Strategies for Consistency Checking

H. Bowman, E.A. Boiten, J. Derrick and M. Steen

Computing Laboratory, University of Kent at Canterbury, Canterbury, CT2 7NF, UK.
(Phone: +44 1227 827913, Fax: 44 1227 762811
Email:
{H.Bowman,E.A.Boiten,J.Derrick,mwas}@ukc.ac.uk.

Introduction

This report describes an initial framework for consistency checking. The report is intended as a companion to the work presented in [1] and it should be read in association with this document. In particular, the body of this report is a single chapter which should be viewed as additional to the chapters included in [1].

This report contains complete proofs of all relevant results, even though some of the results are obvious and some of the proofs are trivial. A much compressed version of the report is being submitted for publication [2]. Thus, the main value of this report is as a reference document for readers who require a complete presentation of the technical issues surrounding the framework presented in [2].

Note. For some parts of the document we use $\overset{1..n}{\mathcal{U}}(dv_i, X_i)$ as a shorthand for $\mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n)$ and similar shorthands for $\mathcal{LU}$, $\mathcal{L}$ and $\mathcal{C}$.

Background Results

The following are straightforward results of the definitions contained in [1]. They will be used later in this report.

Proposition 1
Given $S = \{X_1, ..., X_n\}$ *and* $T = \{X'_1, ..., X'_m\}$ *then,*

$$(\forall X'_j \in T,\ \exists X_i \in S\ s.t.\ X'_j = X_i\ \wedge\ dv'_j = dv_i) \implies$$
$$\mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n) \subseteq \mathcal{U}[dv'_1, ..., dv'_m](X'_1, ..., X'_m).$$

Proof
If $\mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n) = \emptyset$ the result follows trivially. So, take $X \in \mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n)$ i.e. $X\ dv_1\ X_1\ \wedge ... \wedge\ X\ dv_n\ X_n$. For any X'_j $s.t.$ $1 \leq j \leq m$, $X\ dv'_j\ X'_j$ since by our condition $\exists X_i$ (for $1 \leq i \leq n$) such that $dv_i = dv'_j$ and $X_i = X'_j$. So, $X \in \mathcal{U}[dv'_1, ..., dv'_m](X'_1, ..., X'_m)$, as required. □

This proposition expresses the obvious result that a unification of n specifications is a unification of a subset of the n specifications. The correct correspondence of development relations to descriptions is guaranteed in the condition of the implication. An obvious corollary of this result is:

Corollary 1
$\mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n) \subseteq \mathcal{U}[dv_i, ..., dv_j](X_i, ..., X_j)$ *for* $1 \leq i, j \leq n$.

Proof
Immediate from proposition 1. $\square$

Chapter 1

Strategies for Consistency Checking - the Choice of Unification

[1] has investigated consistency in terms of a set of possible unifications, i.e. descriptions $X_1, X_2, ..., X_n$ are consistent if the set of possible unifications $\mathcal{U}[X_1, ..., X_n](X_1, ..., X_n)$ is non-empty. Such a unification set could be very large and very often infinite. Clearly, if a system development trajectory is to be provided for viewpoint models then it is important that we reduce the choice of unification. In particular, we would like to select just one description from the set of unifications. This would enable an incremental development strategy in which a group of viewpoints are unified and then this unification is further composed with another group of viewpoints. This situation amounts to obtaining global consistency from a series of non-global (probably binary) consistency checks and unifications. The objective of this chapter is to characterise the unification that should be chosen from the unification set. This characterisation will, not surprisingly, induce certain properties on the development relations used.

The structure of this chapter is as follows. First we consider the issue of *representative* unifications in section 1.1. This is followed with an investigation of binary consistency checking strategies in section 1.2. From here we focus on the important issue of *least developed unifications* in section 1.3 (this section contains the main technical results of the chapter). Then we consider more restricted classes of consistency in section 1.4 and finally we discuss the results of the chapter in section 1.5.

1.1 Representative Unification

A particular unification algorithm will construct just one member of the unification set. Importantly, we need to know that the unification that we construct is internally valid if and only if an internally valid unification exists; otherwise we may construct an internally invalid unification despite the fact that an alternative unification may be internally valid.

Thus, we introduce the concept of a representative unification, which is defined as follows:-

Definition 1 *$X \in \mathcal{U}[dv_1, dv_2, ..., d_n](X_1, X_2, ..., X_n)$ is a representative unification iff $(\exists X' \in \mathcal{U}[dv_1, dv_2, ..., d_n](X_1, X_2, ..., X_n)$ s.t. $\Psi(X')) \implies \Psi(X)$.*

The following result is very straightforward:-

Proposition 2
ft is implementation complete and $X_1, ..., X_n \in DES_{ft} \implies \forall X \in \mathcal{U}[dv_1, ..., d_n](X_1, ..., X_n)$, X is a representative unification.

So, this result implies that for a language such as LOTOS, representativeness of unification does not arise.

It is also worth pointing out that we would certainly expect the unification strategies that we adopt to yield a representative unification and it would be a major flaw in the strategy if it did not. As a reflection of this, for the remainder of this chapter we will largely assume representativeness of the unification functions that we consider.

1.2 Binary Consistency Checking Strategies

We would like to obtain global consistency through a series of binary consistency checks. We have found that naive pairwise checking does not give us this. However, a combination of binary consistency checks and binary unification of the form shown in figure 1.1 should intuitively work, i.e. X_1 and X_2 are checked for consistency, then a unification of X_1 and X_2 is obtained, which is checked for consistency against X_3, then a unification of X_3 and the previous unification is performed. This process is continued through the n viewpoint descriptions. Thus, the base case is a binary consistency check and then repeated unification and binary consistency checks are performed against the next description. Of course, this is just one possible sequence of binary consistency checks. We would like to obtain full associativity results which support any appropriate incremental consistency checking strategy. However, as an archetypal approach. the binary consistency checking strategy of figure 1.1 will serve as an initial focus for our investigations.

The advantages of such incremental consistency checking strategies are that they do not force the involvement of all viewpoints in every consistency check. In particular, it may be possible to incrementally correct inconsistencies. In addition, such an approach will aid maintaining structure when unifying. It is very unlikely that a single unification of six viewpoints will be able to reconcile the structure of all the views, however, an incremental focus of restructuring may be possible.

The following definition characterises this binary consistency checking strategy. We denote the strategy Π_U, where U is a particular binary unification function, U takes two descriptions and returns a set of unifications of the pair. Notice that we assume U generates a set of possible unifications. This is because we would like to be as general as possible about the results we derive at this stage. In particular, it should be clear that, the generation of a single unification is a special case of the derivation of a set of unifications. We will impose two constraints on U in definition 3 which characterise when U can be viewed to be a *valid* unification function.

Definition 2

$$\Pi_U[dv_1, ..., dv_n](X_1, ..., X_n) \stackrel{def}{=}$$

$$
\begin{array}{ll}
((\exists Y_1 \in U[dv_1, dv_2](X_1, X_2) \;\wedge\; \Psi(Y_1)) \quad \wedge & \textit{- Step 1} \\
(\exists Y_2 \in U[dv_1 \cap dv_2, dv_3](Y_1, X_3) \;\wedge\; \Psi(Y_2)) \quad \wedge & \textit{- Step 2} \\
(\exists Y_3 \in U[dv_1 \cap dv_2 \cap dv_3, dv_4](Y_2, X_4) \;\wedge\; \Psi(Y_3)) \quad \wedge & \textit{- Step 3} \\
\qquad \ldots & \\
\qquad \ldots & \\
(\exists Y_{n-2} \in U[dv_1 \cap dv_2 \cap .... \cap dv_{n-2}, dv_{n-1}](Y_{n-3}, X_{n-1}) \;\wedge\; \Psi(Y_{n-2})) \quad \wedge & \textit{- Step n-2} \\
(\exists Y_{n-1} \in U[dv_1 \cap dv_2 \cap .... \cap dv_{n-1}, dv_n](Y_{n-2}, X_n) \;\wedge\; \Psi(Y_{n-1}))) & \textit{- Step n-1}
\end{array}
$$

Thus, each step in the algorithm considers a unification set using the binary unification function U. The ith step is satisfied if a description, Y_i, can be found in the set of unifications generated by the function U that is internally valid and can be used to satisfy the i+1st step. A depiction of Π_U, with n=4, is given in figure 1.2. It should be apparent that consistency checking is implicit in each step. Thus, the existence of an internally valid ith unification, Y_i, ensures that Y_{i-1} and X_{i+1} are consistent. Clearly, if an internally valid unification does not exist for a particular step then consistency would be lost.

Notice this definition does not prescribe in which order a pair of respective binary consistency checks and binary unifications are to be carried out; in each step of the algorithm we could either check consistency first and then unify or unify and then check consistency. The former of these

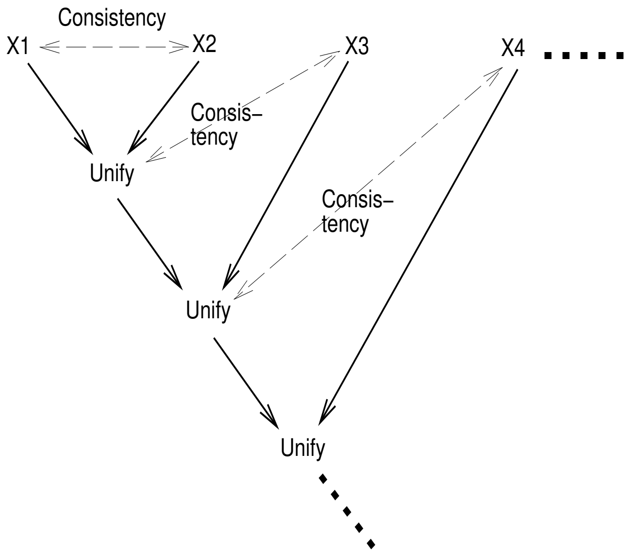

Figure 1.1: Binary Consistency Algorithm

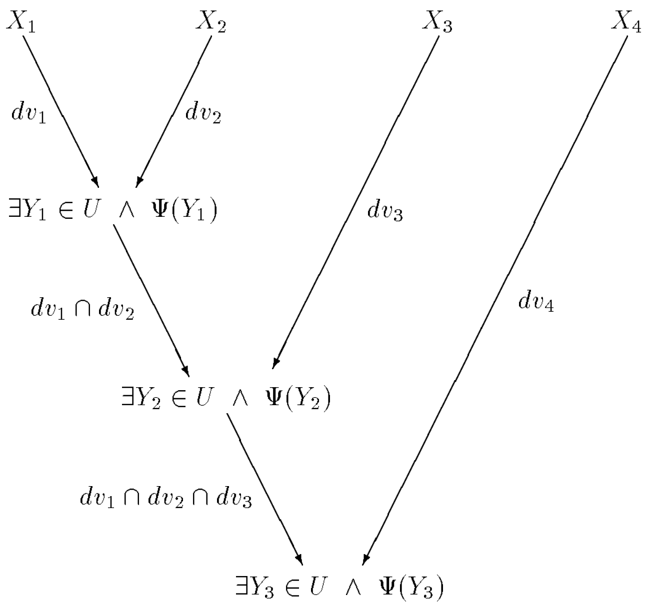

Figure 1.2: Formal Depiction of Binary Consistency Algorithm

alternatives is the strategy we employ for LOTOS, while the latter is the strategy we employ for Z. The reason for these alternatives, is that for LOTOS a unification may not always exist, thus, it is sensible to undertake a consistency check first before looking for a potentially non-existent unification. The situation is reversed for Z, where a unification always exists, but this unification may not be internally valid. Thus, an immediate unification is the obvious strategy to employ.

As mentioned earlier the unification construction function, U, yields a set of unifications, which could possibly be a singleton. We assume U satisfies the following constraints:-

Definition 3
A binary unification function U is valid if and only if,

(U.i) $U[dv, dv'](X, X') \subseteq \mathcal{U}[dv, dv'](X, X')$ *and*
(U.ii) $\mathcal{U}[dv, dv'](X, X') = \emptyset \implies U[dv, dv'](X, X') = \emptyset$.

These are minimal constraints that ensure U is a sensible binary unification method. (U.i) guarantees that the unifications generated by U are in the set of all unifications obtained by $\mathcal{U}$ and (U.ii) ensures that if a unification exists, U will not yield the empty set. Using these constraints we can show that if our binary consistency checking strategy is satisfied the consistency follows:-

Proposition 3
Assuming dv_i, $1 \leq i \leq n$ is a preorder and U satisfies (U.i) and (U.ii),
$\Pi_U[dv_1, ..., dv_n](X_1, ..., X_n) \implies C[dv_1, ..., dv_n](X_1, ..., X_n)$.

