Chapter 5
Covering set correctibility from the perspective of logic: categorical edits

5.1 Introduction

In the previous chapter we formalised edit generation functions as logical deduction functions. In particular, we formalised the edit generation functions FH and MFH as the deduction functions $F$ and $MF$ respectively. We found that $MF$ is essentially the same as $MR_{Th}$.

However, formalising edit generation functions does not formalise the whole of the covering set method, which depends on aspects other than edit generation. Most importantly, the covering set method depends on the property of covering set correctibility.

If the formalisation of the covering set method is to be used for automation, then it is likely to be more powerful if the full method, including the property of covering set correctibility, is formalised in terms of logic. Otherwise, if only the edit generation functions are formalised, then any method based on logic will merely replicate the methods using sets. In this chapter I present a logical formalisation of the property of covering set correctibility. Summaries of the main results presented here were published in my previous papers (Boskovitz and Goré (2005) and Boskovitz, Goré, and Wong (2005)).

The property of covering set correctibility can be formalised by directly translating from the language of sets to the language of logic, using the formalisation function $nc$ and the formalisation of each record $v$ as the function $f_v$. I present a formalisation via such a direct translation in Section 5.2. However, such a direct formalisation is not in terms of any constructs of logic that might be of use in any automation.

We can, however, approach the formalisation in a different way. We will consider the meaning of covering set correctibility via the two properties it links together, namely (a) that of yielding a correction, and (b) that of being a covering set of failed clauses. We will find in Section 5.3 that property (a) is a type of Th-satisfiability, while property (b) can be expressed in terms of whether the empty clause is in a particular clause set. But refutation completeness and soundness also link satisfiability with whether the
empty clause is in a certain set. We will see in Proposition 5.3.7 that covering set correctibility turns out to be an extension of refutation completeness and soundness.

Indeed refutation completeness turns out to be related to one direction of covering set correctibility, namely the error correction guarantee; while soundness turns out to be related to the other direction of covering set correctibility, namely error correction totality. Section 5.4 will elucidate exactly how the error correction guarantee and error correction totality are related to Th-completeness and Th-soundness. In effect the error correction guarantee and error correction totality can be seen as new types of logical completeness and soundness respectively.

We return in Section 5.5 to the original problem addressed by covering set correctibility, namely the problem of error localisation. Covering set correctibility is related to the error localisation problem just as soundness and completeness are related to the propositional satisfiability problem. The relationship extends beyond the problem statements themselves to the underlying properties in their solutions.

The results of this chapter have potential for possible automated implementations of the covering set method. The implementations could be based on the automated logical implementations for the propositional satisfiability problem but modified to take account of the covering set correctibility property.

### 5.2 Direct logical formalisation of covering set correctibility

Chapter 4 gave a logical formalisation for normal edits (as normal clauses), records (as Th-truth functions), fields (as the subscripts of propositional atoms), and edit generation functions (as deduction functions). In this section we will find a formalisation of the remaining aspect of the covering set method, namely of covering set correctibility.

In order to formalise covering set correctibility we will first formalise the concepts used in its definition. Therefore the section starts with a formalisation of the concepts of error localisation, covering sets, and involved fields. We will then formalise covering set correctibility, and its two directions, namely error correction guarantee and error correction totality.

We will use the same name to each of the formalised concepts as to the original concept. That is, within logic the formalised concepts will also be called error localisation, covering sets, involved fields, covering set correctibility, error correction guarantee, and error correction totality. We will also use the same symbols. The ambiguity should not cause any confusion, because the context will be clear.

After defining all the formalised concepts, we will conclude this section with a lemma that tells us that the formalisations are consistent with each other. We will also be able to show that the deduction functions $\mathcal{F}$ and $\mathcal{M}\mathcal{F}$ are covering set correctible.

We start with the problem that covering set correctibility addresses, namely error localisation. The following definition is a translation to logical formulae of Definition 2.2.8 for error localisation using edits.
Definition 5.2.1. Let $v = (v_1, \ldots, v_N)$ be a record and $\Sigma$ be a set of logical formulae. (In general, we will require $\Sigma$ to be a set of normal clauses.) The error localisation problem is the problem of deciding which sets of fields can be changed to correct $f_v$ with respect to $\Sigma$. That is, it is the following problem: given a set $C$ of fields and a record $v$, decide whether there is a record $w = (w_1, \ldots, w_N)$ such that

1. $C \supseteq \{ j \mid w_j \neq v_j \}$, and
2. $f_w$ satisfies $\Sigma$.

If there exists a record $w$ such that a and b are satisfied, then we say that the set $C$ yields a correction $w$ to $v$ with respect to $\Sigma$ or that the set $C$ yields a correction $f_w$ to $f_v$ with respect to $\Sigma$ and we shall also say that that $C$ is a solution to the error localisation problem for $v$ (or $f_v$) and $\Sigma$.

The set of solutions to the error localisation problem for the record $v$ and the set $\Sigma$ of normal clauses will be written $EL(\Sigma, v)$. This means that $C$ yields a correction to $v$ with respect to $\Sigma$ if and only if $C \in EL(\Sigma, v)$.

Note 1: $f_v$ satisfies $\Sigma$ if and only if $\emptyset \in EL(\Sigma, v)$.

Note 2: If $\Box \in \Sigma$ then $EL(\Sigma, v) = \emptyset$.

Before we can give a logical formalisation of covering set correctibility, we must formalise the concept used in its definition, namely covering sets. Since covering sets are defined in terms of involved fields, we first give a logical formalisation of the term “involved”.

Definition 5.2.2. The field $j$ is involved in the normal clause $\sigma$ when $\sigma$ contains a propositional atom of form $p^v_j$, where $v \in A_j$.

Note: Every field is involved in $\top \top$.

We will show below, in Lemma 5.2.7, that the above definition is consistent with the definition of “involved” for edits, that is, that the field $j$ is involved in the edit $e$ if and only if it is involved in its logical formalisation, namely the clause $nc(e)$.

We use the term “involved” to give the logical formalisation of covering sets, in the next definition. We define covering sets in terms of involved fields in the same way as we did for edits in Definition 2.3.3.

Definition 5.2.3. A covering set $C$ of the set $\Sigma$ of normal clauses is a set of fields such that, for each clause $\sigma$ in $\Sigma$, there exists a field in $C$ that is involved in $\sigma$. The set of all covering sets of $\Sigma$ is written $C(\Sigma)$.

Note 1: $\Sigma = \emptyset$ if and only if $\emptyset \in C(\Sigma)$.

Note 2: $\Box \in \Sigma$ if and only if $C(\Sigma) = \emptyset$.

In order to give the logical formalisation of covering set correctibility, we will use the same notation, $X$, as we used for edits, for the set of clauses failed by a Th-truth function, as per the next definition, which follows the definition used for edits (Definition 2.3.5).
Definition 5.2.4. Define \( X(\Sigma, v) \) to be the set of clauses in the clause set \( \Sigma \) that are failed by \( f_v \). That is, \( X(\Sigma, v) = \{ \sigma \in \Sigma \mid f_v(\sigma) = \text{false} \} \).

We will also use the same notation, \( CX \), as we used for edits (Definition 2.3.6), for the set of covering sets of failed clauses, as follows.

Definition 5.2.5. Define \( CX(\Sigma, v) \) to be the set of covering sets of the clauses of \( \Sigma \) failed by \( f_v \). That is, \( CX(\Sigma, v) = C(X(\Sigma, v)) \).

We can now give a logical formalisation of covering set correctibility, and its two directions, error correction totality and error correction guarantee. Once again we follow the definitions used for edits (Definitions 2.3.14 and 2.3.16).

