Equivalence of Linear, Free, Liberal, Structured Program Schemas is Decidable in Polynomial Time (Technical Report).

Michael R.Laurence [d] Sebastian Danicic [a] Mark Harman [d] Rob Hierons [b] John Howroyd [c]

[a] *Department of Computing Sciences, Goldsmiths College, University of London, New Cross, London SE14 6NW, UK.*

[b] *Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex, UB8 3PH, UK.*

[c] *@UK PLC 5 Jupiter House Calleva Park Aldermaston Berks RG7 8NN*

[d] *Department of Computer Science, King's College London, Strand, London WC2R 2LS, United Kingdom*

Abstract

A program schema defines a class of programs, all of which have identical statement structure, but whose expressions may differ. We prove that given any two linear, free, liberal structured schemas, it is decidable whether they are equivalent. Our main result considerably extends the class of program schemas for which equivalence is known to be decidable, and suggests that linearity is a constraint worthy of further investigation.

Key words: structured program schemas, conservative schemas, liberal schemas, free schemas, linear schemas, schema equivalence, static analysis, program slicing

Contents

List of Figures

1 Introduction

A program schema represents the statement structure of a program by replacing real functions and predicates with function and predicate symbols taken from sets $\mathcal{F}$ and $\mathcal{P}$ respectively. A schema, S, thus defines a whole class $[S]$ of programs all of the same structure. Each program in $[S]$ can be obtained from S via a mapping called an *interpretation* which gives meanings to the function and predicate symbols in S. As an example, Figure 1 gives a schema S; and the program P of Figure 2 is in the class $[S]$. The subject of schemas is connected with that of program transformation and was originally motivated by the wish to compile programs effectively.

$$
\begin{array}{lll}
u := h(); & & \\
\mathit{if}\ p(w) & \mathit{then} & v := f(u); \\
 & \mathit{else} & v := g();
\end{array}
$$

Fig. 1. Schema S

$$
\begin{array}{lll}
u := 1; & & \\
\mathit{if}\ w > 1 & \mathit{then} & v := u + 1; \\
 & \mathit{else} & v := 2;
\end{array}
$$

Fig. 2. Program P

This paper gives a class of schemas for which equivalence is decidable. Equivalence is defined as follows. Given any variable v in a variable set $\mathcal{V}$, we say that schemas S, T are *v-equivalent*[1], written $S \cong_v T$, if given any interpretation and an initial state (that is, a mapping from the set of variables into some fixed domain) the programs defined by S and T give the same final value to the variable v, provided they both terminate. We also define $S \cong_\omega T$ to mean that given any interpretation and given any initial state, the programs defined by S and T either both terminate or both fail to terminate. Thus the schema T of Figure 3 satisfies $S \cong_v T$, with S as in Figure 1; but $S \cong_\omega T$ does not hold. The relation $\cong$ ('equivalence') means the conjunction of the relation $\cong_\omega$ and the relations $\cong_v$ for all variables v. Some researchers use the phrase 'functional equivalence' to refer to the relation $\cong_{\mathcal{V} \cup \{\omega\}}$ and 'weak equivalence' for $\cong_{\mathcal{V}}$.

This definition of equivalence takes no account of relations between the symbols, or requirements that a function or predicate symbol have a certain meaning, although

[1] For the class of *all* schemas the relation $\cong_v$ is not transitive, as an example in Section 5 shows, but it *is* an equivalence relation for the class of free, structured, linear schemas in which we will be working (Proposition 22).

$$
\begin{array}{lll}
\textit{while}\, q(v) & \textit{do} & v := k(v); \\
\textit{if}\ p(w) & \textit{then} & \\
 & & \{ \\
 & & u := h(); \\
 & & v := f(u); \\
 & & \} \\
 & \textit{else} & v := g();
\end{array}
$$

Fig. 3. Schema T

definitions of equivalence for which interpretations are defined in this more restricted way have been considered (see Subsection 2.5).

Traditionally schemas were defined using a set of labelled statements (see Subsection 2.1) or equivalently a flow diagram. In Section 2 we discuss this class of schemas. All results proved in this paper only concern *structured* schemas,[2] in which *goto* statements are forbidden, and predicate symbols are only used to build if statements, of the form $\textit{if}\ q(\mathbf{u})\ \textit{then}\ T_1\ \textit{else}\ T_2$, or while statements, of the form $\textit{while}\ p(\mathbf{u})\ \textit{do}\ T$; where in both cases $\mathbf{u}$ is a finite tuple of variables.

It has been shown that it is decidable whether two structured schemas which are Conservative, Free and Linear (CFL), are equivalent [2]. The main result of this paper, Theorem 148, is a strengthening of this result; that it can be decided in polynomial time whether two structured schemas which are Liberal, Free and Linear (LFL), are equivalent.

1.1 Decidability results for different classes of schemas

Many subclasses of schemas have been defined:

Linear schemas, in which each function and predicate symbol occurs at most once.[3]
Conservative schemas, in which every assignment is of the form $v := f(v_1, \ldots, v_r)$ where $v \in \{v_1, \ldots, v_r\}$.
Free schemas, where all paths are executable under some interpretation.
Liberal schemas, in which two assignments along any legal path can always be made to assign distinct values to their respective variables.

[2] Some authors, for example Manna [1], use the phrase *while* schema for what we call a *structured* schema (except that Manna allows statements like $\textit{while}\ \neg p(\mathbf{u})\ \textit{do}\ T$); in this paper a *while* schema means a structured schema consisting of a while loop (Definition 4).
[3] Some authors use the phrase 'non-repeating schemas' to refer to what we call linear schemas.

The last three of these classes were first introduced by Paterson [3]. Of these conditions, the first two can clearly be decided for the class of all schemas. Paterson [3] also proved, using a reduction from the Post Correspondence Problem, that it is not decidable whether a schema is free. He also showed however that it *is* decidable whether a schema is both liberal and free; and since he also gave an algorithm for transforming a schema S into a schema T such that T is both liberal and free if and only if S is liberal, it is clearly decidable whether a schema is liberal. It is an open problem whether freeness is decidable for the class of linear schemas.

All results on the decidability of equivalence of schemas are either negative or confined to very restrictive classes of schemas. In particular Paterson [3] proved, in effect, that equivalence is undecidable for the class of all (unstructured) schemas. He proved this by showing that the halting problem for Turing machines (which is, of course, undecidable) is reducible to the equivalence problem for the class of all schemas. Ashcroft and Manna showed [4] that an arbitrary schema can be effectively transformed into an equivalent structured schema, provided that statements such as $\textit{while}\,\neg p(\mathbf{u})\,\textit{do}\,T$ are permitted; hence Paterson's result shows that any class of schemas for which equivalence can be decided must not contain this class of schemas. Thus in order to get positive results on this problem, it is clearly necessary to define the relevant classes of schema with great care.

Although the class of linear schemas considered in this paper is a highly restrictive one, it has the merit that linear schemas are the main objects studied in the field of Program Slicing (see Subsection 2.6), and that this is therefore a particularly important class.

1.2 Organisation of the Paper

1.2.1 Section 2

Here we give an overview of the history of schemas, and also give some of the formal definitions which will be used in the rest of the paper. In particular, the definition of an interpretation will be formally given.

1.2.2 Section 3

Here we give further definitions. In particular we formally define the *structured* and linear schemas with which we will be working in the remainder of the paper. We define the syntactic (binary) data dependence relation $\rightsquigarrow_S$ (Definition 25) which defines which terms can be generated by following a path through a linear schema S. We then define the relations $\textit{outwhile}_S$, $\textit{outif}_S$, $\textit{back}_S$ and $\textit{thru}_S$ which strengthen the $\rightsquigarrow_S$

relation (Definition 29). We also define the relation $\ll_S$ which expresses the order in which symbols occur in a path through a linear schema S (Definition 34), and the relation $\searrow_S$, when expresses that fact that a symbol 'lies below' a predicate symbol (Definition 7).

We define (in Definition 35) the set $\mathcal{N}_S(u)$, the set of function and predicate symbols *needed* by $u \in \mathcal{V} \cup \{\omega\} \cup Symbols(S)$ in S; this is the set of symbols in S which might conceivably affect the final value of u (if $u \in \mathcal{V} \cup \{\omega\}$) or the term defined by a path passing through u (if u is a symbol of S). The set $\mathcal{N}_S(u)$ is syntactically defined.

We will then define the syntactic relation $simil_u$, which itself is defined as the conjunction of 13 separate conditions, between linear structured schemas (Definition 38). Most of the paper is devoted to proving that $S \cong_V T \iff S\, simil_V\, T$ holds for any set $V \subseteq \mathcal{V} \cup \{\omega\}$ such that $\omega \in V$, and since $simil$ is defined syntactically, it can easily be decided whether $S\, simil_u\, T$ holds for any $u \in \mathcal{V} \cup \{\omega\}$. Our main Theorem follows from this result.

Informally, for two linear structured schemas S, T to satisfy $S\, simil_u\, T$, the following must hold;

- $\mathcal{N}_S(u) = \mathcal{N}_T(u)$. (Condition (1) of $S\, simil_u\, T$).
- S and T have the same *data dependence* relations among those symbols in $\mathcal{N}_S(u)$. (Conditions (4), (5),(6) of $S\, simil_u\, T$).
- S and T have the same set of if predicates and while predicates, of those lying in $\mathcal{N}_S(u)$ (Conditions (2), (3) of $S\, simil_u\, T$).
- S and T have the same set of symbols lying in the body of each while predicate, of those lying in $\mathcal{N}_S(u)$ (Condition (8) of $S\, simil_u\, T$). If $u = \omega$ a weaker statement also holds for while predicates lying under if predicates (Condition (7) of $S\, simil_u\, T$). Also, the bodies of while predicates in S and T satisfy the same data dependence relations between symbols lying in $\mathcal{N}_S(u)$ (Conditions (9), (12) of $S\, simil_u\, T$).
- Conditions (10) and (13) of $S\, simil_u\, T$ are a kind of counterpart for function symbols lying under if predicates to Condition (8) for symbols lying under while predicates.
- Condition (11) of $S\, simil_u\, T$ restricts the ordering of symbols in a linear schema.

1.2.3 Section 4

In this Section we prove that $S\, simil_u\, T$ implies $S \cong_u T$ for every $u \in \mathcal{V} \cup \{\omega\}$ and linear structured schemas S, T (Theorem 55).

1.2.4 Section 5

Here we define an Fpu-couple for a schema S, where $F \in \mathcal{F}^*$ and p is any predicate symbol in S. This is a pair of interpretations which define distinct final values for u

and differ only on one p-predicate term $p(\mathbf{t})$, where the vector term $\mathbf{t}$ is an F-term (Definition 28). We will also prove that the existence of Fpu-couples for a schema is preserved by u-equivalence.

1.2.5 Section 6

Here we prove that u-equivalent LFL schemas have the same set of u-needed symbols and the same data dependencies between these symbols (Theorem 75).

1.2.6 Section 7

We define the notion of a *passage* in a schema (Definition 78) and use this to prove an important theorem, Theorem 90.

1.2.7 Section 8

Here we show that u-equivalent LFL schemas S and T have essentially the same set of if predicates and while predicates (Theorem 96).

1.2.8 Section 9

In this section, we use the theory of pu-couples to show that u-equivalent LFL schemas satisfy similarities in syntactic structure (Theorems 105, 103 and Corollary 104).

1.2.9 Section 10

Here the concept of an *external* predicate term is defined and various results (in particular Lemma 117 and Corollary 116) are proved using this concept, which will be used in Section 11.

1.2.10 Section 11

Here the results in the preceding section are used to prove that if $S \cong_u T$ then S and T have essentially the same set of symbols lying in the body of each while predicate (Theorem 126). (As the function symbol h in the LFL schemas S and T in Figures 1 and 3 shows, no such simple statement holds for symbols lying under if predicates, although if $u = \omega$ then Theorem 118, which is also proved in Section 11, shows that S and T have the same set of while predicates lying under if predicates.)

1.2.11 Section 12

Here we show that if $S \cong_u T$ then the bodies of corresponding while predicates in S and T satisfy the same data dependencies (Theorem 127).

1.2.12 Section 13

Here we prove that u-equivalence implies Condition (11) of u-similarity, which restricts the order in which function symbols in S and T can occur. We also give an example to show that Condition (11) of u-similarity is not implied by the other Conditions of u-similarity.

1.2.13 Section 14

Here we prove that $\{u, \omega\}$-equivalence implies Conditions (12) and (13) of $\{u, \omega\}$-similarity (Theorem 147). Both conditions are concerned with the symbols in the body of a while predicate in S and T. We also give an example to show that Condition (13) of u-similarity is not implied by the other Conditions of u-similarity.

1.2.14 Section 15

Here we state formally the main theorem and discuss further possible directions.

2 Basic definitions and discussion of schemas

Definition 1 (symbol sets) Throughout this paper, $\mathcal{F}$, $\mathcal{P}$ and $\mathcal{V}$ denote fixed infinite sets of *function symbols*, of *predicate symbols* and of *variables* respectively. We assume a function

$$arity : \mathcal{F} \cup \mathcal{P} \to \mathbb{N}.$$

The arity of a symbol x is the number of arguments referenced by x. We assume that for each $n \in \mathbb{N}$ there are infinitely many elements of $\mathcal{F}$ and $\mathcal{P}$ of arity n, so we never run out of symbols of any required arity. Note that in the case when the arity of a function symbol g is zero, g may be thought of as a constant.

Definition 2 (Terms) The set $Term(\mathcal{F}, \mathcal{V})$ of *terms* is defined as follows:

- each variable is a term,
- if $f \in \mathcal{F}$ is of arity n and $t_1, \ldots, t_n$ are terms then $f(t_1, \ldots, t_n)$ is a term. If each term t_i is a variable, then $f(t_1, \ldots, t_n)$ is called a function expression.

We refer to a tuple $\mathbf{t} = (t_1, \ldots, t_n)$, where each t_i is a term, as a vector term. We call $p(\mathbf{t})$ a predicate term if $p \in \mathcal{P}$ and the number of components of the vector term $\mathbf{t}$ is $arity(p)$. If each component of $\mathbf{t}$ is a variable, then $p(\mathbf{t})$ is called a predicate expression.

2.1 Schema Definition

The first task is to define the class of schemas that are to be considered. In Section 3, Definition 4, we will give the formal definition of the class of structured schemas with which we will be working. In order to give a historical sketch of the study of schemas, we give here a more general definition. An (unstructured) *schema* is a finite sequence of *statements*, each beginning with a label, a positive integer, such that no two statements have the same label. Each statement has one of four forms; an assignment statement, $n :\ v := f(\mathbf{v})$; for a function expression $f(\mathbf{v})$, an if statement, $n :\ $ *if* $p(\mathbf{w})$ *then goto* L_1 *else goto* L_2;, where $p(\mathbf{w})$ is a predicate expression and L_1, L_2 are labels of statements, a *goto* statement, $n :\ $ *goto* L, or a *STOP* statement, $n :$ *STOP*. An example of an unstructured schema, taken from [5], is

$1 :\ w := f_1()$;
$2 :\ $ *goto* 5;
$3 :\ w := f_2(v, w)$;
$4 :\ v := f_3(v)$;
$5 :\ $ *if* $p(v)$ *then goto* 6 *else goto* 3;
$6 :\ $ *STOP*.

The symbols upon which schemas are built are given meaning by defining the notions of a state and of an interpretation. It will be assumed that 'values' are given in a single set D, which will be called the *domain*.

Definition 3 (states, (Herbrand) interpretations and the natural state e) Given a domain D, a *state* is either $\perp$ (denoting non-termination) or a function $\mathcal{V} \to D$. The set of all such states will be denoted by $\mathrm{State}(\mathcal{V}, D)$. An interpretation i defines, for each function symbol $f \in \mathcal{F}$ of arity n, a function $f^i : D^n \to D$, and for each predicate symbol $p \in \mathcal{P}$ of arity m, a function $p^i : D^m \to \{\mathsf{T}, \mathsf{F}\}$. The set of all interpretations with domain D will be denoted $\mathit{Int}(\mathcal{F}, \mathcal{P}, D)$.
When the domain used is $\mathit{Term}(\mathcal{F}, \mathcal{V})$, an interpretation i is said to be *Herbrand* if the functions $f^i : \mathit{Term}(\mathcal{F}, \mathcal{V}) \to \mathit{Term}(\mathcal{F}, \mathcal{V})$ for each $f \in \mathcal{F}$ are defined as

$$f^i(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$$

for all n-tuples of terms $(t_1, \ldots, t_n)$.
In the case when the domain is $\mathit{Term}(\mathcal{F}, \mathcal{V})$, the *natural state* $e : \mathcal{V} \to \mathit{Term}(\mathcal{F}, \mathcal{V})$ is defined by $e(v) = v$ for all $v \in \mathcal{V}$.