Proof
Assume $\Pi_U[dv_1, ..., dv_n](X_1, ..., X_n)$ holds. Now from step n-1 in Π_U we deduce:

$$\exists Y \ s.t. \ Y \ (dv_1 \cap dv_2 \cap ... \cap dv_{n-1}) \ Y_{n-2} \ \wedge \qquad (1)$$
$$Y \ dv_n \ X_n \ \wedge \qquad (2)$$
$$\Psi(Y) \qquad (3)$$

We will show that Y is the required common development of X_1 through to X_n to give us $C[dv_1, ..., dv_n](X_1, ..., X_n)$. Firstly, (2) and (3) give us immediately that $\Psi(Y)$ and $Y \ dv \ X_n$. Now from (1) and $Y_{n-2} \in U[dv_1 \cap ... \cap dv_{n-2}, dv_{n-1}](Y_{n-3}, X_{n-1})$ we can deduce that $Y \ dv_{n-1} \ Y_{n-2}$ and $Y_{n-2} \ dv_{n-1} \ X_{n-1}$, thus, from transitivity of dv_{n-1} we have $Y \ dv_{n-1} \ X_{n-1}$. We can perform similar arguments down through the construction of Π to determine that $Y \ dv_{n-2} \ X_{n-2} \wedge ... \wedge Y \ dv_2 \ X_2 \wedge Y \ dv_1 \ X_1$. Thus, Y is the required common development and $C[dv_1, ..., dv_n](X_1, ..., X_n)$ holds. □

Using this result we can show that performing Π with the full unification set function, i.e. instantiating $\mathcal{U}$ for U, is equal to consistency. Clearly, we would expect this to be the case and if it was not we would have to worry about Π.

Proposition 4
Assume dv_i, $1 \leq i \leq n$, is a preorder. Then,

$$\Pi_{\mathcal{U}}[dv_1, ..., dv_n](X_1, ..., X_n) = C[dv_1, ..., dv_n](X_1, ..., X_n).$$

Proof
($\Longrightarrow$) $\mathcal{U}$ trivially satisfies (U.i) and (U.ii); thus we can use the previous result, proposition 3, to give this direction of implication.
($\Longleftarrow$) Assume $C[dv_1, ..., dv_n](X_1, ..., X_n)$ holds, i.e. $\exists Y \in \mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n)$ such that $\Psi(Y)$. We will show that Y can act as the unification in all steps of Π. Firstly, the internal validity requirement of each step will clearly be satisfied for Y. In addition, using corollary 1 we get, $Y \in \mathcal{U}[dv_1, dv_2](X_1, X_2)$ and thus step 1. Step 2 follows since $Y \ dv_3 \ Y_3$, by our assumption and $Y \ dv_1 \cap dv_2 \ Y$ from the reflexivity of development, i.e. $Y \in \mathcal{U}[dv_1 \cap dv_2, dv_3](Y, X_3)$. Using similar arguments we can get step 3 and all steps up to n-1 as required. □

However, if we use a valid unification construction function (i.e. one that satisfies (U.i) and (U.ii)) other than $\mathcal{U}$ the converse to proposition 3 does not, in general, follow, i.e. $C \not\Longrightarrow \Pi_U$, and we clearly require this direction if Π is to be used.

Example 1 *We will give two simple examples of why a binary consistency checking strategy may not give global consistency. The first example is for LOTOS and the second is for Z.*

LOTOS. *Consider the three LOTOS specifications,* $P_1 := i; a; stop[]i; b; stop$, $P_2 := a; stop[]i; b; stop$ *and* $P_3 := a; stop$. *Further consider the consistency check* $C[\mathbf{red}](P_1, P_2, P_3)$, *where* **red** *is the LOTOS reduction relation, which refines through reduction of non-determinism. The three specifications are consistent by reduction since* P_3 *is a reduction of all three specifications. However, if we attempt a binary consistency checking algorithm and started with* P_1 *and* P_2 *we may choose as the unification of these two the process* $P := i; b; stop$, *and* $C[\mathbf{red}](P, P_3)$ *does not hold.*

Z. *Consider the three Z specifications,* $S_1 = [n! : \mathbb{N} \mid n! = 5 \ \vee \ n! = 7]$, $S_2 = [n! : \mathbb{N} \mid n! = 5 \ \vee \ n! = 7]$, *and* $S_3 = [n! : \mathbb{N} \mid n! = 5]$. *The first two specifications could be unified to yield* $[n! : \mathbb{N} \mid n! = 7]$, *which is not consistent with the third. But, the third specification could act as a refinement of all three.*

These examples suggest the class of unifications that we must select. Specifically, we should choose the least developed unification, i.e. the one that is most abstract and is, in terms of development, closest to the original descriptions. In both the above examples this will give the required result. In the LOTOS example P_2 itself should have been chosen as the unification of P_1 and P_2 as it is the least reduced unification, up to testing equivalence. Similarly, in the Z example either of the identical specifications S_1 or S_2 should have been chosen initially. The issue is that we could choose a unification of two descriptions that is too developed to be reconciled with a third description, while a less developed unification that could be reconciled, exists. The problem is evolving the two original specifications unnecessarily far towards the concrete during unification. We will consider this issue of least developed unifications in the next section.

1.3 Least Developed Unifications

We seek an interpretation of the least developed unification. Our interpretation should be a generalisation of the more familiar concept of a least refinement, which generalises to least development in our notation.

Definition 4 Least Development
Y is a least development of X, by dv, iff $Y \ dv \ X$ *and* $\neg(\exists Y' \not\asymp Y \text{ s.t. } Y' \ dv \ X \ \wedge \ Y \ dv \ Y')$.

where $\asymp$ is the notion of equivalence employed in the development framework being considered. So, a least development of X is a development, Y, of X that has no ancestors by dv that are developments of X. Note that another way of looking at the least development of X is that it is a maximal element in the set of possible developments of X. Thus, by reversing the point of reference we can exchange least for maximal. At some points in the text it will be most convenient to make this reversal and talk in terms of maximal elements of sets of developments.

In order to obtain a rich enough theory to work with we will have to put some immediate constraints on development. Firstly, we assume all our development relations are reflexive. This is a natural requirement, although, as we have indicated earlier it can be problematic for inter language consistency. We will say more about the position of inter language consistency shortly.

In addition to reflexivity, we will assume transitivity of development. This is slightly restrictive as it rules out implementation relations (e.g. LOTOS **conf**), but it seems necessary in order to obtain a rich enough theory. Furthermore as we have indicated earlier, this section is motivated by the search for incremental development strategies and transitivity of development seems a prerequisite of such incremental evolution of specifications. In particular without transitivity, we may develop a specification A into a specification B and then evolve B into C and find that C is not a development of A. So, the remaining work in this chapter assumes transitivity and reflexivity of the development relations used, i.e. they are preorders.

We must also consider what interpretation of equivalence, $\asymp$, we should adopt. A natural, and standard, interpretation is:-

$X \asymp_{dv} X'$ iff $X\ dv\ X' \ \wedge\ X'\ dv\ X$

Thus, two descriptions are equivalent if and only if they are both developments of the other. With transitivity of dv this interpretation gives us that two specifications in any cycle by the relation dv are equivalent. It is easy to see that $\asymp_{dv}$ is an equivalence; it will play the role of identity in our theory.

Proposition 5
$\asymp_{dv}$ is an equivalence.

Proof
Reflexivity: $X\ dv\ X \ \wedge\ X\ dv^{-1}\ X$, by reflexivity of dv. Therefore, $X \asymp_{dv} X$.
Symmetry: $X \asymp_{dv} X' \implies (X\ dv\ X' \ \wedge\ X'\ dv\ X) \implies X' \asymp_{dv} X$.
Transitivity: $(X \asymp_{dv} X' \ \wedge\ X' \asymp_{dv} X'') \implies ((X\ dv\ X' \ \wedge\ X'\ dv\ X) \ \wedge\ (X'\ dv\ X'' \ \wedge\ X''\ dv\ X')) \implies ((X\ dv\ X' \wedge X'\ dv\ X'') \wedge (X''\ dv\ X' \wedge X'\ dv\ X)) \implies (X\ dv\ X'' \wedge X''\ dv\ X)$ (by transitivity of dv) $\implies X \asymp_{dv} X''$, as required. □

We can also see the following:-

Proposition 6
dv is a partial order with identity $\asymp_{dv}$.

Proof
Reflexivity, transitivity and antisymmetry all follow by definition. □

We use the following notation in the next proposition:

Definition 5 *For $X \in DES$ and $dv \in DEV$,*

$$D(X, dv) = \{X' \ :\ X'\ dv\ X\}.$$

So, $D(X, dv)$ is the set of all developments of X by dv.

Another expected property of equivalence is expressed in the next proposition. It states that two descriptions have identical development sets, i.e. every description that is a development of one will be a development of the other. Furthermore, this property only arises when the two descriptions are equivalent by $\asymp_{dv}$. This demonstrates that during system development we really can choose any one of a set of equivalent specifications without affecting the possibilities for future development.

Proposition 7
For $dv \in DEV_{ft}$ a preorder and $X, X' \in DES_{ft}$,

$$D(X, dv) = D(X', dv') \iff X \asymp_{dv} X'.$$

Proof
$(\Longrightarrow)$
Firstly, from reflexivity of dv we know that $X \in D(X, dv)$. Thus, from equality of $D(X, dv)$ and $D(X', dv)$ we know that $X \in D(X', dv)$ and hence $X\ dv\ X'$. We can make a similar argument to give $X'\ dv\ X$ and, thus, $X \asymp_{dv} X'$, as required.
$(\Longleftarrow)$
Assume $X \asymp_{dv} X'$ and take $Y \in D(X, dv)$, but $Y \in D(X', dv)$ since $Y\ dv\ X$, $X\ dv\ X'$ and dv is transitive. Thus, $D(X, dv) \subseteq D(X', dv)$ and we can make similar arguments to show that $D(X', dv) \subseteq D(X, dv)$ and, thus, that $D(X, dv) = D(X', dv)$, as required. □

In order to simplify presentation, we will consider *strict development*, i.e. relations $\overline{dv}$ which are subsets of the relations dv with equivalence by $\asymp_{dv}$ factored out.

Definition 6
Overlining is an operation that can be applied to an arbitrary partial order, dv, with the following effect:-

$$\overline{dv} = dv \backslash \asymp_{dv}$$
where $\backslash$ *is set difference, i.e.* $S \backslash T = \{s \in S \mid s \notin T\}$.

$\overline{dv}$ enables us to consider directly the part of dv that excludes identical descriptions by $\asymp_{dv}$. $\overline{dv}$ is strict with regard to dv in the same way that $\subset$ is strict with regard to $\subseteq$. Note in particular that $\overline{dv}$ is not reflexive, as all descriptions are equivalent to themselves. We can now reinterprete least development as:-

$$Y \textit{ is a least development of } X, \textit{ by } dv, \textit{ iff } Y\ dv\ X \textit{ and } \neg(\exists Y' \textit{ s.t. } Y'\ dv\ X \wedge Y\ \overline{dv}\ Y').$$

Least development can be characterised easily:-

Proposition 8
For dv *a preorder,* X *is a least development of itself which is unique up to* $\asymp_{dv}$.

Proof
Firstly, because dv is reflexive, X will be a development of itself. Also, if Y is a development of X and $X\ dv\ Y$ then $X \asymp_{dv} Y$ by definition of $\asymp$. So, there is a no, non equivalent, less developed candidate. Uniqueness of X also follows from this argument. □

So, for dv a preorder the least development can be characterised very easily. Unfortunately, this is not the case when we generalise to least developed unifications. First though, we present our interpretation of least developed unification. We assume dv_i, $1 \leq i \leq n$, are preorders.

Definition 7 (Least Developed Unification)
$X \in \mathcal{U}[dv_1, .., dv_n](X_1, ..., X_n)$ *is a least developed unification iff*

$$\neg(\exists X' \in \mathcal{U}[dv_1, .., dv_n](X_1, ..., X_n), \textit{ s.t. } X\ \overline{\overset{n}{\cap} dv_i}\ X'),$$
$$\textit{where } \overset{n}{\cap} dv_i \textit{ is a shorthand for } dv_1 \cap ... \cap dv_n.$$

This definition ensures that a unification which X is a strict development of does not exist. Notice the interpretation of development, that X and X' are related by $dv_1 \cap ... \cap dv_n$, i.e. the set of unifications is ordered by the intersection of the development relations used in unification. Figure 1.3 depicts a typical situation, X, X' and X'' are unifications of X_1 and X_2 and X, X' and X'' are ordered by $dv_1 \cap dv_2$. In this diagram X is the least developed unification of X_1 and X_2. $dv_1 \cap ... \cap dv_n$ is a natural interpretation of development between unifications because all descriptions in the unification set that are descendents of a least developed unification X are developments of X by all relevant development relations. In addition, the least developed unification concept generalises least development, since, the least developed unification of the check $\mathcal{U}[dv](X)$ clearly corresponds to the least development of X by dv.

Unfortunately, for inter language consistency, the least developed of the set of unifications is a problematic concept. Specifically, descriptions in the unification set, $\mathcal{U}[dv_1, .., dv_n](X_1, ..., X_n)$, are likely to be in a different notation from $X_1, ..., X_n$; thus it is unlikely that the unifications can be related in a type correct manner using $dv_1 \cap .. \cap dv_n$. Thus, this definition and the remaining theory will only be applied to intra language consistency. Ongoing work is addressing generalisation of least developed unification to the inter language setting.