Definition 5.2.6. The deduction function \( D \) has **error correction totality** if for any set \( \Sigma \) of normal clauses and any record \( v \),

\[ \mathcal{E}L(\Sigma, v) \subseteq CX(D(\Sigma), v). \]

The deduction function \( D \) has the **error correction guarantee** if for any set \( \Sigma \) of normal clauses and any record \( v \),

\[ CX(D(\Sigma), v) \subseteq \mathcal{E}L(\Sigma, v). \]

The deduction function \( D \) is **covering set correctible** if for any set \( \Sigma \) of normal clauses and any record \( v \),

\[ CX(D(\Sigma), v) = \mathcal{E}L(\Sigma, v). \]

The next lemma confirms that the above formalisations are consistent with the concepts that they formalise and with the formalisation function \( nc \) introduced in Chapter 4, Section 4.2.5.

Lemma 5.2.7.

1. If \( e \) is a normal edit, then the field \( j \) is an involved field of the normal clause \( nc(e) \) if and only if \( j \) is an involved field of \( e \).

2. If \( E \) is a set of normal edits, then the field set \( C \) is a covering set of \( nc(E) \) if and only if \( C \) is a covering set of \( E \). That is, \( C(nc(E)) = C(E) \).

3. Let \( v \) be a record and let \( E \) be a set of normal edits. Then \( X(nc(E), v) = X(E, v) \).

4. Let \( v \) be a record and let \( E \) be a set of normal edits. Then the field set \( C \) yields a correction to \( f_v \) with respect to the set \( nc(E) \) if and only if the set \( C \) yields a correction to \( v \) with respect to the set \( E \). That is, \( \mathcal{E}L(nc(E), v) = \mathcal{E}L(E, v) \).

5. The deduction function \( D \) has error correction totality if and only if the edit generation function \( nc^{-1} \circ D \circ nc \) has error correction totality.

6. The deduction function \( D \) has the error correction guarantee if and only if the edit generation function \( nc^{-1} \circ D \circ nc \) has the error correction guarantee.
7. The deduction function $D$ is covering set correctible if and only if the edit generation function $\text{nc}^{-1} \circ D \circ \text{nc}$ is covering set correctible.

Proof.

1. By definition, the statement that $j$ is an involved field of the clause $\text{nc}(e)$ means that there is a field value $v$ in $A_j$ such that $p^v_j \subseteq \text{nc}(e)$. Also, by the definition of $\text{nc}$ (Definition 4.2.26),

$$p^v_j = \text{nc}(n_v),$$

where $n_v$ is the normal edit $A_1 \times \cdots \times A_{j-1} \times (A_j \setminus \{v\}) \times A_{j+1} \times \cdots \times A_N$ (and this way of writing $n_v$ is in normal form because $A_j$ has more than one element).

Hence the statement that $j$ is an involved field of the clause $\text{nc}(e)$ is equivalent to saying that there is a field value $v$ in $A_j$ such that $\text{nc}(n_v) \subseteq \text{nc}(e)$. The required result then follows from the following sequence of equivalent statements:

$$\exists v \in A_j \text{ nc}(n_v) \subseteq \text{nc}(e) \iff \exists v \in A_j \text{ nc}(n_v) \models_{\text{Th}} \text{nc}(e), \quad \text{by Lemma 4.2.30}$$

$$\iff \exists v \in A_j \text{ ec}(n_v) \models_{\text{Th}} \text{ec}(e), \quad \text{since for each normal edit } x, \text{ the clause } \text{nc}(x) \text{ is in the equivalence class } \text{ec}(x), \text{ by Proposition 4.2.19 and Definition 4.2.26}$$

$$\iff \exists v \in A_j \text{ } n_v \supseteq e, \quad \text{since } \text{ec} \text{ is a Boolean isomorphism, by Lemma / Definition 4.2.15}$$

$$\iff \exists v \in A_j \text{ } A_j \setminus \{v\} \supseteq A^j, \quad \text{by the definition of } n_v, \text{ and Lemma 2.2.15}$$

$$\iff \text{the field } j \text{ is involved in the edit } e.$$

2. Direct consequence of part 1 of this lemma, since covering sets are defined in terms of involved fields.

3. Suppose $e \in E$. The result follows from the property that $\text{nc}$ preserves record correctness and incorrectness (Proposition 4.2.28); that is: $\text{nc}(e) \in \mathcal{X}(\text{nc}(E), v) \iff f_w(\text{nc}(e)) = \text{false} \iff v \text{ fails } e \iff e \in \mathcal{X}(E, v)$.

4. By definition, $C \in \mathcal{EL}(\text{nc}(E), v)$ if and only if there is a record $w$ that differs from $v$ only on $C$ such that $f_w$ satisfies $\text{nc}(E)$. But by Proposition 4.2.28, $f_w$ satisfies $\text{nc}(E)$ if and only if $w$ satisfies $E$. Thus $C \in \mathcal{EL}(\text{nc}(E), v)$ if and only if there is a record $w$ that differs from $v$ only on $C$ such that $w$ satisfies $E$, that is $C \in \mathcal{EL}(E, v)$.

5. Let $v$ be a record, and let $\Sigma$ be a set of normal clauses. Since $\text{nc}$ is a bijection (by Lemma 4.2.27), let $E$ be the unique edit set such that $\Sigma = \text{nc}(E)$. The error correction totality of $D$ means that for each record $v$ and for each normal clause set $\Sigma$,

$$\mathcal{EL}(\Sigma, v) \subseteq \mathcal{CX}(D(\Sigma), v).$$
We will find expressions for each side of this equation in terms of normal edits rather than in terms of normal clauses. Firstly, the left-hand side:

$$\mathcal{EL}(\Sigma, v) = \mathcal{EL}(E, v),$$

by part 4 of this lemma.

Secondly, the right-hand side:

$$\mathcal{CX}(D(\Sigma), v) = \mathcal{CX}(\text{nc}^{-1} \circ D(\Sigma), v),$$

by part 3 of this lemma, since nc is a bijection

$$= \mathcal{CX}(\text{nc}^{-1} \circ D \circ \text{nc}(E), v),$$

by the definition of E.

Hence since nc is a bijection, error correction totality for clauses is equivalent to saying that for each record v and for each set E of normal edits,

$$\mathcal{EL}(E, v) \subseteq \mathcal{CX}(\text{nc}^{-1} \circ D \circ \text{nc}(E), v).$$

This is the error correction totality of nc$^{-1}$ $\circ$ D $\circ$ nc, as required.

6. The proof is identical to the proof of part 5, except with the subset relation reversed.

7. Follows from parts 5 and 6.

An immediate corollary is that the deduction functions $\mathcal{F}$, $\mathcal{MF}$ and $\mathcal{FCF}_\omega$ (for $\omega$ an ordering of the fields) are covering set correctible, just as are the edit generation functions $\mathcal{FH}$, $\mathcal{MFH}$ and $\mathcal{FCF}_\omega$ that they formalise.

**Corollary 5.2.8.** The deduction functions $\mathcal{F}$, $\mathcal{MF}$ and $\mathcal{FCF}_\omega$ (where $\omega$ is an ordering of the fields) are covering set correctible.

**Proof.** By Chapter 4, Definition 4.2.33, we have that $\mathcal{FH} = \text{nc}^{-1} \circ \mathcal{F} \circ \text{nc}$, that $\mathcal{MFH} = \text{nc}^{-1} \circ \mathcal{MF} \circ \text{nc}$, and that $\mathcal{FCF}_\omega = \text{nc}^{-1} \circ \mathcal{FCF}_\omega \circ \text{nc}$. But $\mathcal{FH}$ is covering set correctible (by Theorem 2.4.6 and Proposition 2.4.9), and $\mathcal{MFH}$ is covering set correctible (by Corollary 2.5.9 and Corollary 2.5.14). The function $\mathcal{FCF}_\omega$ is also covering set correctible, although we will not prove this until Chapter 6 (Lemma 6.3.1 and Proposition 6.8.5). Hence by part 7 of Lemma 5.2.7, we have that $\mathcal{F}$, $\mathcal{MF}$ and $\mathcal{FCF}_\omega$ are covering set correctible.