Note that an interpretation i being Herbrand places no restriction on the mappings $p^i : (\mathit{Term}(\mathcal{F}, \mathcal{V}))^m \to \{\mathsf{T}, \mathsf{F}\}$ defined by i for each $p \in \mathcal{P}$.

It is well known [1, Section 4-14] that Herbrand interpretations are the only ones that need to be considered when considering equivalence of schemas. This fact is stated more precisely in Theorem 21.

A program is obtained from a schema S and an interpretation i by replacing all symbols $f \in \mathcal{F}$ and $p \in \mathcal{P}$ in S by f^i and p^i; and given an initial state $d \in \mathrm{State}(\mathcal{V}, D)$, this program defines a final state

$$\mathcal{M}[\![S]\!]^i_d \in \mathrm{State}(\mathcal{V}, D)$$

in the obvious way. (If the program fails to terminate for an initial state d, or if $d = \perp$, then we define $\mathcal{M}[\![S]\!]^i_d = \perp$.)

For example if $D = \mathbb{N}$ and $f_1^i() = 1$, $f_2^i(m, n) = m \times n$, $f_3^i(n) = \max(n - 1, 0)$ and $p^i(0) = \mathsf{T}$ and $m \geq 1 \Rightarrow p^i(m) = \mathsf{F}$, then the schema in Subsection 2.1 defines the program

$1:\ w := 1;$
$2:\ \mathit{goto}\ 5;$
$3:\ w := v \times w;$
$4:\ v := \max(v - 1, 0);$

$5:$ *if* $v < 1$ *then goto* 6 *else goto* 3;
$6:$ *STOP*

computing the familiar factorial function; $\mathcal{M}[\![S]\!]^i_d(w) = (d(v))!$.

In Subsection 3.3 we give a formal definition of the semantics of a structured schema; for the class of unstructured schemas discussed in this Section, the sketch above seems sufficient.

2.3 Free and liberal schemas

The definitions of *free* and *liberal* schemas were invented in [3]. A schema can be represented by a finite directed graph (flow diagram) whose nodes are labeled by assignments, predicate statements, *START* or *STOP*. Let us call a path through a schema's flow diagram beginning at the *START* node *legal* if there is a Herbrand interpretation such that the program thus defined follows that path if each variable $v \in \mathcal{V}$ has the initial value v (that is, e is the initial state). A schema is *free* if all paths through its flow diagram beginning at the *START* node are legal. A schema is *liberal* if for any legal path through its flow diagram, any two assignment statements through which the path passes under a Herbrand interpretation from the initial state e, define different elements of the domain $Term(\mathcal{F}, \mathcal{V})$.

For example the schema

$1:\ v := f(u);$
$2:\ w := f(u);$

is not liberal if $v \neq u$, since for any interpretation, the variables v, w define the same element of the Herbrand domain ($Term(\mathcal{F}, \mathcal{V})$, that is) at the respective executions of the two statements. It is however liberal if $v = u$.

All predicate-free schemas (that is, schemas which have no predicate symbols) are free; the schema

$1:\ v := f(u);$
$2:$ *if* $p(v)$ *goto* 1 *else goto* 3;
$3:$ *STOP*

is not free if $v \neq u$, as the statement sequence 12123 cannot be executed by any interpretation with any initial state, since the variable v would define the same term on both occasions that statement 2 was executed.

As mentioned in the introduction, it was proved in [3] that it is decidable whether a schema is liberal, or liberal and free, but not whether a schema is free.

2.4 Positive results on the decidability of schema equivalence

Besides the result of [2] mentioned above, positive results on the decidability of equivalence of schemas include the following; in an early result in schema theory, Ianov [6] introduced a restrictive class of schemas, the Ianov schemas, for which equivalence is decidable. Ianov schemas are monadic (that is, they contain only a *single* variable) and all function symbols are unary; hence Ianov schemas are conservative.

Paterson [3] proved that equivalence is decidable for a class of schemas called *progressive schemas*, in which every assignment references the variable assigned by the previous assignment along every legal path.

Sabelfeld [7] proved that equivalence is decidable for another class of schemas called *through schemas*. A through schema satisfies two conditions: firstly, that on every path from an accessible predicate p to a predicate q which does not pass through another predicate, and every variable x referenced by p, there is a variable referenced by q which defines a term containing the term defined by x, and secondly, distinct variables referenced by a predicate define distinct terms under any Herbrand interpretation (Definition 3).

2.5 Variations on the Definition of Schemas and Interpretations

Other work on schemas considers the relative power of different schema languages [8–13]. In order to do this, the restriction that all functions in a schema are uninterpreted is relaxed, leading to the formulation of *semi interpreted schemas* in which some information about the nature of the functions in a schema is available.

Chandra and Manna [12] consider schemas *with equality*. A schema with equality is a schema in which there is a predicate symbol, which under any interpretation defines the equality test. The authors show that there are schemas with equality which are not equivalent to any schema without the equality symbol; thus adding the equality symbol gives a real increase in expressive power. They also show that equivalence of Ianov schemas with equality is undecidable.

Chandra [11] has also considered the consequence of specifying function symbols to be invertible (that is, to define invertible functions on the domain of computation) and specifying pairs of function symbols to commute. As with Ianov's paper, he only considers schemas with one variable, in which all function symbols are unary. He shows that the equivalence problem remains solvable if either commuting function symbols, or invertible function symbols are allowed, but not both.

Functional schemas have a similar relationship to functional programs as the program schemas considered here do to imperative programs. An example of a functional program is

$$F(v) = \mathit{if}\ (v = 0)\ \mathit{then}\ 1\ \mathit{else}\ vF(v-1);$$

an example of a functional schema defining this program would be

$$F(v) = \mathit{if}\ (p(v))\ \mathit{then}\ f()\ \mathit{else}\ h(v, F(g(v))).$$

Ashcroft, Manna and Pnueli [14] define a functional schema to be a finite set of expressions of this form, in which the uppercase function symbols (the function variables) appear on the left of exactly one such expression in the set. An interpretation for a functional schema interprets only the lowercase function symbols and the predicate symbols. Thus a functional schema and an interpretation define a function for each function variable.

It can be easily shown that any monadic program schema S with variable v can be effectively converted into a monadic functional schema, one of whose function variables defines the value $\mathcal{M}[\![S]\!]_d^i(v)$ for any state d and interpretation i. The converse statement is false however; Paterson [15] proved that there is no program schema having *any* number of variables 'equivalent' in this sense to the monadic functional schema

$$F(v) = \mathit{if}\ p(v)\ \mathit{then}\ v\ \mathit{else}\ f_1(F_0(F_0(f_2(v)))).$$

Thus functional schemas extend the expressive power of program schemas.

Two function schemas are said to be equivalent if for every interpretation they define the same function for each function variable. The authors consider a subclass of functional schemas which they call the free functional schemas (whose definition is similar to that of freeness for program schemas) and show that it is decidable whether two free monadic functional schemas (that is, functional schemas with only one variable) are equivalent. This result implies that equivalence of free monadic program schemas is decidable.

These alternative schema definitions use restrictions which are quite different to those used in this paper; in particular no other positive result for any class of linear schemas seems to have been proved, apart from that in [2].

2.6 Relevance of schema theory to program slicing

Our interest in the theory of program schemas is motivated in part by applications in program slicing. Slicing has many applications including program comprehension [16], software maintenance [17], [18], [19], [20], debugging [21], [22], [23], [24], testing [25],

[26], [27], re-engineering [28], [29], component reuse [30], [31], program integration [32], and software metrics [33], [34], [35]. There are several surveys of slicing techniques, applications and variations [36], [37], [38]. All applications of slicing rely on the fact that a slice is faithful to a projection of the original program's semantics, yet it is typically a smaller program.

The field of (static) program slicing is largely concerned with the design of algorithms which given a program and a variable v, eliminate as much code as possible from the program, such that the program (slice) consisting of the remaining code, when executed from the same initial state, will still give the same final value for v as the original program, and preserve termination. One algorithm is thus better than another if it constructs a smaller slice.

No algorithm exists that is guaranteed to delete all the code affecting the final value of the given variable. To see this, consider a program P, and define the program Q to be P followed by the assignment $w := 1;$. (We assume that the variable w does not occur in P.) This final assignment is 'needed' in Q by the variable w if and only if P terminates for at least one input state; hence the halting problem for programs, which is undecidable, can be obtained by reduction from the 'slicing problem' defined above.

Slicing algorithms do not normally take account of the meanings of the functions and predicates occurring in a program, nor do they 'know' when the same function or predicate occurs in more than one place in a program. In effect, therefore, they work with a linear schema defined by the program, and the semantic properties which slices of programs are required to preserve are defined in terms of schema semantics. This motivates the study of schemas, which represent large classes of programs.

Weiser [39] showed that given a program and a variable v, there was a particular set of functions and predicates (corresponding to our set $\mathcal{N}_S(v)$ for schemas in Definition 35) which may affect the final value of v; the symbols not lying in this set may simply be deleted without affecting the final value of v. In Theorem 42 we generalise this by considering ω-equivalence as a slicing criterion. Our main theorem shows that if S is LFL then none of the symbols in $\mathcal{N}_S(u)$ (for $u \in \mathcal{V} \cup \{\omega\}$) can be deleted from S without giving a u-inequivalent schema. This is however false for the class of schemas which are merely linear and free; a counterexample is given in Figure 6 in Section 5.

3 Further definitions and lemmas for structured schemas

This Section gives precise definitions of the syntax and semantics of structured schemas.

Definition 4 (structured schemas) We define the set $Sch(\mathcal{F}, \mathcal{P}, \mathcal{V})$ of all *structured schemas* recursively as follows. The empty schema $\Lambda \in Sch(\mathcal{F}, \mathcal{P}, \mathcal{V})$. An assignment $y := f(\mathbf{x})$; where $y \in \mathcal{V}$, and $f(\mathbf{x})$ is a function expression, lies in $Sch(\mathcal{F}, \mathcal{P}, \mathcal{V})$. From these all schemas in $Sch(\mathcal{F}, \mathcal{P}, \mathcal{V})$ may be 'built up' from the following constructs on schemas;

sequences; $S' = U_1 U_2 \ldots U_r \in Sch(\mathcal{F}, \mathcal{P}, \mathcal{V})$ provided that each schema

$$U_1, \ldots, U_r \in Sch(\mathcal{F}, \mathcal{P}, \mathcal{V}).$$

We define $S\Lambda = \Lambda S = S$ for all schemas S.

if **schemas;** $S'' = \textit{if}\ p(\mathbf{x})\ \textit{then}\ \{T_1\}\ \textit{else}\ \{T_2\}$ lies in $Sch(\mathcal{F}, \mathcal{P}, \mathcal{V})$ whenever $p(\mathbf{x})$ is a predicate expression and $T_1, T_2 \in Sch(\mathcal{F}, \mathcal{P}, \mathcal{V})$.

while **schemas;** $S''' = \textit{while}\ q(\mathbf{y})\ \textit{do}\ \{T\}$ lies in $Sch(\mathcal{F}, \mathcal{P}, \mathcal{V})$ whenever $q(\mathbf{y})$ is a predicate expression and T is a schema.

Finally, $|S|$ will denote the total number of function and predicate symbols in S, with n distinct occurrences of the same symbol counting n times.

Thus a schema is a word in a language over an infinite alphabet, for which Λ is the empty word. We normally omit the braces $\{$ and $\}$ if this causes no ambiguity. Also, we may write $\textit{if}\ p(\mathbf{x})\ \textit{then}\ \{T_1\}$ instead of
$\textit{if}\ p(\mathbf{x})\ \textit{then}\ \{T_1\}\ \textit{else}\ \{T_2\}$ if $T_2 = \Lambda$.

Observe that $f(\mathbf{x})$ and $p(\mathbf{x})$ in Definition 4 are always function and predicate *expressions*; that is, the components of the vector term $\mathbf{x}$ are variables.

For the remainder of this paper, the word 'schema' is intended to mean 'structured schema'.

In Definition 4 S'' will be referred to as an *if schema;* and S''' as a *while schema.* The predicate symbols p and q are called the *guards* of the schemas S'' and S''', respectively.

The sets of *if* and *while* predicate symbols occurring in a schema S are denoted by $\textit{ifPreds}(S)$ and $\textit{whilePreds}(S)$; their union is $\textit{Preds}(S)$. We define $\textit{Funcs}(S) \subseteq \mathcal{F}$ to be the set of function symbols in S and define $\textit{Symbols}(S) = \textit{Funcs}(S) \cup \textit{Preds}(S)$. A schema without predicates is called *predicate-free*; a schema without while predicates is called *while-free.*

Definition 5 (linear schemas) If no element of $\mathcal{F} \cup \mathcal{P}$ appears more than once in a schema S, then S is said to be *linear.* If a linear schema S contains an assignment

$y := f(\mathbf{x})$ then we define $assign_S(f) = y$, $\mathbf{refvec}_S(f) = \mathbf{x}$ and the set of components of $\mathbf{x}$ is $Refset_S(f) \subseteq \mathcal{V}$. If $p \in Preds(S)$ then $\mathbf{refvec}_S(p)$ and $Refset_S(p)$ are defined similarly.

3.1 Subschemas of Linear Schemas

The *subschemas* of a schema are defined as follows; the empty sequence Λ is a subschema of every schema; if $S \in Sch(\mathcal{F}, \mathcal{P}, \mathcal{V})$ is an assignment or Λ then the only subschemas of S are S itself and Λ; the subschemas of $U_1 \ldots U_r$ are those of each U_j for $1 \leq j \leq r$ and also the schemas $U_i U_{i+1} \ldots U_j$ for $i \leq j$; the subschemas of $S'' = \mathit{if}\, p(\mathbf{x})\, \mathit{then}\, \{T_1\}\, \mathit{else}\, \{T_2\}$ are S itself and those of T_1 and T_2; the subschemas of $S''' = \mathit{while}\, q(\mathbf{y})\, \mathit{do}\, \{T\}$ are S''' itself and those of T. The subschemas T_1 and T_2 of S'' are called the *true* and *false* parts of p (or of S''). In the *while* schema the subschema T is called the *body* of q (or of S''').

Definition 6 (subschemas $S(p)$, $part_S^X(p)$ and $body_S(p)$) Let S be a linear schema. If $p \in Preds(S)$ then we sometimes write $S(p)$ for the while or if subschema of S of which p is the guard.
Also, if $p \in ifPreds(S)$ and $X \in \{\mathsf{T}, \mathsf{F}\}$ then we may write $part_S^X(p)$ for the X-part of p in S.
If $p \in whilePreds(S)$ then $body_S(p)$ is the body of p in S.

Definition 7 (the $\searrow_S$ 'lying below' relation, 'immediately below') Let S be a linear schema. If $p \in Preds(S)$, we write $p \searrow_S x$ to mean $x \in Symbols(body_S(p))$ if $p \in whilePreds(S)$ and $x \in Symbols(part_S^{\mathsf{T}}(p)) \cup Symbols(part_S^{\mathsf{F}}(p))$ if $p \in ifPreds(S)$. We may strengthen this to $p \searrow_S x\,(X)$ to mean that either $x \in Symbols(part_S^X(p))$ (if $p \in ifPreds(S)$), or $x \in Symbols(body_S(p))$ (if $X = \mathsf{T}$ and $p \in whilePreds(S)$).
Also, if $A \subseteq Symbols(S)$, then we say that A lies *immediately* below S (or equivalently, S lies immediately above A) if $A \subseteq Symbols(S)$ and there is no $p \in whilePreds(S)$ such that $A \subseteq Symbols(body_S(p))$. In this case, if $S = body_T(q)$ for some linear schema T and $q \in whilePreds(T)$, we may also say that A lies immediately below q in T.

Definition 8 (main subschemas of a linear schema) Let S be a linear schema. The set of *main* subschemas of S contains S itself and the bodies of all while subschemas of S.