It is also disappointing to discover that for arbitrary development relations (even when constrained to be preorders and in the intra language setting) the least developed unification will not necessarily be unique.

Example 2 *If we have four descriptions;* X_1, X_2, X_3 *and* X_4*; and the development relations between descriptions indicated in figure 1.4, both* X_3 *and* X_4 *are least developed unifications of* X_1

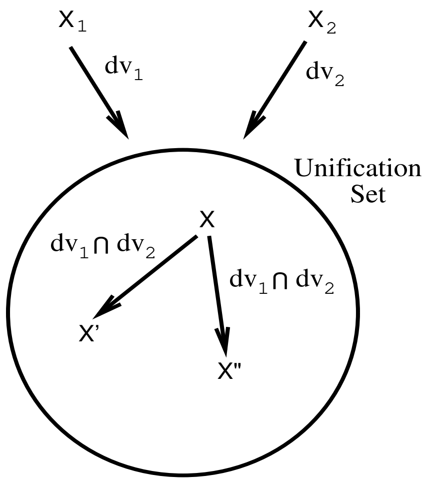

Figure 1.3: A Typical Least Developed Unification Situation

and X_2, i.e. they are clearly both in $\mathcal{U}[dv_1, dv_2](X_1, X_2)$ and neither has an ancestor by $dv_1 \cap dv_2$ in $\mathcal{U}[dv_1, dv_2](X_1, X_2)$. Furthermore, examples of this form are characteristic of situations that foil Π. Specifically, consider the development relations in figure 1.5. In this situation we may unify X_1 and X_2 to X_4 and then fail to find a common development with X_3 even though X_5 could act as the required common development of X_1, X_2 and X_3.

The above example also indicates why least developed unification cannot be characterised as easily as least development. Unification involves reconciling the set of developments of more than one specification. Thus, the least developed unification of $X_1, ..., X_n$ will typically not be one of X_i. This contrasts with the situation for least development where a specification is always its own least development. Consider the situation depicted in figure 1.4, where neither X_1 or X_2 are unifications and are thus, clearly not least developed unifications.

In response to these observations we will divide our discussion of least unification into two parts. First, we will consider the situation in which the least developed unification is not unique then we will discuss the situation in which it is unique. These two cases will be discussed in the following two subsections. In the former case we consider unification according to the set of all least developed unifications. This is a compromise of our ultimate objective which is to locate a single unification, but it allows us to, in general, reduce the specification set to some extend. Our objective is to consider the consequence of using the least developed unification set as unification function. If this gives us the required relationship between Π and C, then we will attempt to be more selective from amongst the least developed unification set and locate under what circumstances we can take just one element from the set.

1.3.1 Non Unique Least Developed Unification

We define the least developed unification set, which we denote $\mathcal{LU}[dv_1, .., dv_n](X_1, ..., X_n)$, as follows:-

Definition 8 (Least Developed Unification Set)
$\mathcal{LU}[dv_1, .., dv_n](X_1, ..., X_n) =$

$$\{X : X \in \mathcal{U}[dv_1, .., dv_n](X_1, ..., X_n) \ \wedge \ \neg(\exists X' \in \mathcal{U}[dv_1, .., dv_n](X_1, ..., X_n), \ s.t. \ X \ \overline{\overset{n}{\cap} dv_i} \ X'\}.$$

Thus, the least developed unification set is the set of all unifications that do not have a non-equivalent ancestor in the unification set. In order to use $\mathcal{LU}$ as the unification function in Π we

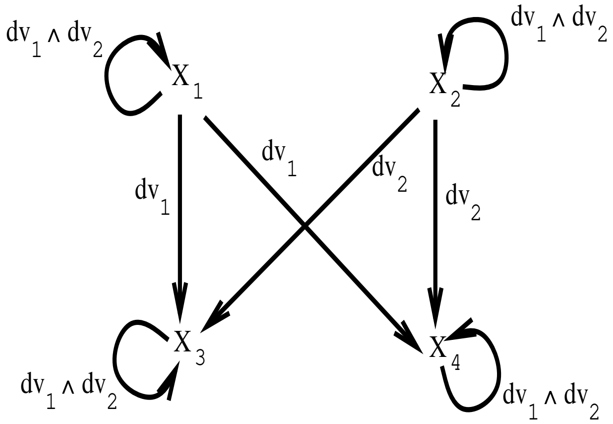

Figure 1.4: Development Relations

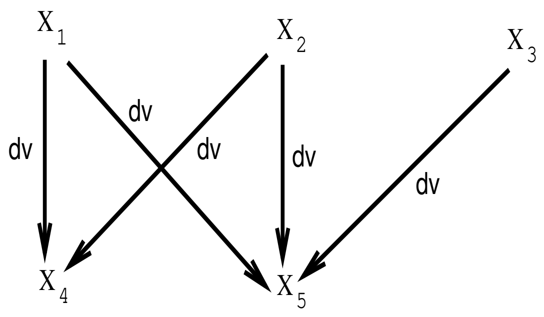

Figure 1.5: More Development Relations

must show that $\mathcal{LU}$ is valid with regard to $\mathcal{U}$, i.e. it satisfies conditions (U.i) and (U.ii). The first of these is straightforward it follows directly from the next proposition.

Proposition 9
$\mathcal{LU}[dv_1, ..., dv_n](X_1, ..., X_n) \subseteq \mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n)$.

Proof
Take $X \in \mathcal{LU}[dv_1, .., dv_n](X_1, .., X_n)$, by the definition of $\mathcal{LU}$ $X \in \mathcal{U}[dv_1, .., dv_n](X_1, .., X_n)$. $\square$

Corollary 2
$\mathcal{LU}[dv, dv'](X, X') \subseteq \mathcal{U}[dv, dv'](X, X')$

Proof
Consequence of proposition 9 with n=2. $\square$

(U.ii) though is more difficult and obtaining this validity constraint is central to showing that $\Pi_{\mathcal{LU}}$ is equal to C. We will have to impose certain "well behavedness" constraints on development in order to obtain this property. With the constraints that we have already imposed on development, i.e. preorder, these properties give us a set of requirements that development in a particular formalism must satisfy in order for it to be used in our framework of unification. In order not to lose the flow of our current argument we will refrain for the moment from consideration of these constraints; they will be discussed in section 1.3.1.1. For the moment we simply state the result that we want; section 1.3.1.1 will provide proofs. We actually need a stronger property than (U.ii) in order to prove the forthcoming theorem, 1. The property that we need is:-

Property 1
$X \in \mathcal{U}[dv_1, .., dv_n](X_1, .., X_n) \implies \exists X' \in \mathcal{LU}[dv_1, .., dv_n](X_1, .., X_n) \; s.t. \; X \overset{n}{\cap} dv_i \; X'$.

This property states that all unifications have an ancestor in the least developed unification set. In other words, all unifications are developments, by $\overset{n}{\cap} dv_i$, of a least developed unification. Notice, a least developed unification is a development of itself. Further notice, implicit in the condition of the unification $\mathcal{U}[dv_1, .., dv_n](X_1, .., X_n) \neq \emptyset$. You may think that such a requirement would naturally hold, but section 1.3.1.1 shows that this is not the case. Once we have property 1 we can easily obtain (U.ii); it will arise as a corollary of the following more general result:-

Proposition 10
Property 1 $\implies$
$$\mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n) \neq \emptyset \implies \mathcal{LU}[dv_1, ..., dv_n](X_1, ..., X_n) \neq \emptyset.$$

Proof
Assume $\mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n) \neq \emptyset$ and take $X \in \mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n)$. Now we can use property 1 to get $\exists X' \in \mathcal{LU}[dv_1, ..., dv_n](X_1, ..., X_n)$. So, $\mathcal{LU}[dv_1, ..., dv_n](X_1, ..., X_n) \neq \emptyset$, as required. $\square$

Corollary 3
Property 1 $\implies$
$$\mathcal{U}[dv, dv'](X, X') \neq \emptyset \implies \mathcal{LU}[dv, dv'](X, X') \neq \emptyset.$$

Proof
This follows from proposition 10 with n=2. $\square$

We now have enough theory to tackle the main concern of this section; obtaining global consistency from binary consistency checking. First, though, there is still the issue of whether the least developed unification is always representative. For standard development relations you would definitely expect this to be the case, since contradictions contained in the unification will reflect contradictions occuring in the original specifications and will not have been introduced during development. Thus, we introduce the following notation and from now on assume that all least developed unifications are representative.

Definition 9 *U is called a representative strategy iff $\forall X \in U[dv_1, ..., dv_n](X_1, ..., X_n)$, X is a representative unification.*

Assuming $\mathcal{LU}$ is a representative strategy simplifies the proceding theory greatly, since it means we do not have to worry about internal validity.

Before we give a full relationship between global consistency and binary consistency checking, we present some sub results which give an associativity property.

Proposition 11
Given property 1,
(i) $\exists Y_1 \in \mathcal{LU}[dv_1, dv_2](X_1, X_2) \ \wedge \ \exists Y_2 \in \mathcal{LU}[dv_1 \cap dv_2, dv_3](Y_1, X_3)$
$\Longleftrightarrow$
(ii) $\exists Y \in \mathcal{LU}[dv_1, dv_2, dv_3](X_1, X_2, X_3)$.

Proof
($\Longrightarrow$)
From (i) and transitivity of development we get $Y_2 \in \mathcal{U}[dv_1, dv_2, dv_3](X_1, X_2, X_3)$ which, by (U.ii), gives us $\exists Y \in \mathcal{LU}[dv_1, dv_2, dv_3](X_1, X_2, X_3)$.
($\Longleftarrow$)
$Y \in \mathcal{LU}[dv_1, dv_2, dv_3](X_1, X_2, X_3) \implies Y \in \mathcal{U}[dv_1, dv_2, dv_3](X_1, X_2, X_3) \implies Y \in \mathcal{U}[dv_1, dv_2](X_1, X_2)$ (by corollary 1). Therefore we can apply property 1 to get $\exists Y' \in \mathcal{LU}[dv_1, dv_2](X_1, X_2)$ such that $Y \ dv_1 \cap dv_2 \ Y'$, but also $Y \ dv_3 \ X_3$ by our assumption, so, $Y \in \mathcal{U}[dv_1 \cap dv_2, dv_3](Y', X_3)$. Now we can use (U.ii) to get that $\exists Y'' \in \mathcal{LU}[dv_1 \cap dv_2, dv_3](Y', X_3)$. So, Y' and Y'' are the required least developed unifications (i.e. Y' for Y_1 and Y'' for Y_2) to give us our result. □

Proposition 12
Given property 1,
(i) $\exists Y_1 \in \mathcal{LU}[dv_2, dv_3](X_2, X_3) \ \wedge \ \exists Y_2 \in \mathcal{LU}[dv_1, dv_2 \cap dv_3](X_1, Y_1)$
$\Longleftrightarrow$
(ii) $\exists Y \in \mathcal{LU}[dv_1, dv_2, dv_3](X_1, X_2, X_3)$.

Proof
($\Longrightarrow$)
From (i) and transitivity of development we get $Y_2 \in \mathcal{U}[dv_1, dv_2, dv_3](X_1, X_2, X_3)$ which, by (U.ii), gives us $\exists Y \in \mathcal{LU}[dv_1, dv_2, dv_3](X_1, X_2, X_3)$.
($\Longleftarrow$)
$Y \in \mathcal{LU}[dv_1, dv_2, dv_3](X_1, X_2, X_3) \implies Y \in \mathcal{U}[dv_1, dv_2, dv_3](X_1, X_2, X_3) \implies Y \in \mathcal{U}[dv_2, dv_3](X_2, X_3)$ (by corollary 1). Therefore we can apply property 1 to get $\exists Y' \in \mathcal{LU}[dv_2, dv_3](X_2, X_3)$ such that $Y \ dv_2 \cap dv_3 \ Y'$, but also $Y \ dv_1 \ X_1$ by our assumption, so, $Y \in \mathcal{U}[dv_1, dv_2 \cap dv_3](X_1, Y')$. Now we can use (U.ii) to get that $\exists Y'' \in \mathcal{LU}[dv_1, dv_2 \cap dv_3](X_1, Y')$. So, Y' and Y'' are the required least developed unifications. □

Corollary 4
Given property 1,
$\exists Y_1 \in \mathcal{LU}[dv_1, dv_2](X_1, X_2) \ \wedge \ \exists Y_2 \in \mathcal{LU}[dv_1 \cap dv_2, dv_3](Y_1, X_3)$
$\Longleftrightarrow$
$\exists Y_1 \in \mathcal{LU}[dv_2, dv_3](X_2, X_3) \ \wedge \ \exists Y_2 \in \mathcal{LU}[dv_1, dv_2 \cap dv_3](X_1, Y_1)$
$\Longleftrightarrow$
$\exists Y \in \mathcal{LU}[dv_1, dv_2, dv_3](X_1, X_2, X_3)$.