In this section, we have formalised covering set correctibility and the main concepts used in its construction. We have also shown that the formalisation is consistent with the definitions used for edits. The formalisation is in terms of new logical constructs, namely $\mathcal{EL}$, C, $\mathcal{X}$, and involved fields, which are obtained via a direct translation from sets. It would be useful to instead have a formalisation in terms of some natural constructs of logic. This is the topic of the next section.
5.3 Covering set correctibility in terms of some natural constructs of logic

In this section we will find a logical expression for covering set correctibility in terms of some constructs that arise naturally in logic. Indeed we will see that the property of covering set correctibility is an extension of the properties of completeness and soundness.

Our method will be to consider separately the two properties linked together by covering set correctibility, namely that of yielding a correction and that of being a covering set of failed clauses. Theorems 5.3.5 and 5.3.6 express each of the latter two properties in terms of some naturally arising constructs of logic. From the two theorems we can derive a logical expression for covering set correctibility, presented in Proposition 5.3.7.

We will first consider the property of yielding a correction. Theorem 5.3.5 will give us a logical expression for the statement that \( C \) yields a correction to the record \( v \) with respect to the set \( \Sigma \) of normal clauses. The logical expression will be in terms of the Th-satisfiability of a related clause set, the set \( \Sigma[v, C] \) of “reduced clauses”, to be defined in Definition 5.3.1.

We will secondly consider the property of being a covering set of failed clauses. The relevant covering sets are the covering sets of the set \( X(D(\Sigma), v) \), where \( D \) is a deduction function. Theorem 5.3.6 will give us a logical expression for the statement that \( C \) is a covering set of \( X(D(\Sigma), v) \), although in the theorem we generalise and replace \( D(\Sigma) \) by \( \Gamma \). The logical expression will again use the reduced clauses of Definition 5.3.1: in this case it will be in terms of whether the empty clause \( \Box \) is in the reduced clause set \( (D(\Sigma))[v, C] \).

Finally, in Proposition 5.3.7, we will link our two logical expressions, one for yielding a correction and the other for being a covering set, to produce a logical expression for covering set correctibility. Thus the covering set correctibility of \( D \) will link the Th-satisfiability of \( \Sigma[v, C] \) with whether the empty clause is in \( (D(\Sigma))[v, C] \). But completeness and soundness of \( D \) link similar properties: they link the Th-satisfiability of \( \Sigma \) with whether the empty clause is in \( D(\Sigma) \). Indeed covering set correctibility turns out to be an extension of completeness and soundness.

We now consider the property of yielding a correction. When we say that \( C \) yields a correction to the record \( v = (v_1, \ldots, v_N) \) with respect to the set \( \Sigma \) of normal clauses, we mean that \( \Sigma \) is Th-satisfiable subject to the restriction that each propositional atom \( p^v_j \) whose field \( j \) is in \( C \) must take the truth value \( f^v(p^v_j) \). Thus we should first assign the truth value \( f^v(p^v_j) \) for each propositional atom \( p^v_j \) whose field is in \( C \), and then recalculate each clause \( \sigma \) in \( \Sigma \) to get a “reduced clause” \( \sigma[v, C] \), to be defined formally below in Definition 5.3.1. It turns out that \( C \) yields a correction to the record \( v \) with respect to \( \Sigma \) if and only if the set \( \Sigma[v, C] \) of reduced clauses is Th-satisfiable - a result given below in the Theorem 5.3.5.

Before giving a formal definition of the reduced clause \( \sigma[v, C] \), we describe it in words as follows. If \( \sigma \) contains a propositional atom of the form \( p^v_j \) for some \( j \) in \( C \), then the reduced clause always takes the truth value \( \text{true} \), and we make the reduced
clause equal to \( \top \top \) in order to make it a normal clause. On the other hand, if \( \sigma \) contains no atom of the form \( p_jv \) for any \( j \) in \( C \), then the reduced clause will always take the same truth value as the clause \( \sigma \setminus \{ p_jy \mid j \in C, y \in A_j \} \), which we will define to be the reduced clause. More formally we make the following definition, where we replace \( C \) by \( Z \):

**Definition 5.3.1.** If \( \sigma \) is a normal clause, \( v = (v_1, \ldots, v_N) \) is a record, and \( Z \) is a field set, then the **reduction of \( \sigma \) by \( v \) and \( Z \)** is

\[
\sigma[v, Z] = \begin{cases} 
\sigma \setminus \{ p_jy \mid j \in Z, y \in A_j \}, & \text{if for each } j \text{ in } Z \text{ we have that } p_jv \not\in \sigma \\
\top \top, & \text{if there is a } j \text{ in } Z \text{ such that } p_jv \in \sigma.
\end{cases}
\]

Where the context is clear, we will call \( \sigma[v, Z] \) the **reduction of \( \sigma \)** or simply the **reduced clause**.

If \( \Sigma \) is a set of normal clauses and \( Z \) is a field set then the **reduction of \( \Sigma \) by \( v \) and \( Z \)** is

\[
\Sigma[v, Z] = \{ \sigma[v, Z] \mid \sigma \in \Sigma \}.
\]

Where the context is clear, we will call \( \Sigma[v, Z] \) the **reduction of \( \Sigma \)** or simply the **set of reduced clauses**.

**Note 1:** An equivalent statement for \( \sigma[v, Z] \) is

\[
\sigma[v, Z] = \begin{cases} 
\bigvee \{ p_jy \mid j \in Z \}, & \text{if for each } j \text{ in } Z \text{ we have that } p_jv \not\in \sigma \\
\top \top, & \text{if there is a } j \text{ in } Z \text{ such that } p_jv \in \sigma.
\end{cases}
\]  

(5.3.1)

**Note 2:** \( \Sigma[v, Z] \) contains only normal clauses.

**Note 3:** In the previous papers (Boskovitz, Goré, and Wong 2005; Boskovitz and Goré 2005), the set \( \Sigma[v, Z] \) was defined to be smaller than here: I dropped \( \sigma[v, Z] \) from \( \Sigma[v, Z] \) when \( \sigma[v, Z] = \top \top \). In the latter paper, I used a different notation: instead of \( \Sigma[v, Z] \) I used \( \Sigma[v_i \mid i \not\in Z] \).

**Note 4:** The method of obtaining the set \( \Sigma[v, Z] \) from the set \( \Sigma \) is based on the same idea as the DPLL splitting rule (Davis et al. 1962) described in Chapter 3, Section 3.2.6. The main difference between the method of obtaining the set \( \Sigma[v, Z] \) and the DPLL splitting rule is that in calculating \( \Sigma[v, Z] \) we simultaneously assign truth values to all the propositional atoms in the set \( \{ p_jy \mid j \in Z, y \in A_j \} \) rather than to just a single propositional atom. Also, \( \Sigma[v, Z] \) retains \( \sigma[v, Z] = \top \top \) under the second condition of the definition of \( \sigma[v, Z] \), whereas the DPLL splitting rule deletes \( \sigma[v, Z] \) under this second condition.

**Example 5.3.2.** Suppose that \( N = 3 \); that \( A_1 = A_2 = A_3 = \{1, 2, 3\} \); that \( Z = \{1\} \);
and that \( \boldsymbol{v} = (3, 3, 3) \). Then

\[
(p_1 \lor p_2^3)[\boldsymbol{v}, \boldsymbol{Z}] = p_2^3;
\]

\[
p_1^3[\boldsymbol{v}, \boldsymbol{Z}] = \top;\]

\[
p_2^3[\boldsymbol{v}, \boldsymbol{Z}] = p_2^3;
\]

\[
(p_1^3 \lor p_2^3)[\boldsymbol{v}, \boldsymbol{Z}] = \top \top;
\]

\[
p_3^3[\boldsymbol{v}, \boldsymbol{Z}] = \top \top.
\]

Having defined reduced reduced clauses, we are ready to express the property of yielding a correction in terms of the Th-satisfiability of reduced clauses. We will show below, in Theorem 5.3.5, that \( \mathcal{C} \) yields a correction to the record \( \boldsymbol{v} \) with respect to \( \Sigma \) if and only if \( \Sigma[\boldsymbol{v}, \mathcal{C}] \) is Th-satisfiable.