Observe that there is exactly one main subschema of a linear schema S lying immediately above a set $A \subseteq Symbols(S)$.

The execution of a program defines a (possibly infinite) sequence of assignments and predicates. Each such sequence will correspond to a *path* through the associated schema. The set $\Pi^\omega(S)$ of paths through S is now given.

Definition 9 (the set $alphabet(S)$ and the set $\Pi^\omega(S)$ of paths through S) If σ is a word, or a set of words over an alphabet, then $pre(\sigma)$ is the set of all prefixes of (elements of) σ. If L is any set, then we write L^* for the set of finite words over L and L^ω for the set containing both finite and infinite words over L, and we write Λ to refer to the empty word; recall that Λ is also a particular schema.
For each schema S the alphabet of S, written $alphabet(S)$ is defined by

$$alphabet(S) = A \cup B$$

where

$$A = \{y := f(\mathbf{x}) \mid \ y := f(\mathbf{x});\ \text{is an assignment in } S\},$$
$$B = \{\lessdot p(\mathbf{x}) = Z \gtrdot \mid \ p(\mathbf{x}) \text{ is a predicate expression in } S,\ Z \in \{\mathsf{T}, \mathsf{F}\}\}.$$

For any letter $l \in alphabet(S)$, we define $symbol(l) \in Symbols(S)$ to be f if l is an assignment with function symbol f, and p if l is $<p(\mathbf{x}) = Z>$ for $Z \in \{\mathsf{T}, \mathsf{F}\}$. The words in $\Pi(S) \subseteq (alphabet(S))^*$ are formed by concatenation from the words of subschemas as follows:

For Λ,

$$\Pi(\Lambda) = \{\Lambda\}.$$

For assignments,

$$\Pi(y := f(\mathbf{x});) = \{y := f(\mathbf{x})\}.$$

For sequences, $\Pi(S_1 S_2 \dots S_r) = \Pi(S_1) \dots \Pi(S_r)$.

For *if* schemas, $\Pi(\ \mathit{if}\ p(\mathbf{x})\ \mathit{then}\ \{T_1\}\ \mathit{else}\ \{T_2\})$ is the set of all concatenations of $\lessdot p(\mathbf{x}) = \mathsf{T} \gtrdot$ with a word in $\Pi(T_1)$ and all concatenations of $\lessdot p(\mathbf{x}) = \mathsf{F} \gtrdot$ with a word in $\Pi(T_2)$.

For *while* schemas, $\Pi(\ \mathit{while}\ q(\mathbf{y})\ \mathit{do}\ \{T\})$ is the set of all words of the form

$$[\lessdot q(\mathbf{y}) = \mathsf{T} \gtrdot\ \Pi(T)]^*\ \lessdot q(\mathbf{y}) = \mathsf{F} \gtrdot$$

where $[<q(\mathbf{y}) = \mathsf{T} > \Pi(T)]^*$ denotes a finite sequence of words which are the concatenation of $\lessdot q(\mathbf{y}) = \mathsf{T} \gtrdot$ with a word from $\Pi(T)$.

We define the set $\Pi^\omega(S)$ of *paths* through S as

$$\Pi^\omega(S) = \Pi(S) \cup \{\sigma \in (alphabet(S))^\omega - (alphabet(S))^* \mid pre(\sigma) - \{\sigma\} \subseteq pre(\Pi(S))\}.$$

When refering to a linear schema S, we will sometimes omit the reference to $\mathbf{refvec}_S(p)$ for $p \in Preds(S)$ when denoting elements of $alphabet(S)$; that is, we will write $\lessdot p = Z \gtrdot$ to refer to $\lessdot p(\mathbf{x}) = Z \gtrdot$. Since the schema S is linear, this is unambiguous.

Definition 10 (Paths *passing through* a Symbol) We say that a path $\sigma \in \Pi^\omega(S)$ *passes* through a function symbol f (or a predicate p) if it contains an assignment with function symbol f (or $\lessdot p(\mathbf{x}) = Z \gtrdot$ for $Z \in \{\mathsf{T}, \mathsf{F}\}$). We may strengthen this by saying that σ passes through an element $l \in alphabet(S)$ if l occurs in σ.

3.3 Semantics of Structured Schemas

Given a schema $S \in Sch(\mathcal{F}, \mathcal{P}, \mathcal{V})$ and a domain D, an initial state $d \in \mathrm{State}(\mathcal{V}, D)$ with $d \neq \bot$ and an interpretation $i \in Int(\mathcal{F}, \mathcal{P}, D)$ we now define the final state $\mathcal{M}[\![S]\!]^i_d \in \mathrm{State}(\mathcal{V}, D)$ and the associated path $\pi_S(i, d) \in \Pi^\omega(S)$.

Definition 11 (The Schema $schema(\sigma)$) Given a word $\sigma \in (alphabet(S))^*$, the predicate-free schema $schema(\sigma)$ consists of all the assignments along σ in the same order as in σ; and $schema(\sigma) = \Lambda$ if σ has no assignments.

Lemma 12 *Let S be a schema.*

(1) *If $\sigma \in pre(\Pi(S))$, the set $\{l \in alphabet(S) \mid \sigma l \in pre(\Pi(S))\}$ is one of the following; the empty set, a singleton containing an assignment, or a pair $\{\lessdot p(\mathbf{x}) = \mathsf{T} \gtrdot, \lessdot p(\mathbf{x}) = \mathsf{F} \gtrdot\}$ where $p \in Preds(S)$.*
(2) *An element of $\Pi(S)$ cannot be a strict prefix of another.*

Proof. Both assertions follow by induction on $|S|$. □

Lemma 12 reflects the fact that at any point in the execution of a program, there is never more than one 'next step' which may be taken.

Definition 13 (The Semantics of Predicate-free Schemas) Given a state $d \neq \bot$, the final state $\mathcal{M}[\![S]\!]^i_d$ and associated path $\pi_S(i, d) \in \Pi^\omega(S)$ of a schema S are defined as follows:

For Λ,

$$\mathcal{M}[\![\Lambda]\!]^i_d = d$$

and

$$\pi_\Lambda(i,d) = \Lambda.$$

For assignments,

$$\mathcal{M}[\![y := f(\mathbf{x});]\!]^i_d(v) \quad = \quad \begin{cases} d(v) & \text{if } v \neq y, \\ f^i(d(\mathbf{x})) & \text{if } v = y \end{cases}$$

(where $d(x_1,\ldots,x_r)$ is defined to be the tuple $(d(x_1),\ldots,d(x_r))$)

and

$$\pi_{y := f(\mathbf{x});}(i,d) \quad = \quad y := f(\mathbf{x}),$$

and for sequences S_1S_2 of predicate-free schemas,

$$\mathcal{M}[\![S_1S_2]\!]^i_d \quad = \quad \mathcal{M}[\![S_2]\!]^i_{\mathcal{M}[\![S_1]\!]^i_d}$$

and

$$\pi_{S_1S_2}(i,d) \quad = \quad \pi_{S_1}(i,d)\pi_{S_2}(i,\mathcal{M}[\![S_1]\!]^i_d).$$

This uniquely defines $\mathcal{M}[\![S]\!]^i_d$ and $\pi_S(i,d)$ if S is predicate-free.

In order to give the semantics of a general schema S, first the path, $\pi_S(i,d)$, of S with respect to interpretation, i, and initial state d is defined.

Definition 14 (The path $\pi_S(i,d)$) Given a schema S, an interpretation i, and a state, $d \neq \bot$, the path $\pi_S(i,d) \in \Pi^\omega(S)$ is defined by the following condition; for all $\sigma \lessdot p(\mathbf{x}) = X \gtrdot \in pre(\pi_S(i,d))$, the equality $p^i(\mathcal{M}[\![schema(\sigma)]\!]^i_d(\mathbf{x})) = X$ holds.

In other words, the path $\pi_S(i,d)$ has the following property; if a predicate expression $p(\mathbf{x})$ along $\pi_S(i,d)$ is evaluated with respect to the predicate-free schema consisting of the sequence of assignments preceding that predicate in $\pi_S(i,d)$, then the value of the resulting predicate term given by i 'agrees' with the value given in $\pi_S(i,d)$.

By Lemma 12, this defines the path $\pi_S(i,d) \in \Pi^\omega(S)$ uniquely.

Definition 15 (the semantics of arbitrary schemas) If $\pi_S(i,d)$ is finite, we define

$$\mathcal{M}[\![S]\!]^i_d = \mathcal{M}[\![schema(\pi_S(i,d))]\!]^i_d$$

(which is already defined, since $schema(\pi_S(i,d))$ is predicate-free) otherwise $\pi_S(i,d)$ is infinite and we define $\mathcal{M}[\![S]\!]^i_d = \bot$. In this last case we may say that $\mathcal{M}[\![S]\!]^i_d$ is not terminating. For convenience, if S is predicate-free and $d : \mathcal{V} \to Term(\mathcal{F},\mathcal{V})$ is a state then we define unambiguously $\mathcal{M}[\![S]\!]_d = \mathcal{M}[\![S]\!]^i_d$; that is, we assume that the interpretation i is Herbrand if $Term(\mathcal{F},\mathcal{V})$ is the domain for which d is defined. For schemas S,T and interpretations i and j we write $\mathcal{M}[\![S]\!]^i_d(\omega) = \mathcal{M}[\![T]\!]^j_d(\omega)$ to mean $\mathcal{M}[\![S]\!]^i_d = \bot \iff \mathcal{M}[\![T]\!]^j_d = \bot$.

Observe that $\mathcal{M}[\![S_1 S_2]\!]^i_d = \mathcal{M}[\![S_2]\!]^i_{\mathcal{M}[\![S_1]\!]^i_d}$ and

$$\pi_{S_1 S_2}(i,d) = \pi_{S_1}(i,d)\pi_{S_2}(i,\mathcal{M}[\![S_1]\!]^i_d)$$

hold for all schemas (not just predicate-free ones).

Definition 16 (Termination from the natural state e) If $\mathcal{M}[\![S]\!]^i_e \neq \bot$, then we say that i is a *terminating* interpretation for S.

Definition 17 (Changing an interpretation) Given an interpretation i and $X \in \{\mathsf{T}, \mathsf{F}\}$ and $p \in \mathcal{P}$, the Herbrand interpretation $i(p = X)$ is given by

$$q^{i(p=X)}(\mathbf{t}) = \begin{cases} q^i(\mathbf{t}) & q \neq p \\ X & q = p \end{cases}$$

for every vector of terms $\mathbf{t}$ of the appropriate length. We generalise this by defining the interpretation $i(p(\mathbf{t}) = X)$ to mean the interpretation j satisfying $p^j(\mathbf{t}) = X$ and $q^j(\mathbf{t}') = q^i(\mathbf{t}')$ for all predicate terms $q(\mathbf{t}') \neq p(\mathbf{t})$.

Definition 18 (prefixes of paths passing through predicate terms) Let S be a linear schema, let d be a state and let $\mu \in \mathit{pre}(\Pi(S))$. We say that μ passes through a predicate term $p(\mathbf{t})$ *starting at d* if μ has a prefix μ' ending in $<p = Y>$ for $y \in \{\mathsf{T}, \mathsf{F}\}$ such that $\mathcal{M}[\![\mathit{schema}(\mu')]\!]_d(\mathbf{refvec}_S(p)) = \mathbf{t}$ holds. We say that $p(\mathbf{t}) = Y$ is a *consequence* of μ starting at d. We may omit the phrase 'starting at d' if $d = e$, or if the intended state seems obvious.

Definition 19 (u-equivalence of Schemas) Given any $u \in \mathcal{V} \cup \{\omega\}$, we say that schemas $S, T \in \mathit{Sch}(\mathcal{F}, \mathcal{P}, \mathcal{V})$ are *u-equivalent,* written $S \cong_u T$, if for every domain D and state $d : \mathcal{V} \to D$ and every $i \in \mathit{Int}(\mathcal{F}, \mathcal{P}, D)$, the following holds; either $u \in \mathcal{V} \wedge \bot \in \{\mathcal{M}[\![S]\!]^i_d, \mathcal{M}[\![T]\!]^i_d\}$, or

$$\mathcal{M}[\![S]\!]^i_d(u) = \mathcal{M}[\![T]\!]^i_d(u).$$

If $V \subseteq \mathcal{V} \cup \{\omega\}$, we write $S \cong_V T$ to mean $S \cong_u T \;\; \forall u \in V$ and we write $S \cong T$ to mean $S \cong_{\mathcal{V} \cup \{\omega\}} T$.

In Subsection 2.3 the definitions of free and liberal schemas were sketched for unstructured schemas. Here we give a formal definition of these terms for the class of structured schemas.

Definition 20 (free and liberal schemas) Let S be a schema.

- If for every $\sigma \in \mathit{pre}(\Pi(S))$ there is a Herbrand interpretation i such that $\sigma \in \mathit{pre}(\pi_S(i,e))$, then S is said to be *free.*
- If for every Herbrand interpretation i and any prefix $\mu \;\; v := f(\mathbf{a}) \;\; \nu \;\; w := g(\mathbf{b}) \in$

$pre(\pi_S(i,e))$, we have

$$\mathcal{M}[\![schema(\mu\ v := f(\mathbf{a}))]\!]_e(v) \neq \mathcal{M}[\![schema(\mu\ v := f(\mathbf{a})\ \nu w := g(\mathbf{b}))]\!]_e(w),$$

then S is said to be *liberal.* (If $f \neq g$ then of course this condition is trivially satisfied.)

Thus a schema S is said to be free if for every path through S, there is a Herbrand interpretation which follows it starting at the natural state e, and a schema S is said to be liberal if given any path through S passing through two assignments and a Herbrand interpretation which follows it starting at the natural state e, the assignments give distinct values to the variables to which they assign.

Observe that if a schema S is free, and

$$\mu \lessdot p(\mathbf{x}) = X \gtrdot \mu' \lessdot p(\mathbf{y}) = Y \gtrdot \in pre(\pi_S(i,e))$$

for some Herbrand interpretation i, then

$$\mathcal{M}[\![schema(\mu)]\!]_e(\mathbf{x}) \neq \mathcal{M}[\![schema(\mu\mu')]\!]_e(\mathbf{y})$$

holds, since otherwise there would be no Herbrand interpretation whose path (for e) has the prefix $\mu \lessdot p(\mathbf{x}) = X \gtrdot \mu' \lessdot p(\mathbf{y}) = \neg X \gtrdot$. Thus a path through a free schema cannot pass twice (starting at e) through the same predicate term.

We will later give a syntactic condition which characterises being liberal and free for linear schemas in the next subsection (Theorem 26).

We will refer to a schema which is liberal, free and linear as an LFL schema.

The following theorem, which is a restatement of [1, Theorem 4-1], ensures that we only need to consider Herbrand interpretations.

Theorem 21 *Let S be a schema and let D be a domain.*

- *For any state $d : \mathcal{V} \to D$ and any interpretation $i \in Int(\mathcal{F},\mathcal{P},D)$, there exists a Herbrand interpretation j such that $\pi_S(j,e) = \pi_S(i,d)$.*
- *If T is a schema, $u \in \mathcal{V} \cup \{\omega\}$ and for all Herbrand interpretations j we have $\mathcal{M}[\![S]\!]^j_e(u) = \mathcal{M}[\![T]\!]^j_e(u)$ (provided both sides terminate if $u \in \mathcal{V}$), then $S \cong_u T$.*

Throughout the remainder of the paper, all interpretations will be assumed to be Herbrand.

Clearly the relation $\cong_\omega$ is an equivalence relation on the set of schemas. Proposition 22 shows that the relation $\cong_v$ for variable v also is on the class of free schemas.