Proof
Immediate from previous results, i.e. propositions 11 and 12. □

This is the sort of result that we are looking for it gives equivalence between a binary least developed unification strategy and global least developed unification. It gives us an associativity result for binary unification strategies.

The main result of this subsection is given in the following theorem:-

Theorem 1
Given property 1,
$\exists X \in \mathcal{U}[dv_1, ..., dv_m](X_1, ..., X_m) \implies \overline{\Pi}_{\mathcal{LU}}[dv_1, ..., dv_m](X_1, ..., X_m)$
where
$\overline{\Pi}_{\mathcal{LU}}[dv_1, ..., dv_m](X_1, ..., X_m) =$
$((\exists Y_1 \in \mathcal{LU}[dv_1, dv_2](X_1, X_2) \ \wedge \ X \in \mathcal{U}[dv_1, dv_2](X_1, X_2)) \ \wedge$
$(\exists Y_2 \in \mathcal{LU}[dv_1 \cap dv_2, dv_3](Y_1, X_3) \ \wedge \ X \in \mathcal{U}[dv_1 \cap dv_2, dv_3](Y_1, X_3)) \ \wedge$
$(\exists Y_3 \in \mathcal{LU}[dv_1 \cap dv_2 \cap dv_3, dv_4](Y_2, X_4) \ \wedge \ X \in \mathcal{U}[dv_1 \cap dv_2 \cap dv_3, dv_4](Y_2, X_4)) \ \wedge$
$\ldots$
$\ldots$
$(\exists Y_{m-2} \in \mathcal{LU}[dv_1 \cap .... \cap dv_{m-2}, dv_{m-1}](Y_{m-3}, X_{m-1}) \wedge X \in \mathcal{U}[dv_1 \cap .... \cap dv_{m-2}, dv_{m-1}](Y_{m-3}, X_{m-1})) \ \wedge$
$(\exists Y_{m-1} \in \mathcal{LU}[dv_1 \cap .... \cap dv_{m-1}, dv_m](Y_{m-2}, X_m) \ \wedge \ X \in \mathcal{U}[dv_1 \cap .... \cap dv_{m-1}, dv_m](Y_{m-2}, X_m)))$.
Notice that we are not considering Π *directly, rather we consider the unification strategy* $\overline{\Pi}$ *which adds a second condition on every step of the algorithm. This condition states that* X*, the original unification, is in the unification set relevant to that step. Carrying this condition will simplify the induction proof that we perform and clearly gives us a stronger result than we actually need. We will relate to* Π *as a corollary to this theorem.*

Proof
We prove this result using induction on the number of descriptions (and hence development relations) that are considered, i.e. induction on m above. We will prove a number of base cases in order to indicate the pattern of the proof. This pattern is reflected in the proof of the inductive step.
Base Case 1, m=2.
Notice $m = 1$ does not exist (although a trivial formulation could be given). We wish to prove:
(As) $\exists X \in \mathcal{U}[dv_1, dv_2](X_1, X_2) \implies$ ((a) $\exists Y_1 \in \mathcal{LU}[dv_1, dv_2](X_1, X_2) \wedge$
(b) $X \in \mathcal{U}[dv_1, dv_2](X_1, X_2)$)
This is straightforward. Firstly, (b) follows immediately from our assumption, (As), then (a) is a direct consequence of (b) from (U.ii).
Base Case 2, m=3.
We wish to prove:
(As) $\exists X \in \mathcal{U}[dv_1, dv_2, dv_3](X_1, X_2, X_3) \implies$
((a) $\exists Y_1 \in \mathcal{LU}[dv_1, dv_2](X_1, X_2) \ \wedge$ (b) $X \in \mathcal{U}[dv_1, dv_2](X_1, X_2) \wedge$
(c) $\exists Y_2 \in \mathcal{LU}[dv_1 \cap dv_2, dv_3](Y_1, X_3) \ \wedge$ (d) $X \in \mathcal{U}[dv_1 \cap dv_2, dv_3](Y_1, X_3)$)

Firstly, by observing that from corollary 1 $X \in \mathcal{U}[dv_1, dv_2, dv_3](X_1, X_2, X_3)$ implies that $X \in \mathcal{U}[dv_1, dv_2](X_1, X_2)$ we can reproduce the argument of base case 1 to obtain (a) and (b).

Now from (a) and (b) we can use property 1 to get $\exists Y_1' \in \mathcal{LU}[dv_1, dv_2](X_1, X_2)$ such that $X \ (dv_1 \cap dv_2) \ Y_1'$ and since $X \ dv_3 \ X_3$ from our assumption, (As), we have $X \in \mathcal{U}[dv_1 \cap dv_2, dv_3](Y_1', X_3)$ which gives us (d) and then we can use (U.ii) to get $\exists Y_2 \in \mathcal{LU}[dv_1 \cap dv_2, dv_3](Y_1', X_3)$, i.e. (c). This completes the verification of base case 2.

Inductive Step.
We wish to prove that: proposition (1) $\implies$ proposition (2), where,
Proposition (1) states:
(As.i) $\exists X \in \mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n) \implies$
((1.1) $\exists Y_1 \in \mathcal{LU}[dv_1, dv_2](X_1, X_2) \ \wedge \ X \in \mathcal{U}[dv_1, dv_2](X_1, X_2) \ \wedge$
(1.2) $\exists Y_2 \in \mathcal{LU}[dv_1 \cap dv_2, dv_3](Y_1, X_3) \ \wedge \ X \in \mathcal{U}[dv_1 \cap dv_2, dv_3](Y_1, X_3) \ \wedge$
$\ldots$

...

...

(1.n-2) $\exists Y_{n-2} \in \mathcal{LU}[dv_1 \cap ... \cap dv_{n-2}, dv_{n-1}](Y_{n-3}, X_{n-1}) \wedge X \in \mathcal{U}[dv_1 \cap ... \cap dv_{n-2}, dv_{n-1}](Y_{n-3}, X_{n-1}) \wedge$
(1.n-1) $\exists Y_{n-1} \in \mathcal{LU}[dv_1 \cap ... \cap dv_{n-1}, dv_n](Y_{n-2}, X_n) \;\wedge\; X \in \mathcal{U}[dv_1 \cap ... \cap dv_{n-1}, dv_n](Y_{n-2}, X_n))$

Proposition (2) states:
(As.ii) $\exists X \in \mathcal{U}[dv_1, ..., dv_{n+1}](X_1, ..., X_{n+1}) \implies$
((2.1) $\exists Y_1 \in \mathcal{LU}[dv_1, dv_2](X_1, X_2) \;\wedge\; X \in \mathcal{U}[dv_1, dv_2](X_1, X_2) \;\wedge$
(2.2) $\exists Y_2 \in \mathcal{LU}[dv_1 \cap dv_2, dv_3](Y_1, X_3) \;\wedge\; X \in \mathcal{U}[dv_1 \cap dv_2, dv_3](Y_1, X_3) \;\wedge$

...

...

...

(2.n-2) $\exists Y_{n-2} \in \mathcal{LU}[dv_1 \cap ... \cap dv_{n-2}, dv_{n-1}](Y_{n-3}, X_{n-1}) \wedge X \in \mathcal{U}[dv_1 \cap ... \cap dv_{n-2}, dv_{n-1}](Y_{n-3}, X_{n-1}) \wedge$
(2.n-1) $\exists Y_{n-1} \in \mathcal{LU}[dv_1 \cap ... \cap dv_{n-1}, dv_n](Y_{n-2}, X_n) \;\wedge\; X \in \mathcal{U}[dv_1 \cap ... \cap dv_{n-1}, dv_n](Y_{n-2}, X_n) \;\wedge$
(2.n) $\exists Y_n \in \mathcal{LU}[dv_1 \cap ... \cap dv_n, dv_{n+1}](Y_{n-1}, X_{n+1}) \;\wedge\; X \in \mathcal{U}[dv_1 \cap ... \cap dv_n, dv_{n+1}](Y_{n-1}, X_{n+1}))$

So, assume proposition (1). It is clear that the first n-1 steps of proposition (2), i.e. (2.1), (2.2), ..., (2.n-2), (2.n-1), can be obtained directly from proposition (1). So, we need that proposition (1) and assumption (As.ii) imply (2.n). We know, $\exists Y_{n-1} \in \mathcal{LU}[dv_1 \cap ... \cap dv_{n-1}, dv_n](Y_{n-2}, X_n)$ and $X \in \mathcal{U}[dv_1 \cap ... \cap dv_{n-1}, dv_n](Y_{n-2}, X_n)$ from (1.n-1), so we can use property 1 to get that $\exists Y'_{n-1} \in \mathcal{LU}[dv_1 \cap ... \cap dv_{n-1}, dv_n](Y_{n-2}, X_n)$ such that $X \ (dv_1 \cap ... \cap dv_{n-1}) \cap dv_n \ Y'_{n-1}$, which implies that $X \in \mathcal{U}[dv_1 \cap ... \cap dv_n, dv_{n+1}](Y'_{n-1}, X_{n+1})$ since $X \ dv_{n+1} \ X_{n+1}$ from (As.ii). This gives us the second half of (2.n) and the first half follows directly from (U.ii).

By the principle of mathematical induction, the result follows. □

We are now in a position to relate C to $\Pi_{\mathcal{LU}}$.

Corollary 5
Given property 1 and $\mathcal{LU}$ a representative unification strategy,

$$C[dv_1, ..., dv_n](X_1, ..., X_n) \implies \Pi_{\mathcal{LU}}[dv_1, ..., dv_n](X_1, ..., X_n)$$

Proof
Theorem 1 gives us that $C[dv_1, ..., dv_n](X_1, ..., X_n) \implies \overline{\Pi}_{\mathcal{LU}}[dv_1, ..., dv_n](X_1, ..., X_n)$, however, from an examination of the conditions of $\overline{\Pi}$, if $\mathcal{LU}$ is representative, $\overline{\Pi}_{\mathcal{LU}}[dv_1, ..., dv_n](X_1, ..., X_n) \implies \Pi_{\mathcal{LU}}[dv_1, ..., dv_n](X_1, ..., X_n)$. □

Corollary 6 $C[dv_1, ..., dv_n](X_1, ..., X_n) = \Pi_{\mathcal{LU}}[dv_1, ..., dv_n](X_1, ..., X_n)$

Proof
Immediate from corollary 5 and proposition 3. □

1.3.1.1 Constraints on Development

The difficulty surrounding constraint (U.ii) is that the chain of candidate least unifications may be infinite, as depicted in figure 1.6 and a maximal member of the chain, Y_i, may not exist. This is unlikely to arise in practice, but, is theoretically possible for arbitrary preorders. Notice that it is certain that the unification set can be infinite, e.g. consider the LOTOS **ext** relation. We would like to locate a constraint on development that prevents the unification set being infinitely increasing in the manner highlighted. As indicated earlier the property that we require is stronger than just (U.ii) it is property 1, i.e.

$$X \in \mathcal{U}[dv_1, .., dv_n](X_1, .., X_n) \implies \exists X' \in \mathcal{LU}[dv_1, .., dv_n](X_1, .., X_n) \; s.t. \; X \overset{n}{\cap} dv_i \; X'.$$

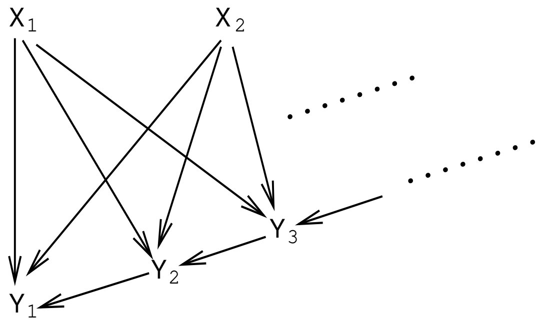

Figure 1.6: Infinite chain of candidate 'least' unifications

The property states that if the unification set is non-empty all unifications are descendents of a least developed unification.

In order to characterise when property 1 can be obtained we need some definitions.

Definition 10 *For $S \subseteq DES$ and $dv \in DEV$,*

$$M(S, dv) = (\exists Y \in S \ s.t. \ \neg(\exists Y' \in S \ s.t. \ Y \ \overline{dv} \ Y')).$$

Such a Y is called a maximal element of S.

Thus, $M(S, dv)$ will hold if and only if the set S of descriptions has a maximal element by dv, i.e. an element, Y, which has no ancestor by dv in S. When we are considering maximal elements of unification sets we will talk about *maximal unifications.*

The next two definitions are interpretations of standard mathematical concepts, see for example [3].

Definition 11 *An infinite set of descriptions $\{X_1, X_2, ...\}$ is said to be an infinitely ascending chain $X_1, X_2, ...$ according to dv iff $X_i \ \overline{dv} \ X_{i+1}$ for all $i \in \mathbb{N}$.*

Definition 12 (Well Founded Set)
S is called a well founded (WF) set by dv iff $\forall S' \subseteq S$, $(S' \neq \emptyset \implies M(S', dv))$.