The proof of Theorem 5.3.5 depends on the following lemma and corollary comparing Th-truth functions that satisfy \( \Sigma \) to Th-truth functions that satisfy the reduced set \( \Sigma[\boldsymbol{v}, \boldsymbol{Z}] \), for \( \boldsymbol{Z} \) a set of fields. Rather than being in terms of a set \( \Sigma \) of clauses, the lemma and corollary are in terms of an individual clause \( \sigma \).

**Lemma 5.3.3.** Let \( \boldsymbol{Z} \) be a set of fields; and let \( \sigma \) be a normal clause. Let \( \boldsymbol{v} = (v_1, \ldots, v_N) \), \( \boldsymbol{x} = (x_1, \ldots, x_N) \) and \( \boldsymbol{w} = (w_1, \ldots, w_N) \) be records such that \( \boldsymbol{w} \) is the same as \( \boldsymbol{v} \) on the fields in \( \boldsymbol{Z} \), and \( \boldsymbol{w} \) is the same as \( \boldsymbol{x} \) on the fields in \( \overline{\boldsymbol{Z}} \). That is, for each \( j \) in \( \{1, \ldots, N\} \),

\[
j \in \boldsymbol{Z} \implies w_j = v_j, \quad \text{and} \quad j \in \overline{\boldsymbol{Z}} \implies w_j = x_j.
\]

Then \( f_\boldsymbol{w} \) satisfies \( \sigma \) if and only if \( f_\boldsymbol{x} \) satisfies \( \sigma[\boldsymbol{v}, \boldsymbol{Z}] \).

**Proof.** We consider the two cases in the definition of \( \sigma[\boldsymbol{v}, \boldsymbol{Z}] \).

**Case 1.** Suppose that for each \( j \) in \( \boldsymbol{Z} \), we have that \( p_j^{v_j} \not\in \sigma \). Then we can deduce the required result from the following sequence of equivalent statements:

\[
f_\boldsymbol{w}(\sigma) = \text{true}
\]

\[
\iff \text{there is a field } k \text{ in } \{1, \ldots, N\} \text{ such that } p_k^{w_k} \in \sigma, \quad \text{by Lemma 4.2.18, since } \sigma \text{ is a positive clause}
\]

\[
\iff \text{there is a field } k \text{ in } \overline{\boldsymbol{Z}} \text{ such that } p_k^{w_k} \in \sigma[\boldsymbol{v}, \boldsymbol{Z}], \quad \text{by Equation 5.3.1 above}
\]

\[
\iff \text{there is a field } k \text{ in } \overline{\boldsymbol{Z}} \text{ such that } p_k^{w_k} \in \sigma, \quad \text{since if } k \in \overline{\boldsymbol{Z}} \text{ then } w_k = x_k.
\]

\[
f_\boldsymbol{x}(\sigma[\boldsymbol{v}, \boldsymbol{Z}]) = \text{true}, \quad \text{by Lemma 4.2.18, since } \sigma[\boldsymbol{v}, \boldsymbol{Z}] \text{ is a positive clause}.
\]

**Case 2.** Suppose that there is a field \( j \) in \( \boldsymbol{Z} \) such that \( p_j^{v_j} \in \sigma \). Then by assumption \( w_j = v_j \), so that \( p_j^{w_j} \in \sigma \). Then \( f_\boldsymbol{w}(\sigma) = \text{true} \). Also, since by its definition, \( \sigma[\boldsymbol{v}, \boldsymbol{Z}] = \top \top \), we have that \( f_\boldsymbol{w}(\sigma[\boldsymbol{v}, \boldsymbol{Z}]) = \text{true} \), and hence we have the required result. \( \dashv \)
Corollary 5.3.4. Let \( Z \) be a set of fields; and let \( \sigma \) be a normal clause. Let \( \mathbf{v} = (v_1, \ldots, v_N) \) and \( \mathbf{w} = (w_1, \ldots, w_N) \) be records such that if \( j \in Z \) then \( w_j = v_j \). Then \( f_w \) satisfies \( \sigma \) if and only if \( f_w \) satisfies \( \sigma[\mathbf{v}, Z] \).

Proof. In Lemma 5.3.3, let \( x = w \). \( \dashv \)

We now use the above lemma and corollary in the proof of the next theorem, giving a logical expression for the property of yielding a correction.

Theorem 5.3.5. Let \( C \) be a set of fields, let \( \mathbf{v} \) be a record, and let \( \Sigma \) be a set of normal clauses. Then the field set \( C \) yields a correction to the Th-truth function \( f_\mathbf{v} \) with respect to the clause set \( \Sigma \) if and only if \( \Sigma[\mathbf{v}, C] \) is Th-satisfiable.

Proof. Forward direction. Suppose that the field set \( C \) yields the correction \( f_w \) to the Th-truth function \( f_\mathbf{v} \) with respect to the clause set \( \Sigma \). We write \( \mathbf{v} = (v_1, \ldots, v_N) \) and \( \mathbf{w} = (w_1, \ldots, w_N) \). Then by Definition 5.2.1,

a. if \( j \in C \) then \( w_j = v_j \); and

b. \( f_w \) satisfies \( \Sigma \).

Then, by Corollary 5.3.4 (with \( Z = C \)), the truth function \( f_w \) satisfies each clause of \( \Sigma[\mathbf{v}, C] \), which is thereby Th-satisfiable, as required.

Backward direction. Suppose that \( \Sigma[\mathbf{v}, C] \) is satisfied by the Th-truth function \( f_x \), where \( \mathbf{v} = (v_1, \ldots, v_N) \) and \( \mathbf{x} = (x_1, \ldots, x_N) \). Define \( \mathbf{w} = (w_1, \ldots, w_N) \) so that it differs from \( \mathbf{v} \) only on the fields of \( C \), where it is defined to be the same as \( \mathbf{x} \):

\[
w_j = \begin{cases} v_j, & j \in C \\ x_j, & j \in C \end{cases}
\]

Then, by Lemma 5.3.3 (with \( Z = C \)), we have that \( f_w \) Th-satisfies \( \Sigma \). Hence \( C \) yields a correction \( f_w \) to \( f_x \) with respect to \( \Sigma \), as required. \( \dashv \)

Having found a logical expression for the property of yielding a correction, we will now find a logical expression for the other component of covering set correctibility, namely the property of being a covering set of the set \( \mathcal{X}(D(\Sigma), \mathbf{v}) \), where \( D \) is a deduction function, \( \Sigma \) is a set of normal clauses, and \( \mathbf{v} = (v_1, \ldots, v_N) \) is a record. To simplify notation and generalise the question, we will replace \( D(\Sigma) \) by \( \Gamma \), and seek a logical expression for the property of being a covering set of \( \mathcal{X}(\Gamma, \mathbf{v}) \).

When we say that \( C \) is a covering set of \( \mathcal{X}(\Gamma, \mathbf{v}) \), we mean that if the propositional atoms with fields in \( \mathcal{C} \) are removed from any clause \( \gamma \) of \( \mathcal{X}(\Gamma, \mathbf{v}) \), then the new clause will not be empty. But the new clause is just the clause \( \gamma[\mathbf{v}, \mathcal{C}] \), as follows: since \( \gamma \in \mathcal{X}(\Gamma, \mathbf{v}) \), we have that \( \gamma \) contains no propositional atom of form \( p_j^{v_j} \), and hence \( \gamma[\mathbf{v}, \mathcal{C}] \) is the clause obtained by removing from \( \gamma \) the propositional atoms with fields in \( \mathcal{C} \). Thus the statement that \( C \) is a covering set of \( \mathcal{X}(\Gamma, \mathbf{v}) \) means that for each \( \gamma \) in...
\( \mathcal{X}(\Gamma, v) \), the clause \( \gamma[v, C] \) is non-empty. In fact we will see in Theorem 5.3.6 that even for those clauses \( \gamma \) not in \( \mathcal{X}(\Gamma, v) \) (but still in \( \Gamma \)), the clause \( \gamma[v, C] \) is not empty. Thus the property of be a covering set can also be expressed in terms of reduced clauses: the next theorem states that \( C \) is a covering set of the set \( \mathcal{X}(\Gamma, v) \) if and only if \( \Gamma[v, C] \) does not contain the empty clause.