Proposition 22 (transitivity of $\cong_v$ for free schemas) *Let $v \in \mathcal{V}$; then the relation $\cong_v$ is an equivalence relation when restricted to the class of free schemas.*

Proof. Only transitivity is at issue. Suppose $S' \cong_v S''$ and $S'' \cong_v S'''$ hold for free schemas S', S'', S'''. Let i be an interpretation and assume that

$$\perp \notin \{\mathcal{M}[\![S']\!]^i_e, \mathcal{M}[\![S''']\!]^i_e\}$$

holds. Let the interpretation j map every predicate term $p(\mathbf{t})$ to F unless $\pi_{S'}(i,e)$ or $\pi_{S'''}(i,e)$ passes through $p(\mathbf{t})$, in which case let $p^j(\mathbf{t}) = p^i(\mathbf{t})$. Thus $\mathcal{M}[\![S']\!]^i_e = \mathcal{M}[\![S']\!]^j_e$ and $\mathcal{M}[\![S''']\!]^i_e = \mathcal{M}[\![S''']\!]^j_e$ hold and j maps finitely many predicate terms to T, hence $\mathcal{M}[\![S'']\!]^j_e \neq \perp$ holds. Thus

$$\mathcal{M}[\![S']\!]^j_e(v) = \mathcal{M}[\![S'']\!]^j_e(v) = \mathcal{M}[\![S''']\!]^j_e(v)$$

holds, giving the result. $\square$

Proposition 22 is false for the set of all schemas. To see this, consider the three schemas

$$\begin{aligned} S' = \ & \mathit{if}\ p(u)\ \mathit{then}\ v := f_1(); \\ & \qquad\qquad \mathit{else}\ v := g(); \end{aligned}$$

$$\begin{aligned} S'' = \ & \mathit{while}\ p(u)\ \mathit{do}\ \Lambda; \\ & v := g(); \end{aligned}$$

$$\begin{aligned} S''' = \ & \mathit{if}\ p(u)\ \mathit{then}\ v := f_2(); \\ & \qquad\qquad \mathit{else}\ v := g(); \end{aligned}$$

of which S'' is not free. Clearly $S' \cong_v S''$ and $S'' \cong_v S'''$ hold, but not $S' \cong_v S'''$.

3.4 Data dependence Relations

Definition 23 (segments of a schema and of segments) Let S be a schema and let $\mu \in \mathit{alphabet}(S)^*$. We say that μ is a
segment (in S) if there are words μ_1, μ_2 such that $\mu_1\mu\mu_2 \in \Pi(S)$. If μ, σ are segments in S, then we say that μ is a segment of σ in S if we can write $\sigma = \mu_1\mu\mu_2$.
We say that a segment μ starts (ends) at $x \in \mathit{Symbols}(S)$ if $\tilde{x} \in \mathit{alphabet}(S)$ is the first (last) letter of μ, with $x = \mathit{symbol}(\tilde{x})$.

Definition 24 (the $\rightsquigarrow_S$ 'data dependence' relation) Let S be a linear schema. We write $f \rightsquigarrow_S x$ for $f \in \mathit{Funcs}(S)$, $x \in \mathit{Symbols}(S)$ if there is a segment $\tilde{f}\sigma\tilde{x}$ in S such that $\tilde{f}$ is an assignment to f and $\tilde{x} \in \mathit{alphabet}(S)$ satisfies $\mathit{symbol}(\tilde{x}) = x$, and there is no assignment to the variable $\mathit{assign}_S(f)$ along σ. We call $\tilde{f}\sigma\tilde{x}$ an fx-segment in this case. We generalise this by defining $f \rightsquigarrow_S v$ for $f \in \mathit{Funcs}(S)$, $v \in \mathcal{V}$ if $f \rightsquigarrow_{S\,w := g(v);}\ g$ holds for any linear schema $S\,w := g(v);$, in which case we define an fv-segment in S to be any segment σ of S such that $\sigma\,w := g(v)$ is a fg-segment in the schema $S\,w := g(v);$. Lastly, we write $v \rightsquigarrow_S x$ for $v \in \mathcal{V}$, $x \in \mathit{Symbols}(S)$ if

$h \rightsquigarrow_{v:=h();\,S} x$ holds for any linear schema $v := h();\ S$, in which case we define a vx-segment in S to be any $\sigma \in \mathit{pre}(\Pi(S))$ such that $v := h()\,\sigma$ is an hx-segment in the schema $v := h();\ S$.
In all cases, we may strengthen the relation $x \rightsquigarrow_S y$ by writing $x \rightsquigarrow_S y\,(n)$ for $n \in \mathbb{N}$ if either $y \in \mathcal{V}$ or the nth component of $\mathbf{refvec}_S(y)$ is x or $\mathit{assign}_S(x)$.

Thus $f \rightsquigarrow_S x$ holds for $f \in \mathit{Funcs}(S)$, $x \in \mathit{Symbols}(S)$ if and only if there exists a path in S along which a (predicate) term $x(\mathbf{t})$ such that $\mathbf{t}$ has a component $f(\mathbf{t}')$ is created; and we may define an fx-segment to be any segment in S which 'witnesses' such a creation. Similar characterisations can be given for the statements $f \rightsquigarrow_S v$ and $v \rightsquigarrow_S x$ for $v \in \mathcal{V}$.
As an example, if T is the linear schema of Figure 3, the relations $v \rightsquigarrow_T q$, $v \rightsquigarrow_T k$, $k \rightsquigarrow_T k$ (but not $k \rightsquigarrow_T v$), $w \rightsquigarrow_T p$, $h \rightsquigarrow_T f$, $h \rightsquigarrow_T u$, $f \rightsquigarrow_T v$, and $g \rightsquigarrow_T v$ hold.

Definition 25 extends the notion of a μ-segment to cases in which μ has any length greater than one.

Definition 25 (μ-segments and μ-prefixes) Let S be a linear schema and let $x_1, \ldots, x_m \in \mathcal{F} \cup \mathcal{P} \cup \mathcal{V}$, for $m \geq 2$. Assume that $x_i \in \mathcal{P} \Rightarrow i = m$ and $x_{i+1}, \ldots, x_{m-1}$ are not variables, and at least one x_i lies in $\mathcal{F} \cup \mathcal{P}$. Then an $x_1 \ldots x_m$-segment of S is a segment $\rho = \tilde{x}_1 \rho_1 \tilde{x}_2 \rho_2 \ldots \rho_{m-1} \tilde{x}_m$ in S such that each $\tilde{x}_i \rho_i \tilde{x}_{i+1}$ is an $x_i x_{i+1}$-segment. Also, we call any $\rho \in \mathit{pre}(\Pi(S))$ a μ-prefix in S if either $\rho = \rho' \rho''$ for some μ-segment ρ'', or $\mu \in \mathcal{F} \cup \mathcal{P}$ and the last letter of ρ has symbol μ.

Observe that if S is a linear schema, $g \in \mathcal{F}$ and $G \in \mathcal{F}^*$, and g lies immediately below S, then there can be no gGg-segment in S.

As mentioned in the introduction, it was proved in [3] that it is decidable whether a schema is liberal, or liberal and free. Theorem 26 gives the essential result for linear schemas.

Theorem 26 (syntactic condition for being liberal and free)
Let S be a linear schema. Then S is both liberal and free if and only if for every segment $\tilde{x}\mu\tilde{y}$ in S with $\tilde{x}, \tilde{y} \in \mathit{alphabet}(S)$, $\mathit{symbol}(\tilde{x}) = \mathit{symbol}(\tilde{y})$ and such that the same symbol does not occur more than once in $\tilde{x}\mu$ or $\mu\tilde{y}$, then the segment $\tilde{x}\mu$ contains an assignment to a variable referenced by $\tilde{y}$.
In particular, it is decidable whether a linear schema is both liberal and free.

Proof [3]. Assume that S is both liberal and free. Then for any segment $\tilde{x}\mu\tilde{y}$ satisfying the conditions given, there is a prefix Θ and an interpretation i such that $\Theta\tilde{x}\mu\tilde{y} \in \mathit{pre}(\pi_S(i, e))$, and distinct (predicate) terms are defined when $\tilde{x}$ and $\tilde{y}$ are reached, thus proving the condition.
To prove sufficiency, first observe that the 'non-repeating' condition on the letters of the segments $\tilde{x}\mu$ and $\mu\tilde{y}$ may be ignored, since segments that begin and end with letters having the same symbol can be removed from within $\tilde{x}\mu$ or $\mu\tilde{y}$ until it is satisfied. Consider the set of prefixes of $\Pi(S)$ of the form $\Theta\tilde{x}\mu\tilde{y}$ with $\mathit{symbol}(\tilde{x}) = \mathit{symbol}(\tilde{y})$ such

that $\tilde{x}\mu\tilde{y}$ satisfies the condition given. By induction on the length of such prefixes, it can be shown that every assignment encountered along such a prefix defines a different term (for initial state e), and the result follows immediately from this.
Since there are finitely many segments in S which contain no repeated symbols except at the endpoints, and these can be enumerated, the decidability of liberality and freeness for the set of linear schemas follows easily. $\square$

Theorem 26 can easily be generalised to apply to arbitrary unstructured schemas; we state it in restricted form in order to simplify the notation used.

Lemma 27 (converting $f \rightsquigarrow_S v$ to $f \rightsquigarrow_S p$) *Let $v \in \mathcal{V}$ and let S be a linear schema. Let H be the linear schema* $\textit{if } p(v) \textit{ then } v := g(); \textit{ else } v := h();$. *Suppose the schema SH is linear. If $f \rightsquigarrow_S v$ holds then $f \rightsquigarrow_{SH} p$ holds. Also, if T is a linear schema such that TH is linear then $S \cong_u T \Rightarrow SH \cong_u TH$ holds for each $u \in \{v, \omega\}$.* $\square$

Definition 28 (F-terms and data dependence in terms) Let $F \in \mathcal{F}^* \cup \mathcal{V}\mathcal{F}^*$. We define F-terms inductively. The variable v is a v-term. If at least one of the terms $t_1, \ldots, t_n$ is an F-term, then the vector term $\mathbf{t} = (t_1, \ldots, t_n)$ is an F-term and the term $g(\mathbf{t})$ is an Fg-term. Any FF'-term is also an F'-term.
If a term t contains a subterm $g(t_1, \ldots, t_n)$ and some t_i is an x-term for $x \in \mathcal{V} \cup \mathcal{F}$, then we may write $x \rightsquigarrow_t g$ or $x \rightsquigarrow_t g\,(i)$. We also write $x \rightsquigarrow_{p(\mathbf{t}')} g$ or $x \rightsquigarrow_{\mathbf{t}'} g$ (for any $p \in \mathcal{P}$) if t is one of the components of $\mathbf{t}'$.

3.5 Other Relations between Schema Symbols

Definition 29 gives four relations which strengthen the data dependence relation.

Definition 29 (the *outif, outwhile, thru* and *back* relations) Let S be a linear schema and let $x \in \mathcal{F} \cup \mathcal{V}$ and $y \in \mathcal{F} \cup \mathcal{P} \cup \mathcal{V}$. Let $p \in \mathcal{P}$. Assume that $x \rightsquigarrow_S y$ holds. Then we make the following definitions.

- If $p \in \textit{whilePreds}(S)$ and $x \in \textit{Funcs}(\textit{body}_S(p))$ and $y \notin \textit{Symbols}(\textit{body}_S(p))$, then we write $\textit{outwhile}_S(p, x, y)$. (Note that $\textit{outwhile}_S(p, x, p)$ may hold). If $p \in \textit{whilePreds}(S)$ and both x and y are symbols in $\textit{body}_S(p)$ but $\neg(x \rightsquigarrow_{\textit{body}_S(p)} y)$ holds (a backward data dependence) then we write $\textit{back}_S(p, x, y)$.
- If $Y \in \{\mathsf{T}, \mathsf{F}\}$ and $p \in \textit{ifPreds}(S)$ and $x \in \textit{Funcs}(\textit{part}_S^Y(p))$ and $\neg(x \rightsquigarrow_{\textit{part}_S^Y(p)} y) \vee (y \in \mathcal{V})$ holds, then we write $\textit{outif}_S(p, Y, x, y)$. If $Y \in \{\mathsf{T}, \mathsf{F}\}$ and $p \in \textit{ifPreds}(S)$ and neither x nor y is a symbol in either of the schemas $\textit{part}_S^{\mathsf{T}}(p)$ or $\textit{part}_S^{\mathsf{F}}(p)$ and every xy-segment contains the letter $<p = Y>$, then we write $\textit{thru}_S(p, Y, x, y)$. (Note that $\textit{thru}_S(p, Y, x, p)$ is always false.)

$$\begin{aligned}&\textit{while}\ p(v)\ \textit{do}\\&\qquad\{\\&\qquad u := g(v);\\&\qquad v := f();\\&\qquad\}\end{aligned}$$

Fig. 4. $back_S(p, f, g)$ holds here

As an example, $back_S(p, f, g)$ holds if S is the linear schema in Figure 4.

Definition 30 (q-competing function symbols and variables) Let S be a linear schema and assume that $f \rightsquigarrow_S x\,(n)$ and $g \rightsquigarrow_S x\,(n)$ for $f, g, x \in Symbols(S) \cup \mathcal{V}$ and $n \in \mathbb{N}$. Let $q \in ifPreds(S)$. We say that f and g are q-competing for x in S if for $\{X, Y\} = \{\mathsf{T}, \mathsf{F}\}$, we have both $outif_S(q, X, f, x) \vee thru_S(q, X, f, x)$ and $outif_S(q, Y, g, x) \vee thru_S(q, Y, g, x)$.

Thus f and g are p-competing for v in the schemas of Figures 1 and 5. Proposition 31 will be used in later sections.

Proposition 31 (Connection between $outif_S$ and $thru_S$) *Let S be a linear schema and assume that $thru_S(p, Y, x, y)$ holds for some $p \in ifPreds(S)$, $Y \in \{\mathsf{T}, \mathsf{F}\}$ and $x, y \in Symbols(S) \cup \mathcal{V}$. Assume that $x \rightsquigarrow_S y\,(n)$ holds for $n \in \mathbb{N}$. Then every path in $\Pi(part_S^{\neg Y}(p))$ passes through some $f' \in \mathcal{F}$ satisfying $outif_S(p, \neg Y, f', y)$ and $f' \rightsquigarrow_S y\,(n)$.*

Proof. We may assume that $x \in \mathcal{F}$ holds, otherwise we may replace S with a linear schema $x := f();\ S$. Since $thru_S(p, Y, x, y)$ holds, there is an xy-segment $\gamma = \mu <\!\!p = Y\!\!> \mu'\mu''$ in S with $\mu' \in \Pi(part_S^Y(p)))$ and $\mu'' \neq \Lambda$. We may assume that p occurs only once in the segment γ; otherwise we could delete a segment from within γ. Let $\sigma \in \Pi(part_S^{\neg Y}(p))$. The segment $\mu <\!\!p = \neg Y\!\!> \sigma\mu''$ does not enter the Y-part of p and so is not an xy-segment, by the definition of $thru_S(p, Y, x, y)$. Thus the variable assigned by x is 'killed' along σ, giving the result. $\square$

Definition 32 (The $above_S$ function) Let S be a linear schema and let x be a

$$\begin{aligned}&u := h();\\&v := f(u);\\&\textit{if}\ p(w)\quad \textit{then}\ \ \Lambda\\&\qquad\qquad\ \ \textit{else}\ \ v := g();\end{aligned}$$

Fig. 5. $thru_S(p, \mathsf{T}, f, v) \wedge outif_S(p, \mathsf{F}, g, v)$ holds here

symbol in S. If x lies immediately below S, then we define $\mathit{above}_S(x) = x$; otherwise we define $\mathit{above}_S(x)$ to be the while predicate lying immediately below S and containing x in its body.

Proposition 33 (Connection between above_S and back_S relations) *Let S be a linear schema and assume that $\mathit{back}_S(p, f, x)$ holds for some $f \in \mathcal{F}$, $x \in \mathit{Symbols}(S)$ and $p \in \mathit{whilePreds}(S)$. If $f \neq x$ then $\mathit{above}_{\mathit{body}_S(p)}(f) \neq \mathit{above}_{\mathit{body}_S(p)}(x)$ holds.*

Proof. Since $f \neq x$ holds, $\mathit{above}_{\mathit{body}_S(p)}(f) = \mathit{above}_{\mathit{body}_S(p)}(x)$ implies

$$\mathit{above}_{\mathit{body}_S(p)}(f) = q \in \mathit{whilePreds}(\mathit{body}_S(p))$$

holds. Thus $f \rightsquigarrow_{S(q)} x$ and hence $f \rightsquigarrow_{\mathit{body}_S(p)} x$ holds, contradicting $\mathit{back}_S(p, f, x)$. $\square$

Definition 34 (The $\ll_S$ relation) Let S be a linear schema and let

$$\{x, y\} \subseteq \mathit{Symbols}(S).$$

Assume that S lies immediately above the set $\{x, y\}$. We define $x \ll_S y$ if $\mathit{above}_S(x) \neq \mathit{above}_S(y)$ and there is a segment in S which begins at $\mathit{above}_S(x)$ and ends at $\mathit{above}_S(y)$.