Thus, a partial order (S, dv) is well founded (WF) if and only if all non-empty subsets of S have at least one maximal element. Clearly, we could consider dual definitions which consider the opposite direction of the development partial order, e.g. minimal elements of ancestors by development. However, our focus is on evolution of descriptions towards development.

Notice that a maximal element of a set is not necessarily unique. There could be a number of unifications with no ancestor by development in the unification set, see for example figure 1.4.

The following is a standard result from mathematical set theory, see [3] for example.

Proposition 13
(i) (S, dv) is well founded.

$$\iff$$

(ii) There is no infinitely ascending chain in (S, dv).

Proof
(i $\implies$ ii) By contradiction, so, assume (i). Now $\neg$(ii) implies that there is an infinite chain in S, i.e. $T = \{X_1, X_2, ...\}$ such that $X_1 \ \overline{dv} \ X_2 \ \wedge \ X_2 \ \overline{dv} \ X_3 \ \wedge \ ....$ Clearly, $T \subseteq S$ and $\neg M(T, dv)$, since all elements in T have an ancestor by dv in T, which contradicts our assumption of (i), as

required.

(i $\Longleftarrow$ ii) By contradiction again, so, assume (ii). Now $\neg$(i) gives us that (S, dv) is not well founded, i.e. $\exists T \subseteq S$ such that $T \neq \emptyset$ and $\neg M(T, dv)$. With this we can construct an infinitely ascending chain as follows:-

1. Select an arbitrary $X_0 \in T$. This will exists as $T \neq \emptyset$.

2. Select $X_1 \in T$ such that $X_0 \ \overline{dv}\ X_1$. Such an X_1 must exist otherwise X_0 would be a maximal element and would contradict $\neg M(T, dv)$.

3. If $X_0, X_1, ..., X_j \in T$ for $j \geq 0$, such that $X_0 \ \overline{dv}\ X_1 \ \wedge \ X_1 \ \overline{dv}\ X_2 \ \wedge \ ... \ \wedge \ X_{j-1} \ \overline{dv}\ X_j$, have already been chosen, then a description X_{j+1} such that $X_j \ \overline{dv}\ X_{j+1}$ can be found. Such an X_{j+1} must exist otherwise X_j would be a maximal element and would contradict $\neg M(T, dv)$.

This construction will generate an infinite ascending chain by $\overline{dv}$ of descriptions $X_0 \ \overline{dv}\ X_1 \ \wedge \ X_1 \ \overline{dv}\ X_2 \ \wedge \ ... \in T$, which contradict our assumption of (ii) as required. $\square$

With these concepts we can characterise under what circumstances property 1 can be obtained.

Proposition 14

(i) There is no infinite ascending chain by $\overline{\overset{n}{\sqcap} dv_i}$ *of descriptions in* $\mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n)$.

$\Longrightarrow$

(ii) $X \in \mathcal{U}[dv_1, .., dv_n](X_1, .., X_n) \implies \exists X' \in \mathcal{LU}[dv_1, .., dv_n](X_1, .., X_n)$ *s.t.* $X \ \overset{n}{\sqcap} dv_i \ X'$.

i.e. (i) $\Longrightarrow$ *property 1.*

Proof

By contradiction, so, assume (i). Now $\neg$(ii) gives:

$\exists X \in \mathcal{U}[dv_1, .., dv_n](X_1, .., X_n)$ *s.t.* $\neg(\exists X' \in \mathcal{LU}[dv_1, .., dv_n](X_1, .., X_n)$ *s.t.* $X \ \overset{n}{\sqcap} dv_i \ X')$.

Now consider the following construction:-

1. $X_0 = X$.

2. Select $X_1 \in \mathcal{U}[dv_1, .., dv_n](X_1, .., X_n)$ such that $X_0 \ \overline{\overset{n}{\sqcap} dv_i}\ X_1$. Such an X_1 must exist, otherwise X_0 would be a least developed unification and a development by $\overset{n}{\sqcap} dv_i$ of itself, which contradicts out assumption of $\neg$(ii).

3. If $X_0, X_1, ..., X_j \in \mathcal{U}[dv_1, .., dv_n](X_1, .., X_n)$ for $j \geq 0$, such that $X_0 \ \overline{\overset{n}{\sqcap} dv_i}\ X_1 \wedge X_1 \ \overline{\overset{n}{\sqcap} dv_i}\ X_2 \wedge ... \wedge X_{j-1} \ \overline{\overset{n}{\sqcap} dv_i}\ X_j$, have already been chosen, then a description X_{j+1} such that $X_j \ \overline{\overset{n}{\sqcap} dv_i}\ X_{j+1}$ can be found. Such an X_{j+1} must exist otherwise X_j would be a least developed unification and by transitivity of development and an ancestor by $\overline{\overset{n}{\sqcap} dv_i}$ of X_0, which would contradict our assumption of $\neg$(ii).

This construction will generate an infinite ascending chain by $\overline{\overset{n}{\sqcap} dv_i}$ of descriptions $X_0, X_1, X_2, ... \in \mathcal{U}[dv_1, .., dv_n](X_1, .., X_n)$, which contradicts our assumption of (i) as required. $\square$

Using this result we can obtain the following important corollary:-

Corollary 7

(i) $(\mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n), \overset{n}{\sqcap} dv_i)$ *is well founded.*

$\Longrightarrow$

(ii) Property 1.

Proof
Immediate from previous two results, proposition 13 and proposition 14. □.

This is a pivotal result, it characterises the properties that are required of $\mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n)$ in order to obtain property 1. In order to use a particular FDT we would actually like to know that any combination of development relations and descriptions in the language will yield a unification set that satisfies, property 1. We will clearly obtain this if an FDT up holds the following:

Property 2
FDT ft satisfies property 2 iff

$\forall X_1, ..., X_n \in DES_{ft} \;\wedge\; \forall dv_1, ..., dv_n \in DEV_{ft}\; (\mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n), \overset{n}{\cap} dv_i)$ *is well founded*

Another way to express property 2 is:

$$\forall X_1, ..., X_n \in DES_{ft} \;\wedge\; \forall dv_1, ..., dv_n \in DEV_{ft},\; (\overset{n}{\cap} D(x_i, dv_i), \overset{n}{\cap} dv_i) \text{ is well founded.}$$

i.e. if the intersection of the development sets of $X_1, ..., X_n$ are always well founded. This is because $\overset{n}{\cap} D(x_i, dv_i) = \mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n)$.

In order verify property 2 we would like to obtain a constraint that can be realistically checked for actual development relations. Thus, we consider a series of "Well Behavedness" properties on development.

The first such well behavedness property states that (i) development sets are well founded and (ii) if two development sets are well founded then the intersection of the development sets is also well founded.

Definition 13 (Well Behaved Condition 1 (WBC1))
For an FDT, ft, we say that development is well behaved (condition 1) iff,

(i) $\forall X \in DES_{ft} \;\wedge\; \forall dv \in DEV_{ft},\; (D(X, dv), dv)$ *is WF.*
(ii) $(D(X, dv), dv) \;\wedge\; (D(X', dv'), dv')$ *are WF* $\Longrightarrow (D(X, dv) \cap D(X', dv'), dv \cap dv')$ *is WF.*

A stronger formulation of the second of these conditions, (WBC1.ii), that may be easier to prove is:

$$\forall Z \in DES_{ft},\; (Z, dv) \;\wedge\; (Z', dv') \text{ are WF} \Longrightarrow (Z \cap Z', dv \cap dv') \text{ is WF.}$$

This condition is stronger since it is defined over all subsets of DES_{ft}, not just the subsets that are development sets by dv and dv'. It should also be clear that from associativity of $\wedge$ and $\cap$, (WBC1.ii) implies:

$$(D(X_1, dv_1), dv_1) \;\wedge .... \wedge\; (D(X_n, dv_n), dv_n) \text{ are WF} \Longrightarrow (\overset{n}{\cap} D(X_i, dv_i), \overset{n}{\cap} dv_i) \text{ is WF.}$$

Now we can show that (WBC1) implies property 2.

Proposition 15
WBC1 $\Longrightarrow$ *property 2.*

Proof
Take $X_1, ..., X_n \in DES_{ft} \;\wedge\; dv_1, ..., dv_n \in DEV_{ft}$, (WBC1.i) gives us that $(D(X_j, dv_j), dv_j)$ is well founded for all j such that $1 \leq j \leq n$. So, now we can apply (WBC1.ii) to get that $(\overset{n}{\cap} D(X_i, dv_i), \overset{n}{\cap} dv_i)$ is well founded, which gives us property 2, as required. □

An alternative is the following condition:-

Definition 14 (Well Behaved Condition 2 (WBC2))
Development is well behaved (condition 2) in FDT ft iff $\forall dv_1, ..., dv_n \in DEV_{ft}$ $(DES_{ft}, \overset{n}{\cap} dv_i)$ *is well founded.*

This states that all non-empty subsets of DES_{ft} have a maximal element by $\overset{n}{\cap} dv_i$. This is clearly a strong condition as it acts over all subsets of DES_{ft} not just those arising from development.

Proposition 16
WBC2 $\Longrightarrow$ *property 2.*

Proof
Immediate since all subsets of $\overset{n}{\cap} D(X_i, dv_i)$ are subsets of DES_{ft}. □

Both WBC1 and WBC2 in some way impose well behavedness constraints on $\overset{n}{\cap} dv_i$, i.e. they require that the intersection of the development relations being used are well behaved in some sense. This focus on the intersection of development relations is not ideal. It would be better if we could check a well behavedness property on each of the development relations individually and not have to consider the interplay of these relations when their intersection is taken. In this way we would be able to check all the development relations individually for a particular FDT and know that we can intersect them as we like. An obvious constraint to consider is well foundedness of constituent development relations, i.e. can we deduce that $\overset{n}{\cap} dv_i$ is well founded if dv_i is well founded for all $1 \leq i \leq n$. This would be a nice result as it would push checking well foundedness out into the constituent development relations. The result we would like is:-

(S, dv) and (S, dv') are well founded $\Longrightarrow$ $(S, dv \cap dv')$ is well founded.

The next result makes a step towards this.

Proposition 17
(i) (S, dv) *and* (S, dv') *are well founded*
$\Longrightarrow$
(ii) S *has no infinite ascending chains by* $\overline{dv} \cap \overline{dv'}$.

Proof
We investigate well foundedness in terms of the existence of infinite chains (proposition 13 justifies this). We will prove the result by contradiction. Thus, we assume (i), i.e. (S, dv) and (S, dv') have no infinite chains, and $\neg$(ii), i.e. S has an infinite chain by $\overline{dv} \cap \overline{dv'}$. So, from $\neg$(ii) we can assume $\exists T \subseteq S$ such that $T = \{X_1, X_2, ...\}$ is an infinite set and $\forall X_i, X_{i+1} \in T$, $X_i \ \overline{dv} \cap \overline{dv'} \ X_{i+1}$. But, the properties of set intersection enable us to deduce that T is an infinite chain on $\overline{dv}$ and on $\overline{dv'}$, which contradicts our assumption of (i), as required. □

In order to obtain the result we require we need to deduce well foundedness of $(S, dv \cap dv')$ from the absence of infinite chains by $\overline{dv} \cap \overline{dv'}$ in S. Notice the distinction between $\overline{dv \cap dv'}$ and $\overline{dv} \cap \overline{dv'}$. The former is strict development by $dv \cap dv'$, while the latter is strict dv and strict dv' intersected. It turns out that we cannot in general deduce well foundedness of $(S, dv \cap dv')$ from the absence of infinite chains by $\overline{dv} \cap \overline{dv'}$ in S. First we need a preparatory result.

Proposition 18
$(dv \cap dv')^{-1} = dv^{-1} \cap dv'^{-1}$

Proof
$dv^{-1} \cap dv'^{-1} = \{(Y, X) | (X, Y) \in dv\} \cap \{(Y', X') | (X', Y') \in dv'\} = \{(Y, X) | (X, Y) \in dv \wedge (X, Y) \in dv'\} = \{(Y, X) | (X, Y) \in dv \cap dv'\} = (dv \cap dv')^{-1}$. □

Next we can relate $\overline{dv \cap dv'}$ and $\overline{dv} \cap \overline{dv'}$.