**Theorem 5.3.6.** If \( C \) is a set of fields and \( \Gamma \) is a set of normal clauses, then \( C \) is a covering set of \( \mathcal{X}(\Gamma, v) \) if and only if \( \Box \not\in \Gamma[v, C] \).

**Proof.** We partition the set \( \Gamma \) into \( \mathcal{X}(\Gamma, v) \) and its complement \( \Gamma \setminus \mathcal{X}(\Gamma, v) \) which we call \( \Delta \). We will prove a statement for each partition:

1. if \( \gamma \in \mathcal{X}(\Gamma, v) \), then \( C \) covers \( \{ \gamma \} \) if and only if \( \gamma[v, C] \neq \Box \);
2. \( \Box \not\in \Delta[v, C] \), where \( \Delta = \Gamma \setminus \mathcal{X}(\Gamma, v) \).

We will then combine the above two statements to obtain the required result.

**Proof of statement 1.** Since \( \gamma \in \mathcal{X}(\Gamma, v) \), then \( f_w(\gamma) = \text{false} \) so that for each field \( j \) we have that \( p^w_j \not\in \gamma \) (by Lemma 4.2.18). Hence by Equation 5.3.1, we have that \( \gamma[v, C] = \bigvee \{ p^w_j \in \gamma \mid j \in C \} \). Hence the statement that \( \gamma[v, C] \neq \Box \) is equivalent to saying that there is some field \( k \) in \( C \) and some field value \( w \) in \( A_k \) such that \( p^w_k \in \gamma \), which is the same as saying that \( C \) covers \( \{ \gamma \} \).

**Proof of statement 2.** Let \( \delta \in \Delta \). Then by the definition of \( \Delta \), we have that \( f_w(\delta) = \text{true} \) and thus there is a field \( k \) such that \( p^w_k \in \delta \). We now consider the two cases in the definition of \( \delta[v, C] \):

- **Case a.** Suppose that for each \( j \) in \( C \), we have that \( p^w_j \not\in \delta \). Hence \( k \in C \). Also, by Equation 5.3.1, we have that \( \delta[v, C] = \bigvee \{ p^w_j \in \delta \mid j \in C \} \), which contains \( p^w_k \) since \( k \in C \). Hence \( \delta[v, C] \) is non-empty.

- **Case b.** Suppose that there is a field \( j \) in \( C \) such that \( p^w_j \in \delta \). Then by definition \( \delta[v, C] = \top \top \), which is non-empty.

In either case \( \delta[v, C] \) is non-empty, as required.

**Combining statements 1 and 2.** The required result follows from the following sequence of equivalent statements:

\[
\begin{align*}
\text{C is a covering set of } \mathcal{X}(\Gamma, v) \\
& \iff \text{for each clause } \gamma \text{ in } \mathcal{X}(\Gamma, v), \text{ the set } C \text{ is a covering set of } \{ \gamma \} \\
& \iff \text{for each clause } \gamma \text{ in } \mathcal{X}(\Gamma, v), \text{ the clause } \gamma[v, C] \neq \Box, \text{ by Statement 1 above} \\
& \iff \Box \not\in (\mathcal{X}(\Gamma, v))[v, C] \\
& \iff \Box \not\in (\mathcal{X}(\Gamma, v))[v, C] \cup \Delta[v, C], \text{ by Statement 2 above} \\
& \iff \Box \not\in \Gamma[v, C], \text{ by the definition of } \Delta.
\end{align*}
\]
With the above theorem we now have a logical expression for each of the two properties linked by covering set correctibility. We bring the two theorems together in the next proposition to give a logical expression for covering set correctibility, as well as of its two directions, error correction totality and error correction guarantee.

**Proposition 5.3.7.** Let $D$ be a deduction function.

1. The deduction function $D$ has error correction totality if and only if the following statement holds: For each set $\Sigma$ of normal clauses, each field set $Z$, and each record $v$

   $$\Sigma[v, Z] \text{ is Th-satisfiable } \Rightarrow \square \notin (D(\Sigma))[v, Z].$$

2. The deduction function $D$ has the error correction guarantee if and only if the following statement holds: For each set $\Sigma$ of normal clauses, each field set $Z$, and each record $v$

   $$\square \notin (D(\Sigma))[v, Z] \Rightarrow \Sigma[v, Z] \text{ is Th-satisfiable}.$$

3. The deduction function $D$ is covering set correctible if and only if the following statement holds: For each set $\Sigma$ of normal clauses, each field set $Z$, and each record $v$

   $$\square \notin (D(\Sigma))[v, Z] \iff \Sigma[v, Z] \text{ is Th-satisfiable}.$$

**Proof.** For the proof of part 1, we use the definition of error correction totality given in Definition 5.2.6. We start with the formula used in that definition and derive the following sequence of equivalent statements:

$$\mathcal{EL}(\Sigma, v) \subseteq \mathcal{CX}(D(\Sigma), v)$$

$\iff$ for each field set $C$, if $C$ yields a correction to $v$ with respect to $\Sigma$, then $C$ is a covering set of $\mathcal{CX}(D(\Sigma), v)$ (by the definitions of $C$ and $\mathcal{EL}$)

$\iff$ for each field set $C$, if $\Sigma[v, C]$ is Th-satisfiable, then $\square \notin (D(\Sigma))[v, C]$ (by Theorems 5.3.5 and 5.3.6)

$\iff$ for each field set $Z$, if $\Sigma[v, Z]$ is Th-satisfiable, then $\square \notin (D(\Sigma))[v, Z]$ (since there is a one-to-one correspondence between field sets and their complements).

Since the above sequence of statements holds for any $\Sigma$ and $v$, then part 1 follows.

The proof of part 2 is identical in structure to the proof of part 1. In the first line of the sequence of equivalent statements, the subset relation is reversed to obtain the
definition of the error correction guarantee. In the remaining lines of the sequence of equivalent statements, the implication direction is reversed.

Part 3 follows directly from parts 1 and 2.

The expressions in the above proposition are strengthenings of refutation Th-soundness and refutation Th-completeness (here we distinguish between refutation and strong Th-soundness as we did in Chapter 3, to maintain the symmetry with refutation and strong Th-completeness). Indeed, if \( Z \) is empty, then the above proposition reduces to a statement of the refutation Th-soundness of those deduction functions that have error correction totality, and a statement of the refutation Th-completeness of those deduction functions that have the error correction guarantee. However refutation Th-soundness and refutation Th-completeness on their own are not enough to ensure error correction totality or the error correction guarantee, as will be seen in the next section.

### 5.4 Soundness, completeness, and covering set correctibility

In this section we look at the relationships between soundness, completeness and covering set correctibility. We will consider separately the two directions of covering set correctibility, namely error correction totality and the error correction guarantee.

We already have, as an immediate consequence of Proposition 5.3.7, that each function with error correction totality is refutation Th-sound, and that each function with the error correction guarantee is refutation Th-complete. In this section we will see that both error correction totality and the error correction guarantee have stronger, but not parallel, properties. We first deal with error correction totality: we will see in Proposition 5.4.1 that error correction totality is equivalent to strong Th-soundness. However, we will find that the parallel property does not apply to the error correction guarantee: it is not equivalent to strong Th-completeness. Instead we will show that the error correction guarantee lies between strong Th-completeness and refutation Th-completeness - in Proposition 5.4.2, Example 5.4.3, Lemma 5.4.4 and Example 5.4.5.

Just as we related soundness and completeness to satisfiability in Lemma 3.2.26, so we here will relate error correction totality and the error correction guarantee to satisfiability. We will see in Proposition 5.4.6 and Example 5.4.7 that error correction totality has an equivalent definition in terms of satisfiability, but for the error correction guarantee the corresponding property is implied rather than equivalent.