Observe the following; if $x \ll_S y$ then every segment in S which begins at x and ends at y passes through every occurrence of x before any occurrence of y, and x and y do not lie in opposite parts of any if predicate. Also, $\ll_S$ is transitive; and $x \ll_S y \wedge y \ll_S x$ never holds, since otherwise S would contain a while predicate containing both $\mathit{above}_S(x)$ and $\mathit{above}_S(y)$ in its body.

3.6 The $\mathcal{N}_S$ and Inv_S sets

The symbol and variable sets of Definition 35 are purely syntactically defined, and contain all the symbols and (initial) variables which can influence the final value of a variable. This is stated precisely in Theorem 42.

Definition 35 (symbols *needed* by variables) Let S be a linear schema and let $x \in \{\omega\} \cup \mathcal{V} \cup \mathit{Symbols}(S)$. Then we define the set $\mathcal{N}_S(x)$ to be the minimal subset of $\mathit{Symbols}(S)$ satisfying the following closure conditions; if $f \in \mathcal{F}$, $y \in \{x\} \cup \mathcal{N}_S(x)$ and $f \rightsquigarrow_S y$ then $f \in \mathcal{N}_S(x)$; and if $x = \omega$ then $\mathit{whilePreds}(S) \subseteq \mathcal{N}_S(x)$; and if $p \searrow_S y$ for $y \in \mathcal{N}_S(x)$ then $p \in \mathcal{N}_S(x)$.
We also define $\mathit{Inv}_S(x) \subseteq \mathcal{V}$ to contain all variables v satisfying $v \rightsquigarrow_S v$ if $v = x \in \mathcal{V}$ or $v \rightsquigarrow_S y$ for some $y \in \mathcal{N}_S(x)$.
We generalise this by defining $\mathcal{N}_S(V) = \cup_{u \in V} \mathcal{N}_S(u)$ for a set V, and similarly with Inv_S.

Note that $\mathcal{N}_S(y)$ is a set of symbols of S, whereas $Inv_S(y)$ is a subset of $\mathcal{V}$.

It can easily be proved that if $v \in \mathcal{V}$ and a linear schema $S = AB$, then $Inv_S(v) = Inv_A(Inv_B(v))$.

Observe that if any of the relations given in Definition 29 hold, and $y \in \mathcal{N}_S(u)$ for some $u \in \mathcal{V} \cup \{\omega\}$, then $x \in \mathcal{N}_S(u)$ holds; in the case that $thru_S(p, Y, x, y)$ holds, this follows from Proposition 31.

Definition 36 (dependence sequences, $depnum_S(y, x)$) Let S be a linear schema and assume that $y \in \mathcal{N}_S(x) \cup Inv_S(x)$. Then an x-dependence sequence for y in S is a word $w \in (\mathcal{P} \cup \mathcal{F} \cup \mathcal{V})^* - \mathcal{V}$ beginning in y and ending in x (or in a element of $whilePreds(S)$ if $x = \omega$) which 'witnesses' this fact; that is, only the first and last letters of w may be variables; also, if $f \in \mathcal{F}$ is in w, and z is the next letter in w after f, then $f \rightsquigarrow_S z$; if $p \in \mathcal{P}$ is in w, and z is the next letter in w after p, then $p \searrow_S z$. We define $depnum_S(y, x) \in \mathbb{N}$ to be the minimal length of any x-dependence sequence for y in S.

Observe that if $v \in \mathcal{V}$ and $x \in \mathcal{N}_S(v) \cup Inv_S(v)$ then $depnum_S(x, v) \geq 2$ holds, since v is not a dependence sequence.
Clearly if w is an x-dependence sequence for y of minimal length, then w contains no repeated letters in $\mathcal{P} \cup \mathcal{F}$. Observe also that if w is minimal for S then $\neg(p \searrow_S z)$ holds for any p in w and any letter z occurring after p in w, unless z occurs immediately after p.

Proposition 37 (form of dependence sequences) *Let S be a linear schema and let $u \in \mathcal{V} \cup \{\omega\}$ and $x \in \mathcal{N}_S(u) \cup Inv_S(u)$.*
If $depnum_S(x, u) \geq 2$, then there exists $y \in \mathcal{N}_S(u) \cup \{u\}$ such that either $x \rightsquigarrow_S y$ or $x \searrow_S y$, and $y = u \vee dep(y, u) = depnum_S(x, u) - 1$ holds.
Furthermore, if also $depnum_S(x, u) \geq 3$ holds, then this $y \in Symbols(S)$, and if $x \searrow_S y(Y)$ holds then $y \rightsquigarrow_S z$ holds for some z such that $(z = u \in \mathcal{V}) \vee (z \in \mathcal{N}_S(u) \wedge depnum_S(z, u) = depnum_S(x, u) - 2 \wedge \neg(x \searrow_S z))$.

Proof. This follows immediately from the definition of a minimal-length u-dependence sequence for x in S. $\square$

3.7 Definition of u-similar and u-congruent linear schemas

Definition 38 (u-similar and u-congruent linear schemas) Let S, T be linear schemas and let $u \in \{\omega\} \cup \mathcal{V}$. Then $S\ simil_u\ T$ (S is u-similar to T) if and only if the following hold:

(1) $\mathcal{N}_S(u) = \mathcal{N}_T(u)$;

(2) $\mathcal{N}_S(u) \cap \mathit{ifPreds}(S) = \mathcal{N}_T(u) \cap \mathit{ifPreds}(T)$;

(3) $\mathcal{N}_S(u) \cap \mathit{whilePreds}(S) = \mathcal{N}_T(u) \cap \mathit{whilePreds}(T)$;

(4) $f \rightsquigarrow_S x\,(n) \wedge x \in \mathcal{N}_S(u) \iff f \rightsquigarrow_T x\,(n) \wedge x \in \mathcal{N}_T(u)$, for all $f \in \mathcal{F}$ and $n \geq 1$;

(5) $f \rightsquigarrow_S u \iff f \rightsquigarrow_T u$ if $u \in \mathcal{V}$ and $f \in \mathcal{F}$;

(6) $v \rightsquigarrow_S x\,(n) \iff v \rightsquigarrow_T x\,(n)$ for all $v \in \mathcal{V}$ and $x \in \mathcal{N}_S(u)$ and $n \geq 1$;

(7) $q \searrow_S p\,(Z) \iff q \searrow_T p\,(Z)$ if $u = \omega$ and $p \in \mathit{whilePreds}(S)$ and q is any predicate and $Z \in \{\mathsf{T}, \mathsf{F}\}$;

(8) $\mathit{Symbols}(\mathit{body}_S(p)) \cap \mathcal{N}_S(u) = \mathit{Symbols}(\mathit{body}_T(p)) \cap \mathcal{N}_T(u)$ if $p \in \mathit{whilePreds}(S)$;

(9) $\mathit{back}_S(p, f, x) \wedge x \in \mathcal{N}_S(u) \iff \mathit{back}_T(p, f, x) \wedge x \in \mathcal{N}_T(u)$;

(10) If $q \in \mathit{ifPreds}(S)$ and $Z \in \{\mathsf{T}, \mathsf{F}\}$ and $f \in \mathcal{F}$ and $x \in \mathcal{N}_S(u) \cup (\mathcal{V} \cap \{u\})$ then

$$\mathit{outif}_S(q, Z, f, x) \vee \mathit{thru}_S(q, Z, f, x) \iff \mathit{outif}_T(q, Z, f, x) \vee \mathit{thru}_T(q, Z, f, x);$$

(11) If $f, f' \in \mathit{Funcs}(S)$ and $f, f' \rightsquigarrow_S x\,(r)$ for $x \in \mathcal{N}_S(u) \cup (\{u\} \cap \mathcal{V})$, and $r \in \mathbb{N}$, and $\bar{S}$, $\bar{T}$ are the main subschemas of S and T respectively lying immediately above $\{f, f'\}$, then either $\neg(f \ll_{\bar{S}} f' \wedge f' \ll_{\bar{T}} f)$ holds, or there exists $q \in \mathit{ifPreds}(S)$ such that f and f' are q-competing for x in S;

(12) If $p \in \mathit{whilePreds}(S)$, $f \in \mathcal{F}$ and $f \rightsquigarrow_S x \wedge x \in \mathcal{N}_S(u)$ and $v = \mathit{assign}_S(f)$ and $w = \mathit{assign}_T(f)$, then

$$f \rightsquigarrow_{\mathit{body}_S(p)} v \wedge v \rightsquigarrow_{\mathit{body}_S(p)} x$$
$$\iff$$
$$f \rightsquigarrow_{\mathit{body}_T(p)} w \wedge w \rightsquigarrow_{\mathit{body}_T(p)} x$$

holds.

(13) If $p \in \mathit{whilePreds}(S)$, $q \in \mathit{ifPreds}(S)$, $f \in \mathit{Funcs}(S)$, $x \in \mathcal{N}_S(u)$, $Z \in \{\mathsf{T}, \mathsf{F}\}$ and $f \rightsquigarrow_S x$, with $v = \mathit{assign}_S(f)$ and $w = \mathit{assign}_T(f)$ and $v \rightsquigarrow_{\mathit{body}_S(p)} x$, then

$$\mathit{outif}_{\mathit{body}_S(p)}(q, Z, f, v) \vee \mathit{thru}_{\mathit{body}_S(p)}(q, Z, f, v)$$
$$\iff$$
$$\mathit{outif}_{\mathit{body}_T(p)}(q, Z, f, w) \vee \mathit{thru}_{\mathit{body}_T(p)}(q, Z, f, w)$$

holds.

If $S\,\mathit{simil}_u\,T$ and also $\mathbf{refvec}_S(x) = \mathbf{refvec}_T(x)$ for all $x \in \mathcal{N}_S(u)$ and $\mathit{assign}_S(f) = \mathit{assign}_T(f)$ for all $f \in \mathcal{N}_S(u) \cap \mathcal{F}$, then we say that S and T are u-congruent, written $S\,\mathit{cong}_u\,T$.

We also write $S\,\mathit{simil}_V\,T$ to mean that $S\,\mathit{simil}_u\,T$ for all $u \in V$, and $S\,\mathit{simil}\,T$ to mean that $S\,\mathit{simil}_{\mathcal{V} \cup \{\omega\}}\,T$ holds. Also $S\,\mathit{cong}_V\,T$ has a similar meaning.

Observe that the two linear predicate-free schemas

$$u := f();$$
$$v := g(u);$$

and

$$u' := f();$$

$$v := g(u');$$

are v-similar but not v-congruent if $u \neq u'$; thus congruence is a stronger condition than similarity.

There is some redundancy in the conditions of $S\,\mathit{simil}_u\,T$; Condition (1) can in fact be proved using Conditions (2), (3),(4),(8) and (10) using Proposition 31. Also Condition (12) is (we conjecture) a consequence of the others, but it seems convenient to state the definition of $S\,\mathit{simil}_u\,T$ in the form given. It is also worth mentioning that if Condition (5) is broadened to allow $f \in \mathcal{V}$, then the definition of $S\,\mathit{simil}_u\,T$ is unchanged, but to make this change in the definition of Definition 38 would make the induction proof of Lemma 51 more difficult.

Theorem 39 ($S\,\mathit{simil}_u\,T$ is decidable in polynomial time) *Given linear schemas S and T and $u \in \mathcal{V} \cup \{\omega\}$, it is decidable in polynomial time whether $S\,\mathit{simil}_u\,T$ holds.*

Proof. Given a linear schema S, encoded as indicated in Definition 4, with the braces $\{\ \}$, the truth of the relations $p \searrow_S x\,(Z)$ for each $p \in \mathit{Preds}(S)$, $x \in \mathit{Symbols}(S)$, $Z \in \{\mathsf{T}, \mathsf{F}\}$ can be established in polynomial time. Given two elements $v, w \in \mathit{alphabet}(S)$, with symbols v', w', we can decide in polynomial time whether w occurs immediately after v in any word in $\Pi(S)$, since this holds if and only if either w' occurs after v' in S without there being any other symbol between them, and $p \searrow_S v' \iff p \searrow_S w'$ for all $p \in \mathit{whilePreds}(S)$, or $v' \in \mathit{whilePreds}(S)$ and v' lies immediately above w' and there are no symbols occurring after v' in S before the closing brace $\}$ defined by v'. Thus we can construct in polynomial time a directed graph G_S, whose vertices are the elements of $\mathit{alphabet}(S)$ and such that there is an edge from vertex v to w in the graph G_S if and only if w occurs immediately after v in a word in $\Pi(S)$. Given $f \in \mathit{Funcs}(S)$ and $x \in \mathit{Symbols}(S)$, we can establish whether $f \rightsquigarrow_S x$ holds by deleting all vertices in G_S that are assignments to $\mathit{assign}_S(f)$ except the one with function symbol f or x, if $x \in \mathcal{F}$, and edges adjacent to deleted vertices, and establishing whether the letter containing x is reachable from the f-assignment in the resulting directed graph. This latter problem is well-known to be polynomial-time decidable in the size of G_S. The values of n for which $f \rightsquigarrow_S x\,(n)$ also holds can also be easily established, as can the truth of the assertions $v \rightsquigarrow_S x\,(n)$ for $v \in \mathcal{V}$ and $f \rightsquigarrow_S u$. Also, the truth of the relations *above* and $\ll$ for appropriate arguments can be decided in polynomial time by studying S. Having obtained this information, we can test the truth of the relations *back*, *outif*, *thru* (and hence the q-competing condition) for appropriate arguments. By comparing this information with that obtained from T and the graph G_T, it can be decided in polynomial time whether S and T satisfy $S\,\mathit{simil}_u\,T$. $\Box$

Lemma 40 (Replacing similar schemas by congruent schemas) *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-similar linear schemas. Then there are linear schemas S',T' which are u-equivalent and u-similar to S and T respectively and such that*

$S' \, cong_u \, T'$.

Proof. Let $x \in \mathcal{N}_S(u)$ and $r \leq arity(x)$ and assume that the rth component of $\mathbf{refvec}_S(x)$ is v, whereas the rth component of $\mathbf{refvec}_T(x)$ is $w \neq v$. Let

$$\Gamma_{assign}, \Gamma_{ref}(n) \subseteq \mathcal{N}_S(u)$$

for $n \in \mathbb{N}$ be the minimal sets satisfying the following two conditions; $x \in \Gamma_{ref}(r)$, and if $g \rightsquigarrow_S y\,(r')$ for $y \in \mathcal{N}_S(u)$, $r' \leq arity(y)$ and $assign_S(g) = v$, then $g \in \Gamma_{assign} \iff y \in \Gamma_{ref}(r')$ holds. Let v' be a variable not occurring in either S or T. Let $\hat{S}$ be the schema obtained from S by setting $assign_{S'}(h) = v'$ for all $h \in \Gamma_{assign}$ and changing the sth component of $\mathbf{refvec}_S(z)$ from v to v' for all $z \in \Gamma_{ref}(s)$. We similarly replace 'corresponding' occurrences of w in T by v' to get $\hat{T}$. Clearly $\hat{S} \, simil_u \, S$ and $\hat{S} \cong_u S$ holds, and the same relations hold for T and $\hat{T}$. We repeat this process for every $x \in \mathcal{N}_S(u)$. The resulting schemas satisfy the conditions stated. $\square$

3.8 Slices of schemas

An important special case of the equivalence problem for schemas S, T is that in which T is a *slice* of S.

Definition 41 A slice of a structured schema S may be obtained recursively by the following rules;

- if $S = S_1 S_2 S_3$ then $S_1 S_3$, $S_1 S_2$ and $S_2 S_3$ are slices of S;
- if T' is a slice of T then $while\, p(\mathbf{u})\, do\, T'$ is a slice of $while\, p(\mathbf{u})\, do\, T$;
- if T' is a slice of T then the if schema $if\ q(\mathbf{u})\ then\, S\, else\, T'$ is a slice of $if\ q(\mathbf{u})\ then\, S\, else\, T$ (the true and false parts may be interchanged in this example);
- a slice of a slice of S is itself a slice of S.