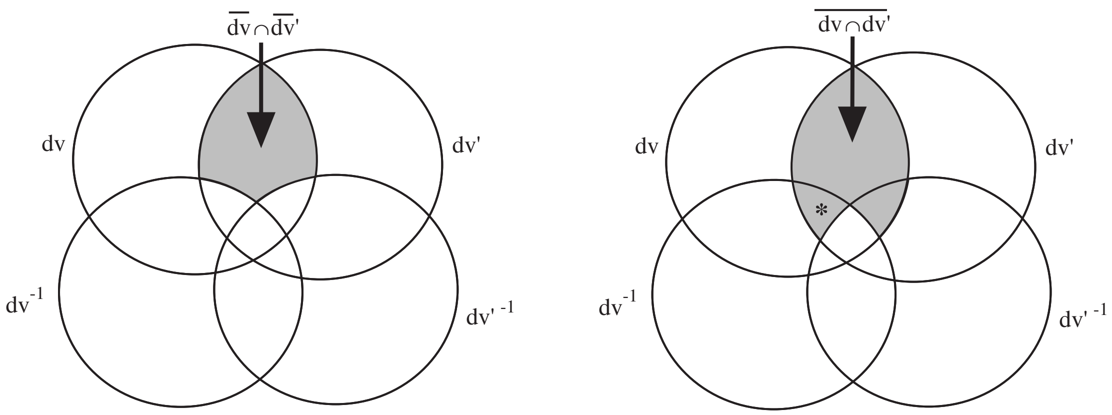

Figure 1.7: Venn diagrams: $\overline{dv} \cap \overline{dv'}$ on left and $\overline{dv \cap dv'}$ on right

Proposition 19
$\overline{dv \cap dv'} \supset \overline{dv} \cap \overline{dv'}$.

Proof
First we will show that $\overline{dv \cap dv'} \supseteq \overline{dv} \cap \overline{dv'}$ and then we will show that, $\overline{dv \cap dv'} \not\subseteq \overline{dv} \cap \overline{dv'}$.
(i) $\overline{dv \cap dv'} \supseteq \overline{dv} \cap \overline{dv'}$
Take X, Y such that $X\ (\overline{dv} \cap \overline{dv'})\ Y$, so,
$X\ dv\ Y\ \wedge\ X\ dv'\ Y\ \wedge\ \neg(X\ dv^{-1}\ Y)\ \wedge\ \neg(X\ dv'^{-1}\ Y) \implies$
$X\ dv \cap dv'\ Y\ \wedge\ \neg(X\ dv^{-1}\ Y\ \vee\ X\ dv'^{-1}\ Y) \implies$
$X\ dv \cap dv'\ Y\ \wedge\ \neg(X\ dv^{-1} \cup dv'^{-1}\ Y) \implies$
$X\ dv \cap dv'\ Y \wedge \neg(X\ dv^{-1} \cap dv'^{-1}\ Y)$ (since if (X, Y) is not in a union of two development relations it cannot be in an intersection of the two relations) $\implies$
$X\ dv \cap dv'\ Y\ \wedge\ \neg(X\ (dv \cap dv')^{-1}\ Y)$ by proposition 18 $\implies X\ \overline{dv \cap dv'}\ Y$.
(ii) $\overline{dv \cap dv'} \not\subseteq \overline{dv} \cap \overline{dv'}$
Take X, Y such that $X\ dv \cap dv' \cap dv^{-1}\ Y$ but, $\neg(X\ dv'^{-1}\ Y)$. From here we can deduce:
$X\ dv \cap dv'\ Y\ \wedge\ (X\ dv^{-1}\ Y\ \wedge\ \neg(X\ dv'^{-1}\ Y)) \implies$
$X\ dv \cap dv'\ Y\ \wedge\ \neg(X\ dv^{-1}\ Y\ \wedge\ X\ dv'^{-1}\ Y) \implies$
$X\ dv \cap dv'\ Y\ \wedge\ \neg(X\ dv^{-1} \cap dv'^{-1}\ Y) \implies$
$X\ dv \cap dv'\ Y\ \wedge\ \neg(X\ (dv \cap dv')^{-1}\ Y)$ by proposition 18 $\implies\ X\ \overline{dv \cap dv'}\ Y$
But, $\neg(X\ \overline{dv} \cap \overline{dv'}\ Y)$ since, $X\ dv\ Y\ \wedge\ X\ dv'\ Y\ \wedge\ X\ dv^{-1}\ Y\ \wedge\ \neg(X\ dv'^{-1}\ Y) \implies$
$(X\ dv\ Y\ \wedge\ X \asymp_{dv} Y)\ \wedge\ (X\ dv'\ Y\ \wedge\ \neg(X \asymp_{dv'} Y)) \implies$
$\neg(X\ \overline{dv}\ Y)\ \wedge\ (X\ \overline{dv'}\ Y) \implies \neg(X\ \overline{dv} \cap \overline{dv'}\ Y)$. So, the pair (X, Y) is in $\overline{dv \cap dv'}$, but is not in $\overline{dv} \cap \overline{dv'}$; the result follows. □

We can illustrate this result with the venn diagrams in figure 1.7 . These diagrams show that $\overline{dv} \cap \overline{dv'}$ and $\overline{dv \cap dv'}$ are different. In particular, we have identified a pair marked * which is in $\overline{dv \cap dv'}$, but is not in $\overline{dv} \cap \overline{dv'}$. Such a pair is used in the second part of the above proof. The issue is that there may be descriptions that are equivalent by a development relation and are thus, not in $\overline{dv}$, but are not equivalent by $\overline{dv \cap dv'}$. So, we cannot obtain a property on $\overline{dv \cap dv'}$ purely from properties on $\overline{dv}$ and $\overline{dv'}$. This is reflected in the next result.

Proposition 20
S has no infinite chains by $\overline{dv} \cap \overline{dv'} \not\Longrightarrow$ *S has no infinite chains by* $\overline{dv \cap dv'}$.

Proof
Assume S has no infinite chains by $\overline{dv} \cap \overline{dv'}$. We will give an example of a chain in S by $\overline{dv \cap dv'}$

that does not invalidate this assumption. Select $T = \{X_1, X_2, ...\}$ as an infinite set of descriptions such that: $\forall X_i, X_{i+1}\ (i \geq 1)\ \ X_i\ dv \cap dv' \cap dv^{-1}\ X_{i+1}$ but, $\neg(X_i\ dv'^{-1}\ X_{i+1})$. From here we can deduce:

$X_i\ dv \cap dv'\ X_{i+1}\ \wedge\ (X_i\ dv^{-1}\ X_{i+1}\ \wedge\ \neg(X_i\ dv'^{-1}\ X_{i+1}))\ \Longrightarrow$
$X_i\ dv \cap dv'\ X_{i+1}\ \wedge\ \neg(X_i\ dv^{-1}\ X_{i+1}\ \wedge\ X_i\ dv'^{-1}\ X_{i+1})\ \Longrightarrow$
$X_i\ dv \cap dv'\ X_{i+1}\ \wedge\ \neg(X_i\ dv^{-1} \cap dv'^{-1}\ X_{i+1})\ \Longrightarrow$
$X_i\ dv \cap dv'\ X_{i+1}\ \wedge\ \neg(X_i\ (dv \cap dv')^{-1}\ X_{i+1})$ by proposition 18 $\Longrightarrow\ X_i\ \overline{dv \cap dv'}\ X_{i+1}$
But, $\neg(X_i\ \overline{dv} \cap \overline{dv'}\ X_{i+1})$ since, $X_i\ dv\ X_{i+1} \wedge X_i\ dv'\ X_{i+1} \wedge X_i\ dv^{-1}\ X_{i+1} \wedge \neg(X_i\ dv'^{-1}\ X_{i+1})\ \Longrightarrow$
$(X_i\ dv\ X_{i+1}\ \wedge\ X_i \asymp_{dv} X_{i+1})\ \wedge\ (X_i\ dv'\ X_{i+1}\ \wedge\ \neg(X_i \asymp_{dv'} X_{i+1}))\ \Longrightarrow$
$\neg(X_i\ \overline{dv}\ X_{i+1})\ \wedge\ (X_i\ \overline{dv'}\ X_{i+1})\ \Longrightarrow\ \neg(X_i\ \overline{dv} \cap \overline{dv'}\ X_{i+1})$. So, T is not an infinite chain for $\overline{dv} \cap \overline{dv'}$, but it is an infinite chain for $\overline{dv \cap dv'}$; the result follows. □

This result is disappointing as it means we cannot obtain well foundedness of unification sets purely from well foundedness of constituent development sets. In order to resolve this difficultly we need to relate equivalence in the constituent development relations with equivalence in $dv \cap dv'$, as performed in the next proposition. The condition of the proposition states that equivalence in either of the development relations implies equivalence in the intersection of the development relations. Thus, it guarantees that unification preserves the equivalence of either constituent development relations. With this condition we can obtain the relationship between $\overline{dv} \cap \overline{dv'}$ and $\overline{dv \cap dv'}$ that we seek.

Proposition 21
Given $X \asymp_{dv} Y\ \vee\ X \asymp_{dv'} Y\ \Longrightarrow\ X \asymp_{dv \cap dv'} Y$*, then,*

(i) S has no infinite chain by $\overline{dv} \cap \overline{dv'}$

$\Longrightarrow$

(ii) S has no infinite chain by $\overline{dv \cap dv'}$.

Proof
By contradiction. So, assume (i) and $\neg$(ii). Select $T = \{X_1, X_2, ...\}$ as an infinite set of descriptions such that: $\forall X_{i+1}, X_i\ (i \geq 1)\ \ X_i\ \overline{dv \cap dv'}\ X_{i+1}$. Therefore:
$X_i\ dv \cap dv'\ X_{i+1}\ \wedge\ \neg(X_i \asymp_{dv \cap dv'} X_{i+1}) \Longrightarrow$
$X_i\ dv\ X_{i+1}\ \wedge\ X_i\ dv'\ X_{i+1}\ \wedge\ \neg(X_i \asymp_{dv} X_{i+1}\ \vee X_i \asymp_{dv'} X_{i+1})$ (by the contrapositive of our given condition) $\Longrightarrow$
$X_i\ dv\ X_{i+1}\ \wedge\ X_i\ dv'\ X_{i+1}\ \wedge\ \neg(X_i \asymp_{dv} X_{i+1})\ \wedge\ \neg(X_i \asymp_{dv'} X_{i+1})$ (by the rules of logic) $\Longrightarrow$
$X_i\ \overline{dv}\ X_{i+1}\ \wedge\ X_i\ \overline{dv'}\ X_{i+1}\ \Longrightarrow\ X_i\ \overline{dv} \cap \overline{dv'}\ X_{i+1}$
i.e. T is an infinite chain in S by $\overline{dv} \cap \overline{dv'}$ which contradicts our assumption of (i), as required. □

The next proposition characterises under what circumstances we can obtain the constraint that we used in proposition 21.

Proposition 22
(i) $(X \asymp_{dv} Y\ \Longleftrightarrow\ X \asymp_{dv'} Y)$
$\Longleftrightarrow$
(ii) $(X \asymp_{dv} Y\ \vee\ X \asymp_{dv'} Y\ \Longrightarrow\ X \asymp_{dv \cap dv'} Y)$.

Proof
$(\Longrightarrow)$
Show $(X \asymp_{dv} Y \vee X \asymp_{dv'} Y\ \Longrightarrow\ X \asymp_{dv \cap dv'} Y)$. Firstly, assume $X \asymp_{dv} Y$, but by (i) this implies $X \asymp_{dv'} Y$, i.e. $X\ dv\ Y \wedge Y\ dv\ X \wedge X\ dv'\ Y \wedge Y\ dv'\ X$, which give us $X\ dv \cap dv'\ Y \wedge Y\ dv \cap dv'\ X$ and hence $X \asymp_{dv \cap dv'} Y$, as required. We can make a similar argument if we assume $X \asymp_{dv'} Y$.
$(\Longleftarrow)$
Assume $X \asymp_{dv} Y$ and try to show that $X \asymp_{dv'} Y$. So, from $X \asymp_{dv} Y$ we can use (ii) to deduce that $X \asymp_{dv \cap dv'} Y$, i.e. $X\ dv \cap dv'\ Y \wedge Y\ dv \cap dv'\ X$. So, $X\ dv\ Y \wedge Y\ dv\ X \wedge X\ dv'\ Y \wedge Y\ dv'\ X$

which gives us that $X \asymp_{dv'} Y$ as required. We can prove $X \asymp_{dv} Y$ from $X \asymp_{dv'} Y$ in a similar way. □

So, if equivalence by one development relation implies equivalence by the other development relation and vice versa then well foundedness of development will yield well foundedness of the unification set. This may again seem a strong constraint, but it has a practical justification. For instance, consider the LOTOS refinement relations **ext** and **red**, although, they are quite different relations they induce the same equivalence, **te**, i.e. $P \asymp_{\mathbf{ext}} Q \iff P \asymp_{\mathbf{red}} Q \iff P \ \mathbf{te}\ Q$. Thus, this theory will help us to obtain wellfoundedness of $\mathcal{U}[\mathbf{ext}, \mathbf{red}]$. We summarise these results in the following well behavedness property.

Definition 15 (Well Behaved Condition 3 (WBC3))
Development is well behaved (condition 3) in FDT, ft, iff

(i) $\forall dv \in DEV_{ft}, dv$ *is well founded.*

(ii) $\forall dv, dv' \in DEV_{ft},\ \asymp_{dv} = \asymp_{dv'}$.

Proposition 23
WBC3 $\Longrightarrow$ *property 2.*

Proof
From propositions 17, 21 and 22 we can deduce property 2. □

So, we have failed to push well behavedness totally out to checks on individual development relations, i.e. we still need to relate equivalence in the distinct development relations. However, the following very strong constraint will succeed in this respect. If development yields a finite development set then property 2 follows. In some circumstances this very strong condition will be sufficient to obtain the result we require.