We start with the equivalence of strong Th-soundness and error correction totality.

**Proposition 5.4.1.** The deduction function \( D \) has error correction totality if and only if it is strongly Th-sound.

**Proof.**

**Forward direction.** Suppose that \( D \) has error correction totality. Let \( \Sigma \) be a set of normal clauses and let \( \nu \) be a record such that \( f_\nu \) satisfies \( \Sigma \), so that \( \emptyset \in EL(\Sigma, \nu) \). Then, since \( D \) has error correction totality, we have that \( \emptyset \in CX(D(\Sigma), \nu) \), which means that
\( \mathcal{X}(D(\Sigma), v) = \emptyset \). Hence \( f_v \) satisfies \( D(\Sigma) \). This gives us that \( D \) is strongly Th-sound.

**Backward direction.** Suppose that \( D \) is strongly Th-sound. We will use the characterisation of error correction totality given by Proposition 5.3.7, part 1. Let \( \Sigma \) be a set of normal clauses; let \( v = (v_1, \ldots, v_N) \) be a record; and let \( Z \) be a set of fields. We suppose that \( \Sigma[v, Z] \) is Th-satisfiable, and we will show that \( \square \notin (D(\Sigma))[v, Z] \).

Suppose that a Th-truth function that satisfies \( \Sigma[v, Z] \) is \( f_x \), with \( x = (x_1, \ldots, x_N) \). Let \( w = (w_1, \ldots, w_N) \) be defined for each \( j = 1, \ldots, N \) by

\[
 w_j = \begin{cases} 
 v_j, & j \in Z \\
 x_j, & j \in \overline{Z}.
\end{cases}
\]

Then by Lemma 5.3.3, we have that \( f_w \) satisfies \( \Sigma \). Then, by the strong Th-soundness of \( D \), we also have that \( f_w \) satisfies \( D(\Sigma) \). Then using Corollary 5.3.4, we have that \( f_w \) satisfies \( (D(\Sigma))[v, Z] \), so that \( \square \notin (D(\Sigma))[v, Z] \), as required.

Although error correction totality is equivalent to strong Th-soundness, the error correction guarantee is not equivalent to strong Th-completeness. However, each function with strong Th-completeness does have the error correction guarantee, as seen in the next proposition.

**Proposition 5.4.2.** If \( D \) is a strongly Th-complete deduction function, then it has the error correction guarantee.

**Proof.** We will use the characterisation of the error correction guarantee given by Proposition 5.3.7, part 2. Let \( \Sigma \) be a set of normal clauses; let \( v = (v_1, \ldots, v_N) \) be a record; and let \( Z \) be a set of fields. We suppose that \( \Sigma[v, Z] \) is not Th-satisfiable, and we will show that \( \square \in (D(\Sigma))[v, Z] \).

Since \( \Sigma[v, Z] \) is not Th-satisfiable, then for each vector \( w \) in \( D \) there is a normal clause \( \alpha_w \) in \( \Sigma[v, Z] \) such that \( f_w(\alpha_w) = \text{false} \). Hence \( \alpha_w \neq \top \top \) and by Definition 5.3.1, there is a normal clause \( \sigma_w \) in \( \Sigma \) such that \( \alpha_w = \sigma_w \setminus \{p_j^y \mid j \in Z, y \in A_j\} \) and for each \( j \) in \( Z \) we have that \( p_j^y \notin \sigma_w \).

Let \( \delta \) be the disjunction of those propositional atoms that are in some \( \sigma_w \) and that have their field in the set \( Z \), that is:

\[
\delta = \bigvee \{p_j^y \mid j \in Z, y \in A_j\}, \text{ and there is a } w \text{ in } D \text{ such that } p_j^y \in \sigma_w \}.
\]

We will prove two things. Firstly, we will show that \( \delta \in D(\Sigma) \). Secondly, we will show that \( \delta[v, Z] = \square \). We will then have the required result that \( \square \in (D(\Sigma))[v, Z] \).

We first show that \( \delta \in D(\Sigma) \), which we obtain by showing that \( \Sigma \vdash_{\text{Th}} \delta \) and then using strong Th-completeness. In order to show that \( \Sigma \vdash_{\text{Th}} \delta \), let \( f_x \) be a Th-truth function that satisfies \( \Sigma \), where \( x = (x_1, \ldots, x_N) \). By the above definition of \( \alpha_w \), we have that \( f_x(\alpha_w) = \text{false} \), so that for each \( j \) in \( \{1, \ldots, N\} \) we have that \( p_j^x \notin \alpha_w \) (by Lemma 4.2.18). Also, since \( f_x \) satisfies \( \Sigma \), we have that \( f_x(\sigma_x) = \text{true} \), so that there is a \( k \) in \( \{1, \ldots, N\} \) such that \( p_k^x \in \sigma_x \). Hence \( p_k^x \in \sigma_x \setminus \alpha_x \), so that \( k \in Z \). Hence, by
the definition of $\delta$, we have that $p_k^j \in \delta$. Hence $f_\delta(\delta) = \text{true}$, so that $\Sigma \models \text{Th} \delta$. Then by strong Th-completeness, $\delta \in D(\Sigma)$.

We now show that $\delta[\nu, Z] = \Box$. We have from above that for each $j$ in $Z$ and each vector $w$, the propositional atom $p_j^y \notin \sigma_w$. Hence by the definition of $\delta$, we also have for each $j$ in $Z$ that $p_j \notin \delta$. Hence by Definition 5.3.1, we have that $\delta[\nu, Z] = \delta \setminus \{p_j^y | j \in Z, y \in A_j\}$, which equals $\Box$.

Since $\delta \in D(\Sigma)$, we have that $\delta[\nu, Z] \in (D(\Sigma))[\nu, Z]$. Hence, since $\delta[\nu, Z] = \Box$, we have that $\Box \in (D(\Sigma))[\nu, Z]$, as required.

Although each strongly Th-complete deduction function has the error correction guarantee, the converse does not apply. The following is an example of a deduction function which has the error correction guarantee but which is not strongly Th-complete.

**Example 5.4.3.** The deduction function $F$ has the error correction guarantee (by Corollary 5.2.8) but is not strongly Th-complete. For example, suppose that $N = 2$, that $A_1 = A_2 = \{1, 2, 3\}$, and that $\Sigma = \{p_1\}$. Then $F(\Sigma) = \{p_1\}$. But $\Sigma \models \text{Th} p_1 \lor p_2$, and yet $p_1 \lor p_2 \notin F(\Sigma)$.

We have seen that the error correction guarantee is a strictly weaker property than strong Th-completeness. However, it is not as weak as Th-refutation completeness. Before giving a demonstrating example, we first confirm that the error correction guarantee is at least as strong as refutation Th-completeness.

**Lemma 5.4.4.** If the deduction function $D$ has the error correction guarantee, then it is refutation Th-complete.

**Proof.** The result follows from Proposition 5.3.7, part 2, in which we replace the set $Z$ by the empty set, and note that for any set $\Gamma$ of normal clauses (including $\Gamma = \Sigma$ and $\Gamma = D(\Sigma)$), we have that $\Gamma[\nu, \emptyset] = \Gamma$.

Although deduction functions with the error correction guarantee are refutation Th-complete, the converse does not apply. In the next example, we present a deduction function which is refutation Th-complete, but which does not have the error correction guarantee.

**Example 5.4.5.** Given a total ordering $\prec$ on the set $\text{PAtm}$ of propositional atoms (Definition 4.2.3), let the deduction function

$$\mathcal{N}R^\prec_{\text{Th}} : \mathcal{P}(\text{NC}) \to \mathcal{P}(\text{NC})$$

be defined for each set $\Sigma$ of clauses by

$$\mathcal{N}R^\prec_{\text{Th}}(\Sigma) = R^\prec_{\text{Th}}(\Sigma) \cap \text{NC},$$

where $R^\prec_{\text{Th}}$ was defined in Definition 3.2.19.

