gules"*

Figure 2.1: Basic Blazon examples.

Although the examples in Figure 2.1 are very basic each is a complete valid Blazoning of a shield.

2.5 Partitioning a Field

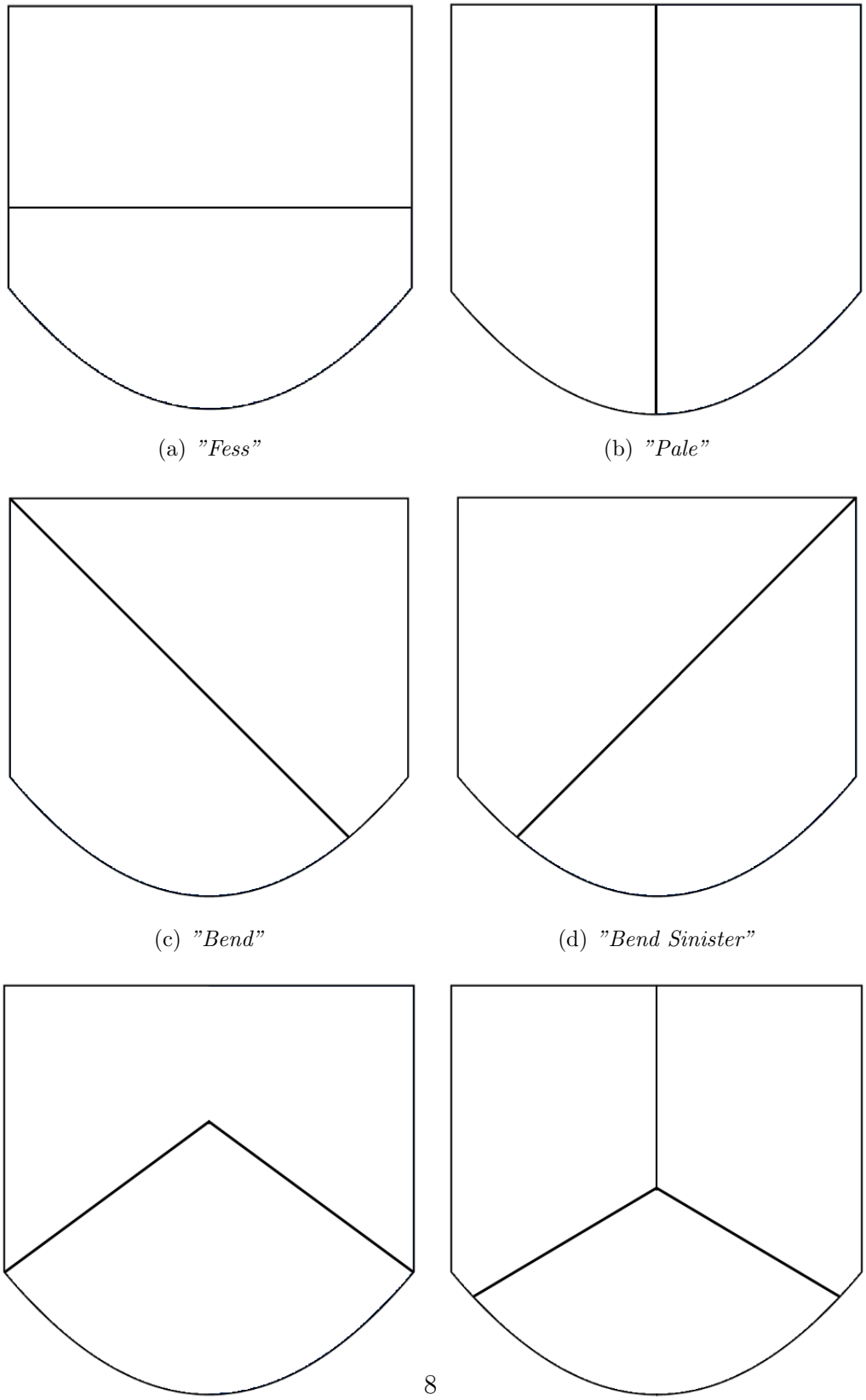
The Blazon I have defined so far is very limited, only allowing for single Tincture shields. The next natural area of the language to define is the operation of Partitioning a Field.