The following facts are easily proved. All slices of a linear schema are also linear. If a set $\Sigma \subseteq Symbols(S)$ (for linear S) satisfies $(x \in \Sigma \wedge p \searrow_S x) \Rightarrow p \in \Sigma$, then there is a unique slice T of S satisfying $Symbols(T) = \Sigma$; the slice T can be obtained from S by successively removing all assignments whose function symbols do not lie in Σ, and every if and while subschema of S whose guard does not lie in Σ.
A special case is given by $\Sigma = \mathcal{N}_S(V)$ for $V \subseteq \mathcal{V} \cup \{\omega\}$. In this case every slice T of S containing all symbols in $\mathcal{N}_S(V)$ satisfies $Inv_T(V) = Inv_S(V)$ and $S \, cong_V \, T$, since deletion from S of symbols not lying in $\mathcal{N}_S(V)$ does not affect the schema properties defining these statements. We will show in Part (2) of Theorem 42 that $S \cong_V T$ also holds.

A slice of an LFL schema need not be free or liberal; for example, the schema $while\, p(v)\, do\, \Lambda$, which is not free, is a slice of the LFL schema below;

while $p(v)$ *do*

$\{$

$u := h(u);$

$w := k(u);$

$v := g(v);$

$\}$

Also, deleting the assignment $u := h(u)$; gives a schema which is free but not liberal. However the slice of an LFL schema S which contains precisely the symbols in $\mathcal{N}_S(V)$ for any $V \subseteq \mathcal{V}$ is itself LFL; this follows from Theorem 26 and the 'backward data dependence' property of $\mathcal{N}_S(V)$.

Theorem 42 *Let S be a (not necessarily free or liberal) linear schema and let T be a slice of S. Let $u \in \mathcal{V} \cup \{\omega\}$, let i, j be interpretations differing only on predicates not lying in $\mathcal{N}_S(u)$, and let c, d be states such that $c(v) = d(v)$ for all $v \in Inv_S(u)$. Assume that T contains every symbol of $\mathcal{N}_S(u)$.*

(1) *If $Symbols(T) = \mathcal{N}_S(u)$, then $\mathcal{M}[\![S]\!]^i_c \neq \bot \Rightarrow \mathcal{M}[\![T]\!]^j_d \neq \bot$.*
(2) *If $u \in \mathcal{V}$ and $\mathcal{M}[\![S]\!]^i_c$ and $\mathcal{M}[\![T]\!]^j_d$ both terminate, then $\mathcal{M}[\![S]\!]^i_c(u) = \mathcal{M}[\![T]\!]^j_d(u)$; and if $u = \omega$ then $\mathcal{M}[\![S]\!]^i_c \neq \bot \iff \mathcal{M}[\![T]\!]^j_d \neq \bot$.*

In particular, $S \cong_u T$ holds.

Proof. Given any $\mu \in (alphabet(S))^\omega$, we define $\Delta(\mu)$ to be the word obtained by deleting every letter in μ whose symbol does not lie in $\mathcal{N}_S(u)$. Let T' be the slice of T (which hence is also a slice of S) that satisfies $Symbols(T') = \mathcal{N}_S(u)$. Observe that $\mathcal{N}_S(u) = \mathcal{N}_T(u) = \mathcal{N}_{T'}(u)$ and $Inv_S(u) = Inv_T(u) = Inv_{T'}(u)$; and $\Delta(\Pi(S)) = \Pi(T') = \Delta(\Pi(T))$ follows easily from the definition of a slice, using induction on $|S|$. Assume that $u \in \mathcal{V} \Rightarrow \pi_S(i, c) \in \Pi(S)$ holds. We will show that

$$\Delta(\pi_T(j, d)) \in pre(\Delta(\pi_S(i, c)))$$

holds. Suppose that this is false; thus we may write $\pi_S(i, c) = \sigma\rho$ such that

$$\Delta(\sigma)\, l \in pre(\Delta(\pi_T(j, d))) - pre(\Delta(\pi_S(i, e)))$$

for a letter $l \in alphabet(T')$ and a finite prefix σ. We will show that l is the first letter of ρ whose symbol is in $\mathcal{N}_S(u)$, proving $\Delta(\sigma)\, l \in pre(\Delta(\pi_S(i, c)))$ and so giving a contradiction. If ρ does not contain any letter whose symbol lies in $\mathcal{N}_S(u)$, and is finite, then $\Delta(\sigma) = \Delta(\pi_S(i, c)) \in \Delta(\Pi(S)) = \Pi(T')$, contradicting Part (2) of Lemma 12 applied to T'. If ρ does not contain such a letter, but is infinite, then ρ contains a while predicate, but no element of $\mathcal{N}_S(u)$. But this implies $u \in \mathcal{V}$, contradicting the finiteness of $\pi_S(i, c)$. Thus we may write $\rho = \rho_1 l' \rho_2$,

for a letter l' whose symbol is in $\mathcal{N}_S(u)$, unlike those of the letters in ρ_1, and so $\Delta(\sigma)l' \in \Delta(pre(\Pi(S))) = \Delta(pre(\Pi(T))) = \Pi(T')$; hence by Part (1) of Lemma 12 applied to T' we must have $l = <p = X>$ and $l' = <p = X'>$, for $p \in \mathcal{N}_S(u)$. Thus $\mathcal{M}[\![schema(\sigma\rho_1)]\!]_c(\mathbf{refvec}_S(p)) = \mathcal{M}[\![schema(\Delta(\sigma))]\!]_c(\mathbf{refvec}_S(p)) = \mathcal{M}[\![schema(\Delta(\sigma))]\!]_d(\mathbf{refvec}_S(p))$ using the conditions on c and d and the fact that $\sigma\rho_1 l'$ and $\Delta(\sigma l')$ contain the same set of Gp-segments for $G \in \mathcal{VF}^*$.
Clearly $\pi_T(j,d)$ has a prefix τl with $\Delta(\tau) = \Delta(\sigma\rho_1) = \Delta(\sigma)$. Since $p \in \mathcal{N}_S(u) = \mathcal{N}_T(u)$ holds, τ and $\Delta(\tau)$ contain the same set of Gp-segments for $G \in \mathcal{VF}^*$, and so

$$\mathcal{M}[\![schema(\tau)]\!]_d(\mathbf{refvec}_S(p)) = \mathcal{M}[\![schema(\Delta(\tau))]\!]_d(\mathbf{refvec}_T(p)),$$

hence

$$X' = p^i(\mathcal{M}[\![schema(\sigma\rho_1)]\!]_c(\mathbf{refvec}_S(p))) = p^j(\mathcal{M}[\![schema(\tau)]\!]_d(\mathbf{refvec}_T(p))) = X$$

and so $l = l'$ holds, giving the required contradiction. Thus we have shown $\Delta(\pi_T(j,d)) \in pre(\Delta(\pi_S(i,c)))$.
Observe that if $Symbols(T) = \mathcal{N}_S(u)$ then $T' = T$, and so $\Delta(\mu) = \mu$ for all $\mu \in \Pi^\omega(T)$, thus proving (1).
We now show (2). By interchanging S and T and using a similar argument to that used above, we can show that if $u \in \mathcal{V} \Rightarrow \pi_T(j,d) \in \Pi(T)$ holds then $\Delta(\pi_S(i,c)) \in pre(\Delta(\pi_T(j,d)))$ follows; thus if $u \in \mathcal{V} \Rightarrow (\pi_T(j,d) \in \Pi(T) \wedge \pi_S(i,c) \in \Pi(S))$ holds then

$$\Delta(\pi_T(j,d)) = \Delta(\pi_S(i,c))$$

follows. Thus if $u \in \mathcal{V}$ and $\pi_T(j,d))$ and $\pi_S(i,c)$ are both finite then $\pi_S(i,c)$ and $\pi_T(j,d)$ are Gu-segments for the same values of $G \in \mathcal{VF}^*$, hence

$$\mathcal{M}[\![schema(\pi_S(i,c))]\!]_c(u) = \mathcal{M}[\![schema(\pi_T(j,d))]\!]_d(u)$$

holds, proving $\mathcal{M}[\![S]\!]^i_c(u) = \mathcal{M}[\![T]\!]^j_d(u)$. If $u = \omega$, then Δ does not delete while predicate letters, and so $\pi_S(i,e)$ and $\pi_T(j,d)$ are both finite or both infinite; thus we have proved Part (2).
Thus we have shown $S \cong_u T$. $\square$

Part (1) of Theorem 42 may fail for a slice T whose symbol set strictly contains $\mathcal{N}_S(u)$; for example, if S is

$v = f();$
$while\, p(v)\, do\, \Lambda$

and T is the slice

$while\, p(v)\, do\, \Lambda$

for a variable $v \neq u$. If the interpretation i maps every predicate term $p(t)$ to T unless $t = f()$ then $\mathcal{M}[\![S]\!]^i_e$ terminates whereas $\mathcal{M}[\![T]\!]^i_e$ does not.

Theorem 55 is the main result of this section. The proof of this theorem procedes along the following lines. By Part (2) of Theorem 42, we may assume that u-similar linear schemas S and T contain only the symbols in $\mathcal{N}_S(u)$, since replacing schemas S and T by their respective slices containing only these symbols preserves u-similarity and congruence. We will prove that we may also assume that S and T are u-congruent. We will prove that we can make a further simplifying assumption about S and T, namely that there are no if predicates lying immediately below S or T. In order to do this, we have to define the schema $S/(q = Z)$ with the $\neg Z$-part of q deleted, where $q \in \mathit{ifPreds}(S)$ and $Z \in \{\mathsf{T}, \mathsf{F}\}$. In essence, a linear schema S which has m if predicates lying immediately below it may be thought of as a set of 2^m linear schemas, one for each choice from $\{\mathsf{T}, \mathsf{F}\}$ for each of these if predicates. We will show that u-congruence of S and T implies u-congruence for each of these schemas with its counterpart for T, and that any interpretation defines paths through S and T which make the same choice from $\{\mathsf{T}, \mathsf{F}\}$ in S and T for each of these if predicates. Thus we need to define the truncated schema $\mathit{trunc}_S\,(q)$, which is the 'bit' of S occurring before the if predicate q is reached, and prove that $\mathit{trunc}_S\,(q)$ and $\mathit{trunc}_T\,(q)$ are v-congruent for all variables v referenced by q. Thus we reduce the problem to the special case in which S and T are sequences of assignments and while schemas.

4.1 Deleting parts of if schemas

Definition 43 (deleting a part of an if schema) Let S be a linear schema and let $q \in \mathcal{P}$ and $Z \in \{\mathsf{T}, \mathsf{F}\}$. Assume that $q \in \mathit{ifPreds}(S)$ holds. The linear schema $S/(q = Z)$ is obtained by replacing the if subschema $S(q)$ of S by $\mathit{part}^Z_S(q)$.

Proposition 44 (characterising $\mathit{pre}(\Pi(S/(q = Z)))$) *Let S be a linear schema. Let $q \in \mathit{ifPreds}(S)$ and $Z \in \{\mathsf{T}, \mathsf{F}\}$. Let $\mu \in \mathit{alphabet}(S)^*$. Then $\mu \in \mathit{pre}(\Pi(S/(q = Z)))$ if and only if there exists $\rho \in \mathit{pre}(\Pi(S))$ not passing through $<\!q = \neg Z\!>$ and such that μ is obtained from ρ by deleting all occurrences of $<\!q = Z\!>$.*
$\square$

Lemma 45 (relating S to $S/(q = Z)$) *Let S be a linear schema. Let $q \in \mathit{ifPreds}(S)$ and $Z \in \{\mathsf{T}, \mathsf{F}\}$. Then the following hold.*

(1) *Let $f \in \mathit{Funcs}(S)$ and $y \in \mathit{Symbols}(S/(q = Z)) \cup \mathcal{V}$. Then*

$$f \rightsquigarrow_{S/(q=Z)} y \iff f \rightsquigarrow_S y \wedge \neg(\mathit{outif}_S(q, \neg Z, f, y) \vee \mathit{thru}_S(q, \neg Z, f, y))$$

holds.

(2) *Let $y \in \mathit{Symbols}(S/(q = Z))$, $Y \in \{\mathsf{T}, \mathsf{F}\}$ and $p \in \mathit{Preds}(S) - \{q\}$. Then $p \searrow_{S/(q=Z)} y\,(Y) \iff p \searrow_S y\,(Y)$.*

(3) *Let* $f \in \mathit{Funcs}(S)$ *and* $y \in \mathit{Symbols}(S/(q=Z)) \cup \mathcal{V}$. *Assume* $f \rightsquigarrow_{S/(q=Z)} y$ *holds. Let* $p \in \mathit{ifPreds}(S) - \{q\}$ *and* $Y \in \{\mathsf{T}, \mathsf{F}\}$. *Then*

$$\mathit{outif}_S(p, Y, f, y) \Rightarrow \mathit{outif}_{S/(q=Z)}(p, Y, f, y)$$

and

$$\mathit{thru}_S(p, Y, f, y) \Rightarrow \mathit{thru}_{S/(q=Z)}(p, Y, f, y)$$

hold, and if $\mathit{above}_S(q) = q$ *and* $\mathit{above}_S(p) \neq p$, *then the converse statements are true.*

Proof.

(1) Assume that $f \rightsquigarrow_{S/(q=Z)} y$ holds. Thus $\neg(q \searrow_S f(\neg Z))$ and so $\neg\mathit{outif}_S(q, \neg Z, f, y)$ holds. Let μ be an fy-prefix in $S/(q=Z)$. By Proposition 44, there is a prefix ρ in S not passing through $<\!q = \neg Z\!>$ such that μ is obtained from ρ by deleting all occurrences of $<\!q = Z\!>$; hence ρ is an fy-prefix in S and so $f \rightsquigarrow_S y \wedge \neg\mathit{thru}_S(q, \neg Z, f, y)$ holds.
Conversely, assume $f \rightsquigarrow_S y \wedge \neg(\mathit{outif}_S(q, \neg Z, f, y) \vee \mathit{thru}_S(q, \neg Z, f, y))$ holds. By the hypotheses, $y \neq q \wedge \neg(q \searrow_S y(\neg Z))$ and hence $\neg(q \searrow_S f(\neg Z))$ holds. Thus there is an fy-prefix ρ in S which does not pass through $<\!q = \neg Z\!>$. By Proposition 44, deleting all occurrences of $<\!q = Z\!>$ from ρ gives an fy-segment in $S/(q = Z)$.

(2) This follows easily from the definition of $S/(q = Z)$.

(3) We have four cases.
- Assume $\mathit{outif}_S(p, Y, f, y)$ holds. Thus $p \searrow_S f(Y)$ and since $f \in \mathit{Funcs}(S/(q = Z))$ by the hypotheses, $p \searrow_{S/(q=Z)} f(Y)$ by Part (2) of this Lemma; in particular, $p \in \mathit{Symbols}(S/(q = Z))$. Assume that $\neg\mathit{outif}_{S/(q=Z)}(p, Y, f, y)$; thus since $f \rightsquigarrow_{S/(q=Z)} y$ holds by the hypotheses, $f \rightsquigarrow_{S/(q=Z)(p,Y)} y$ holds. If $\neg(p \searrow_S q(Y))$ then $S/(q = Z)(p, Y) = \mathit{part}^Y_S(p)$, and if $p \searrow_S q(Y)$ then $S/(q = Z)(p, Y) = \mathit{part}^Y_S(p)/(q = Z)$ can be easily shown. Thus in either case $f \rightsquigarrow_{\mathit{part}^Y_S(p)} y$ holds, contradicting $\mathit{outif}_S(p, Y, f, y)$.
- Now assume $\mathit{thru}_S(p, Y, f, y)$ holds. Thus

$$f, x \notin \mathit{Symbols}(S(p)) \supseteq \mathit{Symbols}(S/(q = Z)(p)).$$

  If $\neg\mathit{thru}_{S/(q=Z)}(p, Y, f, y)$ holds, then since $f \rightsquigarrow_{S/(q=Z)} y$ holds, there is an fy-prefix μ in the schema $S/(q = Z)$ which does not pass through $<\!p = Y\!>$. Thus by Proposition 44, there is a prefix ρ in S not passing through $<\!q = \neg Z\!>$ such that μ is obtained from ρ by deleting all occurrences of $<\!q = Z\!>$; hence ρ is an fy-prefix in S and so $f \rightsquigarrow_S y \wedge \neg\mathit{thru}_S(q, \neg Z, f, y)$ holds. Since $q \neq p$, ρ also does not pass through $<\!p = Y\!>$, contradicting $\mathit{thru}_S(p, Y, f, y)$.
- Assume $\neg\mathit{outif}_S(p, Y, f, y) \wedge \mathit{outif}_{S/(q=Z)}(p, Y, f, y)$ and $\mathit{above}_S(q) = q$ and $\mathit{above}_S(p) \neq p$ hold. Thus $p \searrow_{S/(q=Z)} f(Y)$ and so $p \searrow_S f(Y)$ by Part (2) of this Lemma. Hence there is an fy-segment μ in the schema $\mathit{part}^Y_S(p)$. Clearly $\neg(p \searrow_S q)$ holds under the hypotheses given, and since $p \in \mathit{Preds}(S/(q = Z))$, μ is also an an fy-segment in $S/(q = Z)$, giving a contradiction.