Then $\mathcal{N}R^\prec_{\text{Th}}$ is refutation Th-complete, because (i) $R^\prec_{\text{Th}}$ is refutation Th-complete by Theorem 3.2.30; and (ii) $\Box$ is normal.
However, although the function $NR_{Th}^{-}$ is refutation Th-complete, it does not have the error correction guarantee, as seen in the situation below.

Define the ordering relation $\prec$ on propositional atoms as follows:

$p_i^x \prec p_j^y$ if $i < j,$

or if $i = j$ and $x < y.$

Suppose that $N = 2$, that $A_1 = A_2 = \{1,2,3\}$, and that the clauses $\sigma_1, \sigma_2$, the clause set $\Sigma$, and the record $v$ are defined as follows:

$\sigma_1 = p_1^1 \lor p_2^1,$

$\sigma_2 = p_1^1 \lor p_2^2,$

$\Sigma = \{\sigma_1, \sigma_2\},$  and

$v = (2,3).$

Then the record $v$ fails both clauses of $\Sigma$. It cannot be corrected by changing only field 2 since $\sigma_1$ and $\sigma_2$ use different values for field 2. However it can be corrected by changing field 1, to the value 1. Hence

$EL(\Sigma, v) = \{\{1\}, \{1,2\}\}.$

Then we have the following sets:

$NR_{Th}^{-}(\Sigma) = \Sigma$

$C(NR_{Th}^{-}(\Sigma), v) = \Sigma$

$CX(NR_{Th}^{-}(\Sigma), v) = \{\{1\}, \{2\}, \{1,2\}\}.$

Hence $NR_{Th}^{-}$ does not have the error correction guarantee, because the set of covering sets is not a subset of the set of error localisation solutions. That is, $CX(NR_{Th}^{-}(\Sigma), v) \not\subseteq EL(\Sigma, v).$

So far in this section, we have examined the relationship between error correction totality and Th-soundness, and the relationship between the error correction guarantee and Th-completeness. Figure 5.1 summarises the results of this section: the left-hand box represents the fact that error correction totality is equivalent to strong Th-soundness, while the right-hand box represents the fact that the error correction guarantee lies between strong Th-completeness and refutation Th-completeness.

The results of this section so far may seem odd because they are not symmetrical, even though all the underlying relationships seem to be symmetrical: firstly, error correction totality and the error correction guarantee are generalisations of soundness and completeness respectively; secondly, error correction totality and the error correction guarantee are converses of each other; and thirdly, soundness and completeness are also converses of each other. Yet error correction totality is related to soundness in a different way from the error correction guarantee to completeness.

The underlying reason for the lack of symmetry is the lack of symmetry in equivalent
§5.4 Soundness, completeness, and covering set correctibility

The deduction function $D$ is strongly sound if and only if for each Th-truth function $f$ and each clause set $\Sigma$ the following holds:

$$f \text{ satisfies } \Sigma \Rightarrow f \text{ satisfies } D(\Sigma).$$  \hfill (5.4.1)

However, as noted in Chapter 3 (page 69), the converse statement, for each Th-truth function $f$ and each clause set $\Sigma$, is not equivalent to the strong completeness of $D$, where the converse statement is

$$f \text{ satisfies } D(\Sigma) \Rightarrow f \text{ satisfies } \Sigma.$$  \hfill (5.4.2)

The first statement, Statement 5.4.1, is used in the proof of the fact that error correction totality is equivalent to strong Th-soundness. However we cannot use the second statement, Statement 5.4.2, to prove a similar relationship between the error correction guarantee and strong Th-completeness.

The lack of symmetry persists when relating error correction totality and the error correction guarantee to Statements 5.4.1 and 5.4.2 respectively. Figure 5.2 expands...
Figure 5.2a: Expansion of Figure 5.1a to display the fact that Statement 5.4.1 is equivalent to error correction totality.

Figure 5.2b: Expansion of Figure 5.1b to display the fact that Statement 5.4.2 is equivalent to the error correction guarantee.

Figure 5.2: Expansion of Figure 5.1 to incorporate the relationship of Statements 5.4.1 and 5.4.2 to error correction totality and the error correction guarantee respectively. Figure 5.2a, at the top, represents the fact that error correction totality is equivalent to Statement 5.4.1. Figure 5.2b, at the bottom, represents the fact that the error correction guarantee only implies Statement 5.4.2, which is distinct from refutation completeness.

Figure 5.1 to represent the fact that, although error correction totality is equivalent to Statement 5.4.1, the error correction guarantee only implies Statement 5.4.2. Indeed the three properties, Statement 5.4.2, the error correction guarantee, and refutation completeness, are distinct from each other. We have already seen, in Example 3.2.28 part 2, that Statement 5.4.2 is distinct from refutation completeness. The remaining relationships are presented below in Proposition 5.4.6 and Example 5.4.7.

**Proposition 5.4.6.** Let $\mathcal{D}$ be a deduction function.

1. The deduction function $\mathcal{D}$ has error correction totality if and only if for each Th-truth function $f$ and each clause set $\Sigma$,

$$f \text{ satisfies } \Sigma \Rightarrow f \text{ satisfies } \mathcal{D}(\Sigma).$$

(5.4.1, repeated)
2. If the deduction function \( D \) has the error correction totality, then for each Th-truth function \( f \) and each clause set \( \Sigma \),

\[
f \text{satisfies } D(\Sigma) \Rightarrow f \text{satisfies } \Sigma. \tag{5.4.2, repeated}
\]

Proof.

1. Follows from Lemma 3.2.26 part 3 and Proposition 5.4.1.

2. Suppose that \( D \) has the error correction guarantee; that \( \Sigma \) is a clause set; and that \( f \) is a Th-truth function that satisfies \( D(\Sigma) \). We will show that \( f \) satisfies \( \Sigma \). Firstly, since \( f \) is a Th-truth function, it can be written \( f_v \) for some record \( v \). Since \( f_v \) satisfies \( D(\Sigma) \), the empty set yields a correction to \( f_v \) with respect to the clause set \( D(\Sigma) \). Hence by Theorem 5.3.5, with \( C = \emptyset \), we have that \( (D(\Sigma))[v,\emptyset] \) is Th-satisfiable. Hence since \( \square \) is not Th-satisfiable, we have that \( \square \not\in (D(\Sigma))[v,\emptyset] \), from which we can deduce, using Proposition 5.3.7 part 2, that \( \Sigma[v,\emptyset] \) is Th-satisfiable. Once again using Theorem 5.3.5 with \( C = \emptyset \), we can deduce that the empty set yields a correction to the Th-truth function \( f_v \) with respect to the clause set \( \Sigma \), that is \( f_v \) satisfies \( \Sigma \), as required.

\( \square \)

The above proposition tells us that error correction totality is equivalent to Statement 5.4.1, and that the error correction guarantee implies Statement 5.4.2. The next example demonstrates that the error correction guarantee is indeed distinct from Statement 5.4.2.

Example 5.4.7. This is an example of a deduction function for which Statement 5.4.2 is distinct from the error correction guarantee and also from refutation completeness.

The identity deduction function \( I \) defined for each clause set \( \Sigma \) by

\[
I(\Sigma) = \Sigma
\]

satisfies Statement 5.4.2, but does not have the error correction guarantee, nor is it refutation complete.

We have established in this section the parallels between the properties of covering set correctibility and of soundness / completeness and displayed them in Figure 5.2. The next step is to return to the original problem of error localisation and its parallel with the propositional satisfiability problem.

## 5.5 Error localisation and propositional satisfiability

As discussed in Section 3.2.6 of Chapter 3, the propositional satisfiability problem (known as SAT) is the problem of deciding whether a given set \( \Sigma \) of clauses is satisfiable. The error localisation problem is an extension of SAT: it is a satisfiability problem constrained by the given field set \( C \) and the given record \( v \). In this section, we will see
that not only is the error localisation problem an extension of the SAT problem, but the methods of solution are in parallel, and the reasons why the pure deduction method works for error localisation is an extension of the reasons why the pure deduction method works for SAT.