As stated previously a Field can be either Tinctured or Partitioned. To Partition a Field the key-word *"Per"* is used. It is obligatory that the word immediately after *"Per"* is a type of partition. Blazon has several pre-defined partitions ranging from the very simple *"Fess"* which divides a Field horizontally in half.

Partitioning a Field divides it up into several smaller fields the number and shape of which depend on the type of Partition used. After the Partition has been stated the Blazon sentence must go on to address the resulting new Fields.

A shield is Implicitly Blazoned from top left to bottom right with the top most Field taking priority firstly and if two or more Fields are adjacent at the same height the left most takes priority.

The Blazon sentence is only complete when all the Fields have been Tinctured. If a Blazon sentence Partitions a shield into two Fields and then provides only one Tincture that sentence is Invalid.



(a) *"Fess"*

(b) *"Pale"*

(c) *"Bend"*

(d) *"Bend Sinister"*

(e) *"Cheveron"*

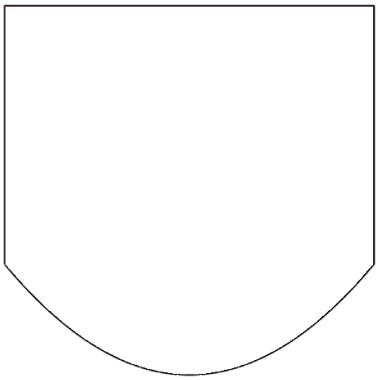
(f) *"Pall"*

Figure 2.2: *"Valid Partitions of a Field"*

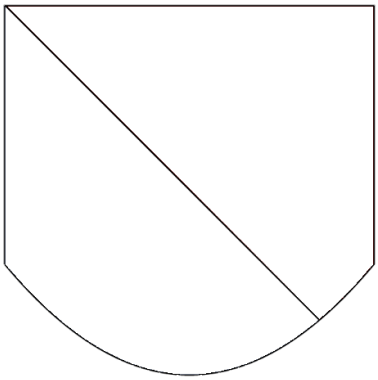
2.6 How to Blazon a Coat of Arms: Part Two

Partitioning is a very powerful aspect of Blazon and increases the number of possible shield designs immensely. A lot of very striking designs can be Blazoned onto a shield with very short Blazon sentences making use of Partitioning.

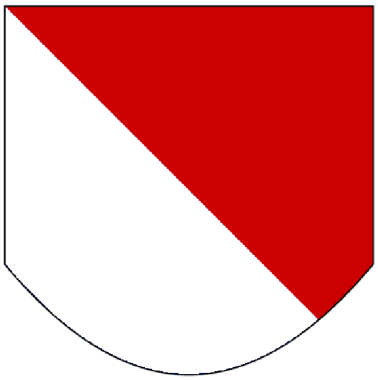
The Blazon sentence,*"Per Bend Gules and Azure"* Implicitly starts with a single Field which encompasses the entire body of the shield before partitioning that Field into two smaller Fields with the keyword *"Per"* declaring a partition followed by the word *"Bend"* which is a diagonal division of a field from top left to bottom right. The Blazon goes onto Tincture the two new fields with the two Tinctures *"Gules"* and *"Azure"* respectively, following Blazon's rule about evaluating the topmost and then leftmost field first the upper section of the shield is Tinctured *"Gules"* and then the lower half is Tinctured *"Azure"*. There is no more Blazon reaming in the sentence and there are also no empty fields upon the shield, therefore this is a valid Blazon sentence and a striking red and blue shield has been produced.