Definition 16 (Well Behaved Condition 4 (WBC4))
For an FDT, ft, we say development is well behaved (condition 4) iff, $\forall X \in DES_{ft} \ \wedge\ \forall dv \in DEV_{ft},\ D(X, dv)$ *is finite.*

The following simple result will allow us to relate WBC4 and property 2.

Proposition 24
Z is finite and non-empty $\Longrightarrow$ $M(Z, dv)$ *for any dv.*

Proof
We will use induction to show that all finite sets with n elements have a maximal element.

Base Case.
Consider the singleton set $Z_1 = \{X_1\}$. X_1 is trivially a maximal element since it has no ancestors in Z_1.

Induction Step.
Assume $Z_n = \{X_1, ..., X_n\}$ and $M(Z_n, dv)$. Now consider $Z_{n+1} = \{X_1, ..., X_{n+1}\}$ and take $X \in Z_n$ such that $\neg(\exists X' \in Z_n\ s.t.\ X\ \overline{dv}\ X')$, i.e. X is the maximal element which we know exists in Z_n. Now if $\neg(X\ \overline{dv}\ X_{n+1})$ we are done, as X is the required maximal element of Z_{n+1}. So, assume $X\ \overline{dv}\ X_{n+1}$. We will show by contradiction that an X' such that $X_{n+1}\ \overline{dv}\ X'$ cannot exist. In order to do this we assume such an X' exists. Clearly, $X' \in Z_n$ as Z_{n+1} only adds one more element to Z_{n+1} which we have already catered for. But, from our assumptions and transitivity of $\overline{dv}$ we have that $X\ \overline{dv}\ X'$ which contradicts the maximality of X in Z_n and gives us the required contradiction. So, such an X' cannot exist and X_{n+1} is maximal in Z_{n+1}, as required.

The result follows by the principle of mathematical induction. □

Corollary 8
All finite sets are well founded.

Proof
All subsets of a finite set are finite. So, the result follows from proposition 24. □

Using this result we can easily obtain the following:-

Proposition 25
WBC4 $\Longrightarrow$ property 2.

Proof
Clearly, if $D(X_i, dv_i)$ is finite then $\overset{n}{\cap} D(X_i, dv_i)$ is finite. So, we can use the previous corollary, 8, to see that $(\overset{n}{\cap} D(X_i, dv_i), \overset{n}{\cap} dv_i)$ is well founded, as required. □

1.3.2 Unique Least Developed Unification

Clearly, we would like to unify to a single description. So, far we have only considered situations in which we have to test every element of a set of unifications in order to obtain global consistency. Although, the set of least developed unifications is likely to be significantly smaller than the full unification set, it could still be very large. This subsection considers under what circumstances we can safely select any member from the set of least developed unifications and know that further consistency checking and unification with the chosen unification will yield global consistency. In order to do this we need to impose stronger constraints on the unification set. In particular, we must ensure that unification sets possess a *greatest* element.

Definition 17 *An element $X \in S$ is a greatest element of a partially ordered set, (S, dv), iff $\forall X' \in S,\ X'\ dv\ X$. We denote such a greatest element as $g(S, dv)$. If a greatest element does not exist $g(S, dv) = \perp$.*

Clearly we could define the dual notion of a least element. Greatest elements are stronger than maximal elements since for greatest elements all other members of the set must be developments of the greatest element. This is not required with maximal elements for which their may exist elements that are not ancestors or descendents of a maximal element. We have a number of immediate results.

Proposition 26
A greatest element is a maximal element.

Proof
If $g(S, dv)$ is not maximal then $\exists X \in S$ *s.t.* $g(S, dv)\ \overline{dv}\ X$, but by the definition of a greatest element, $X\ dv\ g(S, dv)$, which is a contradiction as $\overline{dv}$ is strict development. □

Corollary 9
If it exists, $g(\overset{1..n}{\mathcal{U}}(dv_i, X_i), \overset{n}{\cap} dv_i) \in \mathcal{LU}[dv_1, ..., dv_n](X_1, ..., X_n)$, i.e. the greatest element is a least developed unification.

Proof
Immediate from proposition 26. □

Proposition 27
A greatest element is unique up to equivalence.

Proof
If X and Y are both greatest elements, then $Y\ dv\ X$ and $X\ dv\ Y$ by the definition of greatest. So, $X \asymp_{dv} Y$, as required. □

We introduce the following obvious notation.

Notation 1
If it exists, we call $g(\overset{1..n}{\mathcal{U}}(dv_i, X_i), \overset{n}{\cap} dv_i)$ *the greatest unification.*

The next result is particularly important as it shows that the existence of a greatest unification is the only circumstance that will yield a unique least developed unification, i.e. the least developed unification is unique up to equivalence if and only if the unification set has a greatest element.

Proposition 28
Assuming property 1
$\forall X, X' \in \overset{1..n}{\mathcal{LU}}(dv_i, X_i),\ X \asymp_{\overset{n}{\cap} dv_i} X'$
$\iff$
$g(\overset{1..n}{\mathcal{U}}(dv_i, X_i), \overset{n}{\cap} dv_i) \neq \perp$.

Proof
$(\Longrightarrow)$
Take $Y' \in \overset{1..n}{\mathcal{U}}(dv_i, X_i)$ as an arbitrary unification. By property 1 every unification is a descendent by $\overset{n}{\cap} dv_i$ of a least developed unification. So, $\exists X \in \overset{1..n}{\mathcal{LU}}(dv_i, X_i)$ such that $Y' \overset{n}{\cap} dv_i\ X$. However, if we select $Y \in \overset{1..n}{\mathcal{LU}}(dv_i, X_i)$ as the required greatest element of $\overset{1..n}{\mathcal{U}}(dv_i, X_i)$, since all pairs of descriptions in $\overset{1..n}{\mathcal{LU}}(dv_i, X_i)$ are equivalent by $\overset{n}{\cap} dv_i$ we know that $X \asymp_{\overset{n}{\cap} dv_i} Y$ and thus by transitivity of development that $Y' \overset{n}{\cap} dv_i\ Y$. So, Y' is a development of our chosen greatest element, as required.
$(\Longleftarrow)$
Take $Y = g(\overset{1..n}{\mathcal{U}}(dv_i, X_i), \overset{n}{\cap} dv_i)$; by corollary 9 we know that $Y \in \overset{1..n}{\mathcal{LU}}(dv_i, X_i)$. If we choose a $Y' \in \overset{1..n}{\mathcal{LU}}(dv_i, X_i)$ clearly $Y' \in \overset{1..n}{\mathcal{U}}(dv_i, X_i)$ so $Y' \overset{n}{\cap} dv_i\ Y$ by our assumption. Now if $\neg(Y \overset{n}{\cap} dv_i\ Y')$ we have a contradiction since Y' would not be a least developed unification (as Y would be a distinct ancestor). Therefore, it must be that $Y \asymp_{\overset{n}{\cap} dv_i} Y'$ and thus all descriptions in $\overset{1..n}{\mathcal{LU}}(dv_i, X_i)$ are equivalent, as required. □

As expected, the property that we will impose on the unification set, in order to allow us to choose any member of the set of least developed unifications, is that it has a greatest element, i.e.

Property 3
If $\overset{1..n}{\mathcal{U}}(dv_i, X_i) \neq \emptyset$ *then* $g(\overset{1..n}{\mathcal{U}}(dv_i, X_i), \overset{n}{\cap} dv_i) \neq \perp$.

We assume the following greatest unification function, $\mathcal{L}$,:

Definition 18
If $g(\overset{1..n}{\mathcal{U}}(dv_i, X_i), \overset{n}{\cap} dv_i) = \perp$ *then* $\overset{1..n}{\mathcal{L}}(dv_i, X_i) = \emptyset$ *otherwise* $\overset{1..n}{\mathcal{L}}(dv_i, X_i) = \{g(\overset{1..n}{\mathcal{U}}(dv_i, X_i), \overset{n}{\cap} dv_i)\}$

So, the function $\mathcal{L}$ returns the empty set if a greatest unification does not exist and a singleton set containing the greatest unification otherwise. Now we need to validate that $\mathcal{L}$ up holds (U.i) and (U.ii). (U.i) will arise as a corollary to the next proposition.

Proposition 29
Given property 3,
$\overset{1..n}{\mathcal{L}}(dv_i, X_i) \subseteq \overset{1..n}{\mathcal{U}}(dv_i, X_i)$.

Proof
If $\overset{1..n}{\mathcal{L}}(dv_i, X_i) = \emptyset$ then the result follows trivially, so, assume $\overset{1..n}{\mathcal{L}}(dv_i, X_i) = \{X\}$. Now clearly $X \in \overset{1..n}{\mathcal{U}}(dv_i, X_i)$ and the result follows. □

Corollary 10
Given property 3,
$\overset{1..2}{\mathcal{L}}(dv_i, X_i) \subseteq \overset{1..2}{\mathcal{U}}(dv_i, X_i)$.

Proof
Immediate from proposition 29 with n=2. $\square$

(U.ii) arises as a corollary of the next result.

Proposition 30
Given property 3,
$\overset{1..n}{\mathcal{U}}(dv_i, X_i) \neq \emptyset \implies \overset{1..n}{\mathcal{L}}(dv_i, X_i) \neq \emptyset$

Proof
Assume $\overset{1..n}{\mathcal{U}}(dv_i, X_i) \neq \emptyset$; we can immediately apply property 3 to get that a greatest element exists and thus that $\overset{1..n}{\mathcal{L}}(dv_i, X_i) \neq \emptyset$, as required. $\square$

We can also consider the equivalent of property 1 for $\mathcal{L}$. This property is stronger than proposition 30.

Property 4
$X \in \overset{1..n}{\mathcal{U}}(dv_i, X_i) \implies X \overset{n}{\cap} dv_i\ Y$ *where* $Y \in \overset{1..n}{\mathcal{L}}(dv_i, X_i)$.

We can see that this property follows directly from the existence of a greatest element.

Proposition 31
Property 3 $\implies$ *property 4.*

Proof
$\overset{1..n}{\mathcal{U}}(dv_i, X_i) \neq \emptyset \implies \overset{1..n}{\mathcal{L}}(dv_i, X_i) \neq \emptyset$, the result follows immediately from the definition of $\mathcal{L}$. $\square$

We will also use the following simple result

Proposition 32
Given property 3,
$Y \in \mathcal{L}[dv, dv'](X, X') \ \wedge\ Y' \in \mathcal{L}[dv, dv', dv''](X, X', X'') \implies Y'\ dv \cap dv'\ Y.$

Proof
Clearly, $Y' \in \mathcal{U}[dv, dv', dv''](X, X', X'')$, but we can use corollary 1 to get $Y' \in \mathcal{U}[dv, dv'](X, X')$ and by the definition of $\mathcal{L}$ we have $Y'\ dv \cap dv'\ Y$, as required. $\square$

We are now in a position to relate binary consistency strategies to global consistency when greatest unifications exist. We seek an associativity result and in order to express this clearly we consider a function β which is derived from $\mathcal{L}$. In β the development relation and description arguments are presented as pairs; i.e. a development relation and the description it is to be applied to are paired as a single argument. The function returns a pair, with first element the intersection of the development relations considered and second element the greatest unification. Notice a bottom element is returned as greatest unification if either a greatest unification does not exist or one of the descriptions given as an argument is undefined.

Definition 19
$$
\begin{aligned}
&\beta(< dv, X >, < dv', X' >) = < dv \cap dv', Y > \\
&\qquad\qquad\qquad \textit{where} \\
&\qquad\qquad\qquad\quad \textit{if } X = \perp \ \vee\ X' = \perp \ \vee\ \mathcal{L}[dv, dv'](X, X') = \emptyset \textit{ then } Y = \perp \\
&\qquad\qquad\qquad\quad \textit{otherwise } Y \in \mathcal{L}[dv, dv'](X, X').
\end{aligned}
$$

We will prove associativity of β by relating the two possible binary bracketings of β to $\mathcal{L}[dv, dv', dv''](X, X', X'')$.