- Assume $\neg thru_S(p, Y, f, y) \wedge thru_{S/(q=Z)}(p, Y, f, y)$ and $above_S(q) = q$ and $above_S(p) \neq p$ hold. Let $above_S(p) = p' \neq p$. Clearly $p' \in whilePreds(S)$. Since $p \in ifPreds(S/(q = Z))$ and $above_S(q) = q \neq p'$, $S(p')$ is a while schema in $S/(q = Z)$; thus since
$$thru_{S/(q=Z)}(p, Y, f, y)$$
holds, either $p' \searrow_S f$ or $p' \searrow_S y$ holds, since otherwise an fy-segment in $S/(q = Z)$ need never enter the body of p', contradicting $thru_{S/(q=Z)}(p, Y, f, y)$. If both hold then $thru_{S(p')}(p, Y, f, y)$ and hence $thru_S(p, Y, f, y)$ follows, giving a contradiction. If $p' \searrow_S f \wedge \neg(p' \searrow_S y)$ holds, then
$$thru_{S(p')}(p, Y, f, assign_S(f))$$
holds, and if $p' \searrow_S y \wedge \neg(p' \searrow_S f)$ holds, then
$$thru_{S(p')}(p, Y, assign_S(f), y)$$
holds, in both cases contradicting $\neg thru_S(p, Y, f, y)$.

$\square$

Corollary 46 (iterating Lemma 45) *Let S be a linear schema. Let $\{q(1), \ldots, q(m)\}$ be a set of if predicates in S which all lie immediately below S and let $Z(1), \ldots, Z(m) \in \{\mathsf{T}, \mathsf{F}\}$. Define $S' = S/(q(1) = Z(1))/\ldots/(q(m) = Z(m))$. Then the following hold.*

(1) *Let $f \in Funcs(S)$ and $y \in Symbols(S') \cup \mathcal{V}$. Then*
$$f \rightsquigarrow_{S'} y$$
$$\Longleftrightarrow$$
$$f \rightsquigarrow_S y \wedge (\bigwedge_{i \leq m} \neg(outif_S(q(i), \neg Z, f, y) \vee thru_S(q(i), \neg Z, f, y)))$$
holds.

(2) *Let $y \in Symbols(S')$, $Y \in \{\mathsf{T}, \mathsf{F}\}$ and $p \in Preds(S) - \{q(1), \ldots, q(m)\}$. Then $p \searrow_{S'} y\,(Y) \iff p \searrow_S y\,(Y)$.*

(3) *Let $f \in Funcs(S)$ and $y \in Symbols(S') \cup \mathcal{V}$. Assume $f \rightsquigarrow_{S'} y$ holds. Let $p \in ifPreds(S) - \{q(1), \ldots, q(m)\}$ and $Y \in \{\mathsf{T}, \mathsf{F}\}$. Then*
$$outif_S(p, Y, f, y) \Rightarrow outif_{S'}(p, Y, f, y)$$
and $thru_S(p, Y, f, y) \Rightarrow thru_{S'}(p, Y, f, y)$ hold, and if $above_S(p) \neq p$, then the converse statements are true.

Proof. All parts follow straightforwardly from Lemma 45, using induction on m. $\square$

Theorem 47 (u-similarity is inherited by part deletion) *Let S, T be linear schemas and assume $S\ simil_u\ T$ for $u \in \mathcal{V}\cup\{\omega\}$. Assume $Symbols(S) = Symbols(T) = \mathcal{N}_S(u)$. Let $q(1), \ldots, q(m)$ be the set of all if predicates in $\mathcal{N}_S(u)$ lying immediately below S and let $Z(1), \ldots, Z(m) \in \{\mathsf{T}, \mathsf{F}\}$. Define*

$$S' = S/(q(1) = Z(1))/\ldots/(q(m) = Z(m))$$

and define T' similarly. Then $S'\ simil_u\ T'$ holds.

Proof. Let $x \in \mathcal{N}_{S'}(u)$. Then $x \in \mathcal{N}_{T'}(u)$ follows from Parts (1) and (2) of Corollary 46 and Proposition 37, Condition (10) of $S\ simil_u\ T$ and Proposition 31, using induction on $depnum_S\,(x, u)$. Thus $\mathcal{N}_{S'}(u) \subseteq \mathcal{N}_{T'}(u)$ holds. By interchanging S and T and using Part (1) of $S\ simil_u\ T$ we get $\mathcal{N}_{S'}(u) = \mathcal{N}_{T'}(u)$, thus proving Condition (1) of $S'\ simil_u\ T'$. Conditions (2)–(7) of $S'\ simil_u\ T'$ follow at once from this, using Corollary 46. Condition (8),(9), (12) and (13) of $S'\ simil_u\ T'$ follow from the fact that if $p \in whilePreds(S')$ then the while schema $S'(p) = S(p)$. Condition (10) of $S'\ simil_u\ T'$ is given by Corollary 46, Part (3) and Condition (10) of $S\ simil_u\ T$. Lastly, Condition (11) follows from Condition (10) of $S'\ simil_u\ T'$ and the fact that $f \ll_{S'} g \iff f \ll_S g$ for any $f, g \in Funcs(S')$, and Part (3) of Corollary 46. $\Box$

4.2 Truncated schemas

Definition 48 (q-truncated schema) Let S be a linear schema and let $q \in ifPreds(S)$ lie immediately below S. We define the schema $trunc_S\,(q)$ as follows. Let $\{p(1), \ldots, p(m)\}$ be the set of all if predicates containing q in one part; say $p(i) \searrow_S q\,(Z(i))$ for $Z(i) \in \{\mathsf{T}, \mathsf{F}\}$. Define $S' = S/(p(1) = Z(1))/\ldots/(p(m) = Z(m))$. Write $S' = S''TS'''$ where T is the if subschema of S guarded by q. Then define $trunc_S\,(q) = S''$.

Observe that for any linear schema S, $Symbols(trunc_S\,(q)) = \{x \in Symbols(S) | x < <_S q\}$; and if $x \in Symbols(S)$ then $f \rightsquigarrow_{trunc_S\,(q)} x\,(r) \iff (f \rightsquigarrow_S x\,(r) \wedge x \ll_S q)$ and $x \searrow_{trunc_S\,(q)} y\,(Z) \iff (x \searrow_S y\,(Z) \wedge x \ll_S q)$ hold.

Lemma 49 (inheritance of congruence in truncated schemas) *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-congruent linear schemas. Assume $Symbols(S) = Symbols(T) = \mathcal{N}_S(u)$. Let $q \in ifPreds(S)$ lie immediately below S and let $v \in Refset_S(q)$. Then $trunc_S\,(q)\, cong_v\, trunc_T\,(q)$ holds.*

Proof. Write $S' = trunc_S\,(q)$ and define T' similarly. We prove only $\mathcal{N}_{S'}(v) = \mathcal{N}_{T'}(v)$; all other conditions of $S'\ simil_v\ T'$ follow easily from this; hence $S'\ simil_v\ T'$ and so $S'\ cong_v\ T'$ follows.
Let $x \in \mathcal{N}_{S'}(v)$. We show $x \in \mathcal{N}_{T'}(v)$ by induction on $depnum_{S'}\,(x, v)$; the result will then follow by interchanging S and T.
If $x \in \mathcal{F}$ then $x \rightsquigarrow_{S'} y$ for some $y \in \mathcal{N}_{S'}(v)\cup\{v\}$ with $depnum_{S'}\,(y, v) < depnum_{S'}\,(x, v)$.

If $y = v$ then $x \rightsquigarrow_{T'} y$ and so $x \in \mathcal{N}_{T'}(v)$ is easily shown. If $y \in \mathcal{N}_{S'}(v)$ then $x \rightsquigarrow_S y$ and $y \in \mathcal{N}_{T'}(v)$ by the inductive hypothesis applied to y and $y \ll_T q$ holds; thus $x \rightsquigarrow_T y$ since $S\, cong_u\, T$ and so $x \rightsquigarrow_{T'} y$ follows, proving $x \in \mathcal{N}_{T'}(v)$.
If $x \in \mathcal{P}$ then $x \searrow_{S'} y$ for some $y \in \mathcal{N}_{S'}(v)$ with $depnum_{S'}(y, v) < depnum_{S'}(x, v)$; and by Proposition 37, $y \rightsquigarrow_{S'} z$ for some $z \in \{v\} \cup \mathcal{N}_{S'}(v)$ with $depnum_{S'}(z, v) < depnum_{S'}(y, v)$ and $\neg(y \searrow_{S'} z)$. Thus $y \in \mathcal{N}_{T'}(v)$ by the inductive hypothesis applied to y. If $x \in whilePreds(S')$ then $x \searrow_{T'} y$ follows from Condition (8) of $S\, simil_u\, T$, so assume $x \in ifPreds(S')$. If $z = v$ then $outif_S(x, X, y, q)$ for some $X \in \{\mathsf{T}, \mathsf{F}\}$ and so by Condition (10) of $S\, simil_u\, T$, $outif_T(x, X, y, q) \vee thru_T(x, X, y, q)$ holds; hence by Proposition 31, $outif_T(x, Y, y', q)$ for some $y' \in \mathcal{F}$ and $Y \in \{\mathsf{T}, \mathsf{F}\}$ with $depnum_{S'}(y', v) \leq depnum_{S'}(y, v)$, and so $y' \in \mathcal{N}_{T'}(v)$ by the inductive hypothesis and so $x \in \mathcal{N}_{T'}(v)$ follows. If instead $z \in \mathcal{N}_{S'}(v)$ then $x \in \mathcal{N}_{T'}(v)$ can be proved similarly. $\square$

4.3 Relating bodies of u-similar while schemas

Lemma 50 (relating while schemas to their bodies) *Let the linear schema $S =$ while $p(\mathbf{x})$ do S'. Then the following hold.*

(1) *$v \rightsquigarrow_{S'} z \iff v \rightsquigarrow_S z$ and $f \rightsquigarrow_{S'} v \iff f \rightsquigarrow_S v$ hold for all $v \in \mathcal{V}$, $z \in Symbols(S')$ and $f \in Funcs(S')$.*
(2) *$f \rightsquigarrow_{S'} z \iff (f \rightsquigarrow_S z \wedge \neg back_S(p, f, z))$ for all $f \in \mathcal{F}$ and $z \in Symbols(S')$.*
(3) *Let $u \in \mathcal{V} \cup \{\omega\}$; then $\mathcal{N}_{S'}(Inv_S(u)) = \mathcal{N}_S(u) - \{p\}$ and $Inv_{S'}(Inv_S(u)) \subseteq Inv_S(u)$.*

Proof. We prove only (3); (1) and (2) are straightforward.
Let $v \in Inv_S(u)$ and $x \in \mathcal{N}_{S'}(v)$. Clearly $x \neq p$. We show $x \in \mathcal{N}_S(u)$ by induction on $depnum_{S'}(x, v)$. If $depnum_{S'}(x, v) = 2$ then $x \rightsquigarrow_S v$ and either $v = u$ (and so $x \in \mathcal{N}_S(v) = \mathcal{N}_S(u)$) or $v \rightsquigarrow_S y$ for $y \in \mathcal{N}_S(u)$; hence $x \rightsquigarrow_S y$ and so $x \in \mathcal{N}_S(u)$. If $depnum_{S'}(x, v) \geq 3$ then by Proposition 37 $x \rightsquigarrow_{S'} y \vee x \searrow_{S'} y$ for some $y \in \mathcal{N}_{S'}(v)$ with $depnum_{S'}(y, v) < depnum_{S'}(x, v)$; thus $x \in \mathcal{N}_S(u)$ follows by the inductive hypothesis applied to y. Thus we have shown $\mathcal{N}_S(u) - \{p\} \supseteq \mathcal{N}_{S'}(Inv_S(u))$.
Conversely, let $x \in \mathcal{N}_S(u) - \{p\}$. We show $x \in \mathcal{N}_{S'}(Inv_S(u))$ by induction on $depnum_S(x, u)$. If $x \rightsquigarrow_S u \in \mathcal{V}$ then this is immediate since $u \in Inv_S(u)$. If not, then $depnum_S(x, u) \geq 2$ and so by Proposition 37 $x \rightsquigarrow_S y \vee x \searrow_S y$ for some $y \in \mathcal{N}_S(u)$ with $depnum_S(y, u) < depnum_S(x, u)$. If $y = p \vee back_S(p, x, y)$ then $x \rightsquigarrow_S v$ for some $v \in Refset_S(y)$ and $v \rightsquigarrow_S y$, hence $v \in Inv_S(u)$ and so $x \in \mathcal{N}_{S'}(Inv_S(u))$. Lastly, if $x \searrow_S y$ or $x \rightsquigarrow_{S'} y$ then $x \in \mathcal{N}_{S'}(Inv_S(u))$ follows from the inductive hypothesis applied to y.
Thus we have shown $\mathcal{N}_S(u) - \{p\} = \mathcal{N}_{S'}(Inv_S(u))$. Since $v \in Inv_{S'}(Inv_S(u))$ if and only if $v \rightsquigarrow_{S'} v \in Inv_S(u) \vee v \rightsquigarrow_S x \in \mathcal{N}_{S'}(Inv_S(u))$ and $v \in Inv_S(u)$ if and only if $v \rightsquigarrow_{S'} v \in Inv_S(u) \vee v \rightsquigarrow_S x \in \mathcal{N}_S(u)$, we get $Inv_{S'}(Inv_S(u)) \subseteq Inv_S(u)$. $\square$

Lemma 51 (congruent while schemas have congruent bodies) *Let* $u \in \mathcal{V} \cup \{\omega\}$ *and let* $S = \mathit{while}\, p(\mathbf{x})\, \mathit{do}\, S'$ *and* $T = \mathit{while}\, p(\mathbf{x})\, \mathit{do}\, T'$ *be linear schemas. Assume* $S\, \mathit{cong}_u\, T$. *Then* $S'\, \mathit{cong}_{\mathit{Inv}_S(u)}\, T'$ *holds. Also, if* $u = \omega$ *then* $S'\, \mathit{cong}_\omega\, T'$ *holds.*

Proof. Let $v \in \mathit{Inv}_S(u)$. We will show that $S'\, \mathit{simil}_v\, T'$ holds. Part (3) of Lemma 50 proves $\mathcal{N}_{S'}(v) \cup \mathcal{N}_{T'}(v) \subseteq \mathcal{N}_S(u) = \mathcal{N}_T(u)$. Thus $x \in \mathcal{N}_{S'}(v) \iff x \in \mathcal{N}_{T'}(v)$ can be proved by induction on $\mathit{depnum}_{S'}(x, v)$ using Proposition 31, Part (2) of Lemma 50 and Conditions (1),(9) of $S\, \mathit{simil}_u\, T$, proving Conditions (1) and hence (4) and (6) of $S'\, \mathit{simil}_v\, T'$. All other conditions of $S'\, \mathit{simil}_v\, T'$ except (5) and (10) follow straightforwardly from this.
We now prove Condition (5) of $S'\, \mathit{simil}_v\, T'$. Assume that $f \rightsquigarrow_{S'} v$ for $f \in \mathcal{F}$. Thus $f \in \mathcal{N}_{S'}(v) \subseteq \mathcal{N}_S(u)$. We will show $f \rightsquigarrow_{T'} v$. If $v = u$ then $f \rightsquigarrow_{T'} v$ is immediate. If not, then $v \rightsquigarrow_S y$ for $y \in \mathcal{N}_S(u)$; hence $f \rightsquigarrow_{T'} v$ follows from Condition (12) of $S\, \mathit{simil}_u\, T$ and so Condition (5) of $S'\, \mathit{simil}_v\, T'$ follows by interchanging S and T.
Lastly we show Condition (10) of $S'\, \mathit{simil}_v\, T'$. Assume
$\mathit{outif}_{S'}(q, Z, f, x) \vee \mathit{thru}_{S'}(q, Z, f, x)$ holds for $q \in \mathit{ifPreds}(S)$ and $Z \in \{\mathsf{T}, \mathsf{F}\}$ and $f \in \mathcal{F}$ and $x \in \mathcal{N}_S(v) \cup \{v\}$. If $x \in \mathcal{N}_{S'}(v)$ then $\mathit{outif}_{T'}(q, Z, f, x) \vee \mathit{thru}_{T'}(q, Z, f, x)$ follows from Condition (10) of $S\, \mathit{simil}_u\, T$. If not, then $x = v$ holds, and so $v \rightsquigarrow_S y$ for some $y \in \mathcal{N}_S(u)$; thus $\mathit{outif}_{T'}(q, Z, f, x) \vee \mathit{thru}_{T'}(q, Z, f, x)$ follows from Condition (12) and (13) of $S\, \mathit{simil}_u\, T$. Thus Condition (10) of $S'\, \mathit{cong}_v\, T'$ follows by interchanging S and T.
Hence we have shown $S'\, \mathit{simil}_v\, T'$ and $\mathcal{N}_{S'}(v) = \mathcal{N}_{T'}(v) \subseteq \mathcal{N}_S(u) = \mathcal{N}_T(u)$ for every $v \in \mathit{Inv}_S(u)$ and so $S'\, \mathit{simil}_{\mathit{Inv}_S(u)}\, T'$ and hence $S'\, \mathit{cong}_{\mathit{Inv}_S(u)}\, T'$ holds.
It is a similar exercise to show that $S\, \mathit{cong}_\omega\, T \Rightarrow S'\, \mathit{cong}_\omega\, T'$ holds. $\square$

Lemma 52 (schema body equivalence implies schema equivalence) *Let* $S = \mathit{while}\, p(\mathbf{x})\, \mathit{do}\, S'$ *and* $T = \mathit{while}\, p(\mathbf{x})\, \mathit{do}\, T'$ *be linear schemas. Let* $V \subseteq \mathcal{V}$ *and assume*

$$\mathit{Inv}_{S'}(V) \cup \mathit{Inv}_{T'}(V) \cup \mathit{Refset}_S(p) \subseteq V$$

holds. Assume that $S' \cong_V T'$.