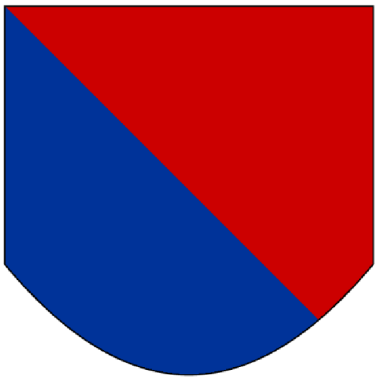
(a) *"A single empty Field."*



(b) *"The Field has been partitioned Per Bend"*



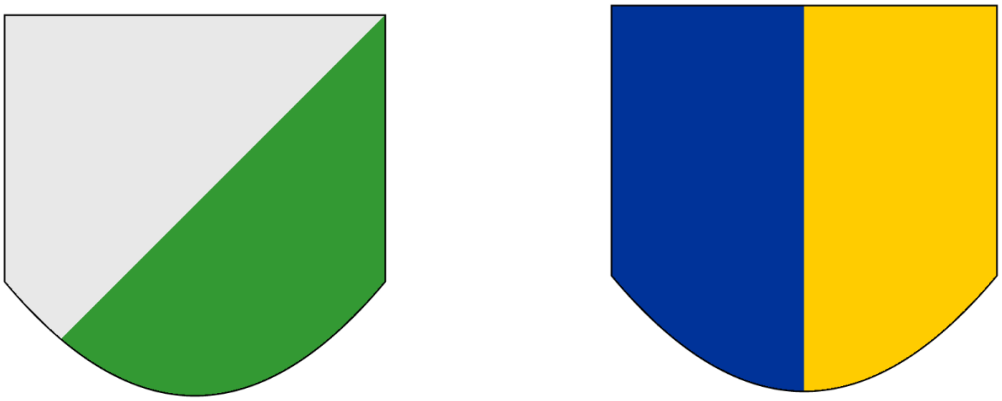
(c) *"The Topmost Field is Tinctured Gules"*



(d) *"The final Field is Tinctured Azure"*

Figure 2.3: *"Per Bend Gules and Azure"*.

Applying the same method as show above it is possible to validate the following Blazon sentences of similar complexity.



(a) *"Per Bend Sinister Argent and Vert."*

(b) *"Per Pale Azure and Or"*

Figure 2.4: *"Two more examples of valid Blazon sentences".*

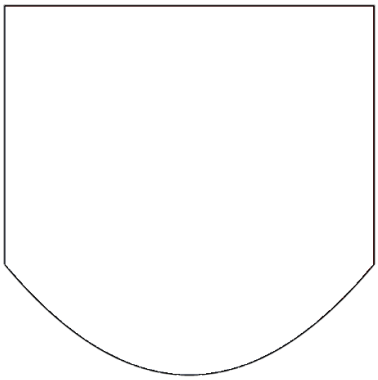
2.7 Sub-Partitioning

The process of Partitioning a field produces two or more smaller Fields. This allows for some very striking, though simple, designs. Blazon allows however for fields to be Sub-Partitioned. The new Fields produced after a Partitioning can themselves be Partitioned.

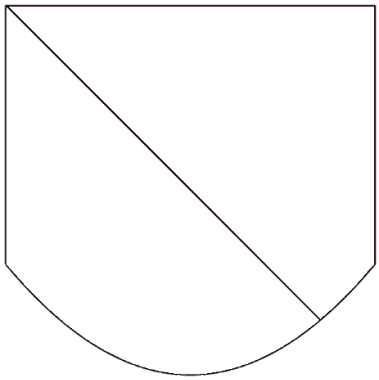
Considerably more advanced patterns can be achieved buy using Sub-Partitioning. The complexity is potentially infinite as there is no limit enforced by the language as to how many Partitions are allowed. It is rare to see a historical shield Sub-Partitioned more than three times. Sub-Partitioning makes use of exactly the same set of Partition types as regular Partitioning.

Sub-Partitioning is expressed in exactly the same way as partitioning. Once the initial field has been partitioned instead of providing a Tincture for one of the new fields the keyword *"Per"* is used to define a partition and then the type of partition is stated for example *"Fess"*. Then each of the new Fields need to be addressed either by being Tinctured or further partitioned.

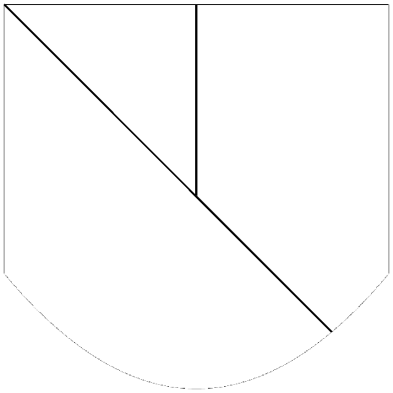
The new Fields created by the Partition fully evaluated before any other Fields on the shield are addressed. The new sub-fields are addressed according to the same rules present in regular Partitioning, namely the top most Fields take Priority then the left most, once they have all been evaluated the next field according to partitioning rules is addressed.