Table 5.1 summarises the parallels between error localisation and propositional satisfiability, each of which is allocated a separate column. The table lists several aspects for each of the two problems, and in each case the description for the error localisation problem is a strengthening of the description for the propositional satisfiability problem. We will consider each aspect in turn in the following discussion. The first aspect is the problem statement itself: as stated above, the error localisation problem is a strengthening of the SAT problem.

For both the error localisation problem and the SAT problem, most solution methods can be seen as a trade-off between search and deduction, as discussed in Chapter 2, Section 2.7, and Chapter 3, Section 3.2.6. In particular, as summarised in aspect 2 of Table 5.1, both problems can be solved by a pure deduction method, with no search component. For example, as discussed in Chapter 3, SAT can be solved using a suitable deduction function such as resolution, as seen in the work of Davis and Putnam (1960), and Rish and Dechter (Rish and Dechter 2000; Dechter and Rish 1994). On the other hand, the error localisation problem can also be solved using a suitable deduction function such as the Fellegi-Holt deduction function $F$.

Indeed, the two pure deduction methods use deduction in a similar way. For a suitable deduction function $D$, the set $\Sigma$ is satisfiable if and only if $\square \notin D(\Sigma)$. On the other hand, the set $\Sigma$ has an error localisation solution for the record $v$ and the field set $C$ if and only if $\square \notin (D(\Sigma))[v, C]$.

Not only can both problems be solved in a similar way using a suitable deduction function $D$, but $D$ is suitable in similar circumstances. On the one hand, SAT can be solved using $D$ if and only if $D$ is both refutation sound and refutation complete. For example, the resolution deduction function is both refutation sound and refutation complete. On the other hand, the error localisation problem can be solved using $D$ if and only if $D$ has both error correction totality and the error correction guarantee, which are strengthenings of refutation soundness and refutation completeness respectively. For example, the Fellegi-Holt deduction function has both error correction totality and the error correction guarantee.

Not only are the properties for error localisation strengthenings of the properties for SAT, but the two pairs of properties play the same roles with regard to the two problems: that is, refutation completeness plays the same role for the SAT problem as the error correction guarantee plays for the error localisation problem; and refutation soundness plays the same role for the SAT problem as error correction totality plays for the error localisation problem. In more detail, we first note that both problems have been stated as “yes / no” questions: the SAT problem as “is $\Sigma$ satisfiable?”, and the error localisation problem as “does the field set $C$ yield a correction to $v$ with respect to $\Sigma$?”. We summarise the situation for the answer “yes” in aspect 3 of Table 5.1. On the one hand, for SAT, the answer “yes” from the deduction method is always correct if and only if the deduction function is refutation complete. On the other hand, for error...
### Table 5.1: Comparison of error localisation and propositional satisfiability

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Error localisation</th>
<th>Propositional satisfiability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Problem statement</td>
<td>Given a set $\Sigma$ of normal clauses, a record $v$ and a set $C$ of fields, is $\Sigma[v, C]$ Th-satisfiable?</td>
<td>Given a set $\Sigma$ of clauses, is $\Sigma$ satisfiable?</td>
</tr>
<tr>
<td>2. Pure deduction solution method, using deduction function $D$</td>
<td>Return “yes” if and only if $\Box \notin (D(\Sigma))[v, C]$.</td>
<td>Return “yes” if and only if $\Box \notin D(\Sigma)$.</td>
</tr>
<tr>
<td>3. The property that the output “yes” is always correct</td>
<td>Error correction guarantee: $\Box \notin (D(\Sigma))[v, C] \Rightarrow \Sigma[v, C]$ is Th-satisfiable</td>
<td>Refutation completeness: $\Box \notin D(\Sigma) \Rightarrow \Sigma$ is satisfiable</td>
</tr>
<tr>
<td>4. The property that the output “no” is always correct</td>
<td>Error correction totality: $\Sigma[v, C]$ is Th-satisfiable $\Rightarrow \Box \notin (D(\Sigma))[v, C]$</td>
<td>Refutation soundness: $\Sigma$ is satisfiable $\Rightarrow \Box \notin D(\Sigma)$</td>
</tr>
<tr>
<td>5. Examples of deduction function $D$</td>
<td>Fellegi-Holt deduction $F$</td>
<td>Th-resolution $R_{Th}$</td>
</tr>
<tr>
<td></td>
<td>Field Code Forest $FCF_\omega$</td>
<td>Ordered Th-resolution $R_{Th}^\prec$</td>
</tr>
</tbody>
</table>
localisation, the answer “yes” is always correct if and only if the deduction function has the error correction guarantee. There is a similar parallel when the answer is “no”, as summarised in aspect 4 of Table 5.1. For SAT the answer “no” is always correct if and only if the deduction function is refutation sound; for error localisation the answer “no” is always correct if and only if the deduction function has error correction totality.

Given the above parallels, it is not surprising that there are parallels between the deduction functions for which the pure deduction methods work. We give two pairs of functions in aspect 5 of Table 5.1. The first pair consists of the Fellegi-Holt deduction function $F$ and the Th-resolution deduction function $R_{\text{Th}}$: we proved in Chapter 4 that $F$ is essentially the same as $R_{\text{Th}}$. The second pair of parallel deduction functions consists of the field code forest deduction function $\mathcal{FCF}$ and the ordered Th-resolution function $R_{\text{Th}}^\prec$. The function $\mathcal{FCF}$ builds on $F$ in the same way as $R_{\text{Th}}^\prec$ builds on $R_{\text{Th}}$: in both cases deduction steps are only allowed on the lowest order propositional atoms of the input clauses.

Thus the error localisation problem and the SAT problem are strongly related, in the problem statement, pure deduction solution method, and the reasons that the pure deduction method works. This means that the automated techniques of SAT could potentially be adapted to solve the error localisation problem.

5.6 Conclusion

This chapter has presented the beginnings of a theoretical logical framework for analysing methods of solving the error localisation problem. The main aspects are listed below.

1. The error localisation problem is a strengthening of the propositional satisfiability (SAT) problem.

2. Both the error localisation problem and the propositional satisfiability problem can be solved by a mixture of deduction and search, and indeed both problems can be solved by a pure deduction method.

3. The pure deduction method for the error localisation problem, that is the covering set method, is a strengthening of the pure deduction method for the SAT problem. Each method seeks to determine whether the empty clause is obtained in a calculated set, but for the error localisation problem the calculated set is more complex to calculate.

4. The pure deduction method of solving the error localisation problem depends on covering set correctibility in just the same way as the pure deduction method of solving the SAT problem depends on refutation completeness and soundness.

In order to obtain the above results, we first formalised covering set correctibility in terms of logic by directly translating from sets to logic the key definitions such as “involved” and “covering set”. We ensured that the direct translation was consistent with sets, so that the formalisations of functions that have covering set correctibility in sets also have covering set correctibility in logic. Any method of solving the error
localisation problem based on this formalisation would also be a direct translation from the methods used in sets.

We then analysed the meaning of covering set correctibility and sought some constructions of logic that have the same meaning. Covering set correctibility links together two other properties, namely that of yielding a solution and that of being a covering set of a set of failed clauses. Both properties can be defined in terms of sets of “reduced clauses”, obtained by fixing the truth values of certain propositional atoms. When expressed in terms of reduced clauses, covering set correctibility can be seen to be a strengthening of refutation completeness and soundness. The analysis of covering set correctibility in terms of logical constructs might help in converting the methods of SAT to error localisation.

Logic gives three benefits. Firstly, it gives an alternative way of analysing the problem and thus potentially gives new theoretical insights. Secondly, its collection of sophisticated automated tools could potentially be modified to use covering set correctibility for solving error localisation problems. Thirdly, the strong parallels between the two subjects are of aesthetic appeal.
Covering set correctibility from the perspective of logic: categorical edits