Proposition 33
Given property 3,
$r(\beta(< dv, X >, \beta(< dv', X' >, < dv'', X'' >))) \asymp_{dv \cap dv' \cap dv''} Y$ *where* $Y \in \mathcal{L}[dv, dv', dv''](X, X', X'')$ *and* r *is the right projection function, which yields the second element of a pair.*

Proof
Take $Y = r(\beta(< dv, X >, \beta(< dv', X' >, < dv'', X'' >)))$ and $Y' \in \mathcal{L}[dv, dv', dv''](X, X', X'')$. By transitivity of development $Y \in \mathcal{U}[dv, dv', dv''](X, X', X'')$, so by the definition of $\mathcal{L}$ we get $Y\ dv \cap dv' \cap dv''\ Y'$. Also, let $Y'' = r(\beta(< dv', X' >, < dv'', X'' >))$. By, proposition 32 $Y'\ dv' \cap dv''\ Y''$. Also, $Y' \in \mathcal{U}[dv, dv', dv''](X, X', X'')$ so $Y'\ dv\ X$ and therefore, $Y' \in \mathcal{U}[dv, dv' \cap dv''](X, Y'')$. But, $Y \in \mathcal{L}[dv, dv' \cap dv''](X, Y'')$, so, it is the greatest element in $\mathcal{U}[dv, dv' \cap dv''](X, Y'')$ and thus, $Y'\ dv \cap dv' \cap dv''\ Y$. This gives us $Y\ dv \cap dv' \cap dv''\ Y'$ and $Y'\ dv \cap dv' \cap dv''\ Y$ and thus, $Y \asymp_{dv \cap dv' \cap dv''} Y'$, as required. □

Proposition 34
Given property 3,
$r(\beta(\beta(< dv, X >, < dv', X' >), < dv'', X'' >)) \asymp_{dv \cap dv' \cap dv''} Y$ *where* $Y \in \mathcal{L}[dv, dv', dv''](X, X', X'')$

Proof
Similar to proof of proposition 33. □

Now if we define equality pairwise as,

$$< dv, X > = < dv', X' > \text{ iff } dv = dv' \ \wedge \ X \asymp_{dv \cap dv'} X'$$

the following result is straightforward.

Corollary 11 *Given property 3*
$\beta(< dv, X >, \beta(< dv', X' >, < dv'', X'' >)) = \beta(\beta(< dv, X >, < dv', X' >), < dv'', X'' >)$

Proof
Follows immediately from previous two results, propositions 33 and 34. □

This is a full associativity result which gives us that any bracketing of $\beta(< dv_1, X_1 >, ..., < dv_n, X_n >)$ is equal. Since β is just an alternative coding of $\mathcal{L}$ that facilitates clarity of expression, we have full associativity of $\mathcal{L}$ and that a consistency strategy using $\mathcal{L}$ can be composed of any ordering of binary consistency checks, in particular, $\Pi_{\mathcal{L}} = C$. So, if greatest unifications exist, we can obtain global consistency from any appropriate series of binary consistency checks. This is an important result that arises from a very well behaved class of unification.

1.3.2.1 Constraints on Development

We know that the existence of a greatest unification will allow us to safely choose just one description from the least developed unification set. What conditions can we impose on development in order to obtain the existence of such a greatest element. We will investigate suitable conditions in a similar way to our investigation of maximal elements in section 1.3.1.1.

In a similar way to in section 1.3.1.1 we generalise the condition we require to all possible unifications that can be performed in an FDT.

Property 5 *An FDT, ft, satsifies property 5 iff,*

$$\forall X_1, ..., X_n \in DES_{ft} \wedge \forall dv_1, ..., dv_n \in DEV_{ft},\ g(\mathcal{U}[dv_1, ..., dv_n](X_1, ..., X_n), \overset{n}{\cap} dv_i) \neq \perp.$$

This property ensures that any possible combination of descriptions and development relations in ft will generate a unification set with a greatest element. Satisfaction of this property will guarantee that we can always safely select just one element from the least developed unification set.

The first condition that will ensure property 5 corresponds to WBC1 of section 1.3.1.1.

Definition 20 (Well Behaved Condition a (WBCa))
For an FDT, ft, development is well behaved (condition a) iff

(i) $\forall X \in DES_{ft} \ \wedge \ \forall dv \in DEV_{ft},\ g(D(X,dv),dv) \neq \perp.$

(ii) $g(D(X,dv),dv) \neq \perp \ \wedge \ g(D(X',dv'),dv') \neq \perp \implies g(D(X,dv) \cap D(X',dv'), dv \cap dv') \neq \perp$

Proposition 35
WBCa $\implies$ *property 5.*

Proof
Take $X_1,..,X_n \in DES_{ft} \ \wedge \ dv_1,...,dv_n \in DEV_{ft}$, (WBCa.i) gives us that $g(D(X_j,dv_j),dv_j) \neq \perp$ for all j such that $1 \leq j \leq n$ and we can apply (WBCa.ii) to get that $g(\overset{n}{\cap} D(X_i,dv_i), \overset{n}{\cap} dv_i) \neq \perp$, which gives us property 5, as required. □

The following is an alternative condition that corresponds to WBC2 of section 1.3.1.1.

Definition 21 (Well Behaved Condition b (WBCb))
For an FDT, ft, development is well behaved (condition a) iff $\forall dv_1,..,dv_n \in DEV_{ft}, \forall S \subseteq DES_{ft},\ g(S, \overset{n}{\cap} dv_i) \neq \perp.$

Proposition 36
WBCb $\implies$ *property 5.*

Proof
Immediate, since $\overset{1..n}{\mathcal{U}}(dv_i,X_i) \subseteq DES_{ft}$. □

In a similar way as in section 1.3.1.1 we would also like to derive a property that we can check solely on individual development relations, without having to consider the interplay of these relations on intersection. The following proposition demonstrates that this cannot be easily obtained.

Proposition 37
$\forall S \subseteq DES_{ft}$ *s.t.* $S \neq \emptyset$,

(i) $g(S,dv) \neq \perp$ *and* $g(S,dv') \neq \perp$

$\not\Longrightarrow$

(ii) $g(S, dv \cap dv') \neq \perp.$

Proof
By counterexample. So, assume (i), i.e. $\forall S \subseteq DES_{ft}$ *s.t.* $S \neq \emptyset$, $g(S,dv) \neq \perp$ and $g(S,dv') \neq \perp$. Consider the set $S' \subseteq DES_{ft}$ with two elements, i.e. $S' = \{X_1, X_2\}$ and assume that $X_1\ dv\ X_2$ and $X_2\ dv'\ X_1$ and no other relations hold between X_1 and X_2. This gives us $X_2 = g(S',dv)$ and $X_1 = g(S',dv')$. So, our assumption of (i) is not invalidated, but, $dv \cap dv' = \emptyset$, so both X_1 and X_2 are maximal elements by $dv \cap dv'$ and neither are greatest elements. Thus, $g(S', dv \cap dv') = \perp$, and S' is the required counterexample. □

So, in the same way as we struggled to push well foundedness solely into development we are struggling to push the existence of greatest elements solely into the constituent development relations. The following shows that the strong condition that we finally used to do this in section 1.3.1.1 does not work here.

Proposition 38
S is finite and non-empty $\not\Longrightarrow g(S, dv) \neq \perp$.

Proof
Consider the set S' used as the counterexample in the last propositon, 37; S' is finite but has no greatest element. □

So, enforcing finiteness of development sets cannot guarantee the existence of greatest elements in unification sets either. We are left then with a smaller set of well behavedness properties for this section.

1.4 More Restricted Classes of Consistency Checking

The majority of our work has considered more restricted classes of consistency than this chapter has so far focussed on, in particular, we have, to date, almost exclusively focussed on balanced consistency in our work with Z and LOTOS. So, what happens to the theory considered so far in this chapter in these circumstances? This section then restricts itself to balanced intra language consistency and dv a preorder.

We have a number of preparatory definitions. The following is the standard set theoretic notion of a lower bound of a set.

Definition 22 *$X \in DES_{ft}$ is a lower bound of $Z \subseteq DES_{ft}$ iff $\forall X' \in Z,\ X\ dv\ X'$. The set of all lower bounds of Z is denoted, $lb < Z, dv >$. If a lower bound does not exist $lb < Z, dv >= \emptyset$*

A lower bound of Z is a development of all elements of Z. Notice a lower bound does not have to be a member of Z in contrast to a maximal or greatest element. The dual concept of an upper bound can be similarly defined. It should be clear that for balanced consistency lower bounds correspond to unifications, i.e. $\mathcal{U}[dv](X_1, ..., X_n) = lb < \{X_1, ..., X_n\}, dv >$. In particular, the fact that the ordering of descriptions in balanced unification is unimportant is reflected by the descriptions being interpreted as a set in lb.

In standard fashion we can also define the concept of a greatest lower bound.

Definition 23 *For $Z \subseteq DES_{ft}$ $glb < Z, dv >$ is a lower bound such that all other lower bounds are a development of $glb < Z, dv >$; i.e. $glb < Z, dv > \in lb < Z, dv > \ \wedge (\forall X \in lb < Z, dv >\ X\ dv\ glb < Z, dv >)$. If a greatest lower bound does not exist $glb < Z, dv >= \perp$.*

Once again we can also define the dual concept of a least upper bound. It should again be clear that a greatest lower bound of a set of descriptions is a greatest unification of the descriptions. In particular, note that the ordering of the unification set by $\overset{n}{\sqcap} dv_i$ in the general (unbalanced) case has been collapsed to just dv.

We can now define consistency in this restricted setting:-

Definition 24 $C[dv](X_1, ..., X_n) \Longleftrightarrow \exists X \in lb < \{X_1, ..., X_n\}, dv >\ \ s.t.\ \Psi(X)$.

With this theory we can also simply characterise when all descriptions in an FDT are balanced consistent by dv, i.e. the FDT is completely consistent by dv.

Proposition 39
$\forall Z \subseteq DES_{ft} \wedge dv \in DEV_{ft},\ \exists X \in lb < Z, dv > \ \wedge \Psi(X) \iff \forall X_1, ..., X_n \in DES_{ft},\ C[dv](X_1, ..., X_n)$ *holds.*

Proof
Straightforward. □

i.e. if all subsets of DES_{ft} have a lower bound then all specifications are consistent by dv.

An alternative check for complete consistency is that an internally valid terminal element exists for dv. A development relation dv has a *terminal* or *bottom* element, denoted $\perp_{dv}$, if and only if $\forall X \in DES_{ft},\ \perp_{dv}\ dv\ X$.

Proposition 40
DES_{ft} has an internally valid bottom element $\Longrightarrow \forall X_1, .., X_n \in DES_{ft},\ C[dv](X_1, ..., X_n)$ *holds.*

Proof
Immediate. □

What, in this restricted setting, enables us to obtain global consistency from binary consistency? We would like to locate an equivalent of the existence of greatest unifications. As indicated earlier, the greatest lower bound gives us this equivalent.

Proposition 41
$glb < \{X_1, ..., X_n\}, dv > \neq \perp \Longrightarrow glb < \{X_1, ..., X_n\}, dv > \in \mathcal{L}[dv](X_1, ..., X_n)$.

Proof
By definition. □

So, the property that we require for balanced consistency checking to be performed incrementally is:

Property 6
$\forall \{X_1, ..., X_n\} \subseteq DES_{ft} \wedge \forall dv \in DEV_{ft},\ lb < \{X_1, ..., X_n\}, dv > \neq \emptyset \Longrightarrow glb < \{X_1, ..., X_n\}, dv > \neq \perp$.

This property ensures that if a lower bound exists then a greatest lower bound can be found, i.e. the unification of $X_1, ..., X_n$ is non-empty implies a greatest unification exists. It is clear from the theory of greatest unifications we have presented and from set theory that taking greatest lower bounds is associative, i.e.

$$glb < \{glb < \{X_1, X_2\}, dv >, X_3\}, dv > = glb < \{X_1, glb < \{X_2, X_3\}, dv >\}, dv >$$

With these concepts we can identify what is the most well behaved class of development.

Definition 25 *(DES_{ft}, dv) is cocomplete iff* $\forall S \subseteq DES_{ft},\ glb < S, dv > \neq \perp$.

Cocompleteness is related to the standard concept of a complete partial order, see for example [3], which considers the existence of least upper bounds as opposed to greatest lower bounds in our framework. If development is cocomplete for a particular FDT then all specifications are consistent and we can adopt any relevant incremental consistency checking strategy. All descriptions are consistent since a lower bound exists for all collections of descriptions and incremental consistency checking strategies are well behaved since a single greatest unification always exists.

1.5 Discussion

The results of this chapter are summarised in the following table. In general, the consistency problem is more straightforward and well behaved the further down the table you go.

Class of Consistency			**Implications**
Unbalanced	Inter lang.		No results
Unbalanced	Intra lang.	Not WF unif. set	No incremental cons checking
Unbalanced	Intra lang.	WF unif. set	Set of least developed unifications
Unbalanced	Intra lang.	Greatest unifs.	Unique incremental cons. checking
Balanced	Intra lang.	Not WF unif. set	No incremental cons. checking
Balanced	Intra lang.	WF without glb's	Set of least developed unifications
Balanced	Intra lang.	glbs always exist	Unique incremental cons. checking
Balanced	Intra lang.	Cocomplete	Completely consistent and unique incremental cons. checking

Bibliography

[1] E. Boiten, H. Bowman, J. Derrick, and M. Steen. Cross viewpoint consistency in Open Distributed Processing (intra language consistency). Technical Report 8-95, Computing Laboratory, University of Kent at Canterbury, 1995.

[2] H. Bowman, E.A. Boiten, J. Derrick, and M.W.A. Steen. Strategies for consistency checking, the choice of unification. In *submitted to BCS FACS Group, 7th Refinement Workshop*, Bath, England, July 1996.

[3] J. Loeckx and K. Sieber. *The Foundations of Program Verification*. Wiley, 1984.