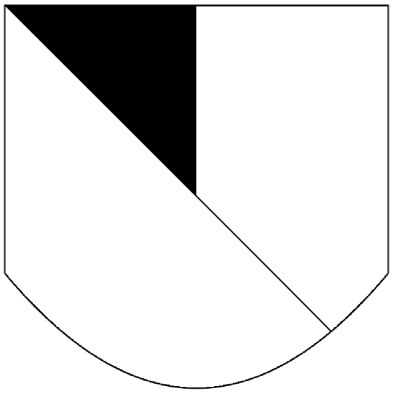
(a) *"A single empty Field."*



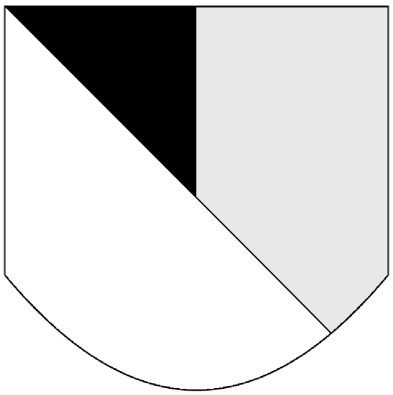
(b) *"The Field has been partitioned Per Bend"*



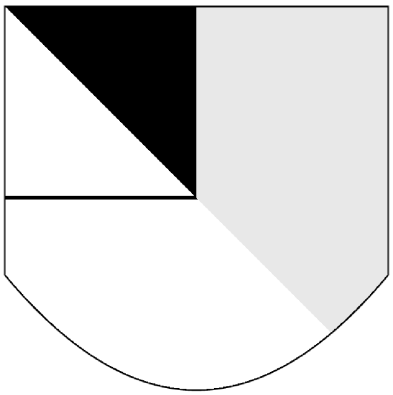
(c) *"The Topmost Field has then been Sub partitioned Pale"*



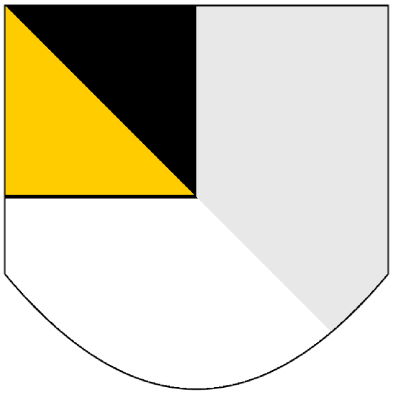
(d) *"The Top Left most Sub-Field is Tinctured Sable"*



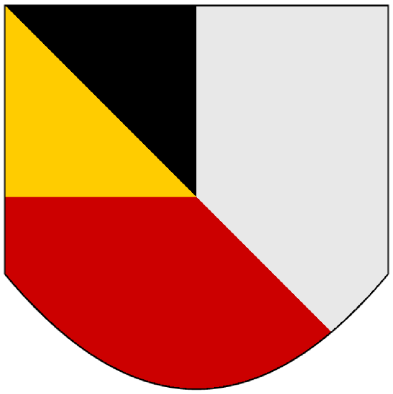
(e) *"The remaining Field is Sub-Partitioned Fess"*



(f) *"The Top Left Most Sub-Field is Tinctured Or"*



(g) *"The remaining Sub-Field is Sub-Partitioned Fess"*



(h) *"The final Field is Tinctured Gules"*

Figure 2.4: *"Per Bend Per Pale Sable and Argent Per Fess Or and Gules"*.

(a) *"A more complicated."*

Figure 2.5: *" "*.

2.8 Line Types

2.9 Directions and Sides

2.10 Charges

2.11 The Rule of Tincture

Chapter 3

Defining a Grammar for Blazon

3.1 What is a grammar?

3.2 Context Free Blazon

3.3 Formatting Input

3.4 Tokenizing Strings

Chapter 4

Parsing

Chapter 5

Graphical Representations

Chapter 6

Design