(1) *Then* $S \cong_V T$ *holds.*
(2) *If* $\mathit{Inv}_{S'}(\omega) \cup \mathit{Inv}_{T'}(\omega) \subseteq V$ *and* $S' \cong_\omega T'$ *holds then* $S \cong_\omega T$ *holds.*

Proof.

(1) Let i be an interpretation and assume that $\mathcal{M}[\![S]\!]^i_e$ and $\mathcal{M}[\![T]\!]^i_e$ both terminate. Suppose that $<p = \mathsf{T}>$ occurs exactly n times in the path $\pi_S(i, e)$. By induction on r, and using Part (2) of Theorem 42 and the fact that $\mathit{Inv}_{S'}(V) \cup \mathit{Inv}_{T'}(V) \subseteq V$, $\mathcal{M}[\![(S')^r]\!]^i_e(v) = \mathcal{M}[\![(T')^r]\!]^i_e(v)$ for all $v \in V$ and $r \in \mathbb{N}$, and hence since $\mathit{Refset}_S(p) \subseteq V$, $<p = \mathsf{T}>$ occurs exactly n times in $\pi_T(i, e)$. Thus $\mathcal{M}[\![S]\!]^i_e = \mathcal{M}[\![(S')^{n+1}]\!]^i_e(v) = \mathcal{M}[\![(T')^{n+1}]\!]^i_e(v) = \mathcal{M}[\![T]\!]^i_e(v)$ for all $v \in V$.
(2) Similar to Part (1) of this Lemma.

$\square$

Lemma 53 *Let S be a linear schema and let $u \in \mathcal{V} \cup \{\omega\}$. Suppose T is another linear schema and $S\,cong_u\,T$ holds, and $Symbols(S) = Symbols(T) = \mathcal{N}_S(u) = \mathcal{N}_T(u)$, and $S = S_1 \ldots S_m$ and $T = T_1 \ldots T_n$ holds such that each S_j and T_j is either a while schema or an assignment. Then $m = n$ and there is a permutation χ of the set $\{1, \ldots, m\}$ such that $Symbols(S_j) = Symbols(T_{\chi(j)})$, and if S_j is an assignment, then $T_{\chi(j)} = S_j$ for all j, and if any S_j is a while schema then so is $T_{\chi(j)}$, with the same guard; and $u \in \mathcal{V} \Rightarrow \chi(m) = m$ holds.*

Proof. The existence of χ linking subschemas of the same type and symbol set, implying $m = n$, follows straightforwardly from the congruence definition. To show $u \in \mathcal{V} \Rightarrow \chi(m) = m$, assume that $u \in \mathcal{V}$ and $\chi(m) = l < m$. Since the symbols of $T_{l+1} \ldots T_m$ all lie in $\mathcal{N}_S(u)$, $T_{l+1} \ldots T_m$ must contain an assignment $u := f(\mathbf{x})$ such that $f \rightsquigarrow_T u$. Similarly S_m (and hence T_l) contains an assignment $u := g(\mathbf{y})$ with $f \neq g$ such that $g \rightsquigarrow_S u$. Thus $u := f(\mathbf{x})$ occurs in $S_1 \ldots S_{m-1}$. By Conditions (5), (10) and (11) of $S\,simil_u\,T$, there is some $q \in \mathit{ifPreds}(S) \cap \mathcal{N}_S(u)$ such that f and g are q-competing for u. But q would have to occur in S_m and $T_{l+1} \ldots T_m$, clearly contradicting the other conditions on χ, giving a contradiction. $\square$

Lemma 54 (replacing similar schemas by congruent schemas) *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-similar linear schemas. Then there are linear schemas S', T' which are u-equivalent and u-similar to S and T respectively and such that $S'\,cong_u\,T'$.*

Proof. Assume that S and T are not already u-congruent. Then there exists $x \in \mathcal{N}_S(u)$ and $r \leq arity(x)$ such that the rth component of $\mathbf{refvec}_S(x)$ is v, whereas the rth component of $\mathbf{refvec}_T(x)$ is $w \neq v$; or a similar situation holds in which S and T differ on an assigned variable; this case can be treated analogously. We will proceed by replacing v and w in S and T respectively by a new variable v' to give schemas $\hat{S}$, $\hat{T}$ which are 'more congruent' than S and T are, but in order to do this, we must replace v and w not just in $\mathbf{refvec}_S(x)$ and $\mathbf{refvec}_T(x)$, but at other points in both schemas which are affected by the replacement in $\mathbf{refvec}_S(x)$ and $\mathbf{refvec}_T(x)$. Thus we define symbol sets

$$\Gamma_{assign}, \Gamma_{ref}(n) \subseteq \mathcal{N}_S(u)$$

for $n \in \mathbb{N}$ which give, respectively, the symbols at which the assigned variable and th nth referenced variable must be changed from v or w to v'. Γ_{assign} and $\Gamma_{ref}(n)$ are defined to be the minimal sets satisfying $x \in \Gamma_{ref}(r)$ and the following closure condition; if $g \rightsquigarrow_S y\,(r')$ for $y \in \mathcal{N}_S(u)$, $r' \leq arity(y)$ and $assign_S(g) = v$, then $g \in \Gamma_{assign} \iff y \in \Gamma_{ref}(r')$ holds. Let v' be a variable not occurring in either S or T. Let $\hat{S}$ be the schema obtained from S by setting $assign_{S'}(h) = v'$ for all $h \in \Gamma_{assign}$ and changing the sth component of $\mathbf{refvec}_S(z)$ from v to v' for all $z \in \Gamma_{ref}(s)$. We similarly replace 'corresponding' occurrences of w in T by v' to get $\hat{T}$. $\hat{S}\,simil_u\,S$ follows from the conditions of Definition 38, since the relation $\rightsquigarrow_S$ is unaffected by this replacement, and similarly $\hat{S} \cong_u S$ follows by observing that for every interpretation i, the path $\pi_{\hat{S}}(i, e)$ can be obtained from $\pi_S(i, e)$ by replacing some occurrences of

v by v'; and the same relations hold for T and $\hat{T}$. We repeat this process for every $x \in \mathcal{N}_S(u)$. The resulting schemas satisfy the conditions stated. $\square$

Theorem 55 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S and T be u-similar linear schemas. Then S and T are u-equivalent.*

Proof. This follows by induction on $|S| + |T|$. By Lemma 54 we may assume that $S\,cong_u\,T$, and by Part (2) of Theorem 42, and the transitivity of u-congruence, we may assume that $Symbols(S) = Symbols(T) = \mathcal{N}_S(u) = \mathcal{N}_T(u)$, since S and T may be replaced by their slices containing only the symbols in $\mathcal{N}_S(u)$. Let i be an interpretation and assume that if $u \in \mathcal{V}$ then for each $U \in \{S, T\}$, $\pi_U(i, e)$ is finite. We will show that $\mathcal{M}[\![S]\!]^i_e(u) = \mathcal{M}[\![T]\!]^i_e(u)$. Let Q be the set of all $q \in ifPreds(S)$ for which $above_S(q) = q$. For each $q \in Q$ and $U \in \{S, T\}$, if $\pi_U(i, e)$ passes through q then the prefix of $\pi_U(i, e)$ preceding the sole occurrence of q contains precisely the same sequence of assignments as $\pi_{trunc_U\,(q)}(i, e)$. Thus by Lemma 49 and the inductive hypothesis applied to $trunc_S\,(q)$ and $trunc_T\,(q)$, for each $q \in Q$ there exists $Z(q) \in \{\mathsf{T}, \mathsf{F}\}$ such that neither $\pi_S(i, e)$ nor $\pi_T(i, e)$ passes through $\lessdot q = \neg Z(q) \gtrdot$. Let S' be obtained from S by deleting the $Z(q)$-part of each $q \in Q$ and define T' similarly. For each $U \in \{S, T\}$, let $\mu(U) \in alphabet(U)^*$ be obtained from $\pi_U(i, e)$ by deleting all occurrences of $\lessdot q = Z(q) \gtrdot$ for $q \in Q$. Thus each $\mu(U) = \pi_{U'}(i, e)$ and so if $Q \neq \emptyset$ then $\mathcal{M}[\![S]\!]^i_e(u) = \mathcal{M}[\![T]\!]^i_e(u)$ follows from Theorem 47 and the inductive hypothesis applied to S' and T'. Hence we may assume that $Q = \emptyset$.
Thus we may write $S = S_1 \ldots S_m$ and $T = T_1 \ldots T_n$ such that each S_j and each T_j is either a while schema or an assignment.
By Lemma 53, $m = n$ holds. We now consider two cases.

- Suppose $u \in \mathcal{V}$. By Lemma 53,

  $$Symbols(S_m) = Symbols(T_m) \neq \emptyset$$

  holds. Let $V = Inv_{S_m}(u)$. Clearly $V = Inv_{T_m}(u)$, $S_m\,cong_u\,T_m$ and

  $$S_1 \ldots S_{m-1}\,cong_V\,T_1 \ldots T_{m-1}$$

  can be proved using $S\,cong_u\,T$. Thus $S_1 \ldots S_{m-1} \cong_V T_1 \ldots T_{m-1}$ follows from the inductive hypothesis, and $S_m \cong_u T_m$ holds, using Lemma 51 and Part (1) of Lemma 52 if S_m and T_m are while schemas. Thus $S \cong_u T$ follows from Part (2) of Theorem 42.
- Suppose $u = \omega$. We may assume (after interchanging S and T if necessary) that for some $k \leq m$, the path $\pi_S(i, e)$ reaches S_k and fails to terminate in S_k. Thus it suffices to prove that $\pi_T(i, e)$ is infinite. By Lemma 53, there is a schema T_l such that

  $$Symbols(S_k) = Symbols(T_l) \neq \emptyset$$

  holds. Let $V = Inv_{S_k}(\omega)$. As in Case (1), it is easy to show that $V = Inv_{T_l}(\omega)$, $S_k \cong_\omega T_l$, $S_1 \ldots S_{k-1}\,cong_V\,T_1 \ldots T_{l-1}$ and hence $S_1 \ldots S_{k-1} \cong_V T_1 \ldots T_{l-1}$ hold using $Scong_u T$, the inductive hypothesis and Lemmas 51 and 52. Thus either the path $\pi_T(i, e)$ fails to reach T_l (and thus is infinite) or it reaches T_l and so $\mathcal{M}[\![T_1 \ldots T_l]\!]^i_e(\omega) = \perp$ follows from Part (2) of Theorem 42.

$\square$

If a schema S contains a predicate p such that the values of $p^i(\mathbf{t})$ for an interpretation i can affect $\mathcal{M}[\![S]\!]^i_e(u)$, then changing i at a single predicate term must be able to change $\mathcal{M}[\![S]\!]^i_e(u)$. This motivates Definition 56.

Definition 56 (Fpu-couples and $p(\mathbf{t})u$-couples for a schema) Let i, j be interpretations and let $p \in \mathcal{P}$. We say that the set $\{i, j\}$ is a p-couple if there is a term $\mathbf{t}$ such that i and j differ only at the predicate term $p(\mathbf{t})$. In this case we may also say that $\{i, j\}$ is a $p(\mathbf{t})$-couple. If $\mathbf{t}$ is an F-term, then $\{i, j\}$ is an Fp-couple. If S is a schema and $u \in \mathcal{V} \cup \{\omega\}$, and $\mathcal{M}[\![S]\!]^i_e(u) \neq \mathcal{M}[\![S]\!]^j_e(u)$ (note that if $u = \omega$, this means that exactly one side terminates) and if $u \in \mathcal{V}$ then both sides terminate, then we may say that $\{i, j\}$ is an Fpu-couple (or a $p(\mathbf{t})u$-couple) for S.

Note that a pu-couple is simply an Fpu-couple with F as the empty word.

Proposition 57 (equivalence preserves existence of pu-couples) *Let S and T be schemas, let $p \in \mathcal{P}$ and $u \in \mathcal{V} \cup \{\omega\}$, suppose $S \cong_u T$ and let I be a pu-couple for S. Suppose either that $S \cong_\omega T$ holds, or that T is free and I is (q, T)-finite for all $q \in Preds(T)$; then I is also a pu-couple for T.* $\Box$

We will frequently use Proposition 57 in the remainder of this paper by studying the characteristics of the Fpu-couples of u-equivalent LFL schemas. In order to do this, we will define functions which we call *grades* on interpretations and couples (Definitions 62 and 108).

Definition 58 (head and tails of a couple) Let S be an LFL schema, let $u \in \mathcal{V} \cup \{\omega\}$, and let $q \in Preds(S)$. Let $I = \{i, j\}$ be a qu-couple for S and write

$$\pi_S(k, e) = \mu <\!q = Z(k)\!> \rho(k)$$

for each $k \in I$ and $\{Z(i), Z(j)\} = \{\mathsf{T}, \mathsf{F}\}$; thus, μ is the maximal common prefix of $\pi_S(i, e)$ and $\pi_S(j, e)$. Then we define $tail_S(k, I) = \rho(k)$ for each $k \in I$ and $head_S(I) = \mu$.

If $p \in \mathcal{P}$, we also define $Tailterms_S(p, k, I)$ to be the set of predicate terms $p(\mathbf{t})$ such that

$$\mathbf{t} = \mathcal{M}[\![schema(head_S(I)\nu)]\!]_e(\mathbf{refvec}_S(p))$$

for some segment ν of S satisfying $\nu <\!p = X\!> \in pre(tail_S(k, I))$ for some $X \in \{\mathsf{T}, \mathsf{F}\}$, and $Headterms_S(p, I)$ to be the set of predicate terms $p(\mathbf{t})$ such that

$$\mathbf{t} = \mathcal{M}[\![schema(\nu)]\!]_e(\mathbf{refvec}_S(p))$$

for some prefix ν of S satisfying $\nu <\!p = X\!> \in pre(head_S(I))$ for some $X \in \{\mathsf{T}, \mathsf{F}\}$.

Given $Y \in \{\mathsf{T}, \mathsf{F}\}$, we define $\mathit{Tailterms}_S(p, Y, k, I)$ to be the set

$$\{p(\mathbf{t}) \in \mathit{Tailterms}_S(p, Y, k, I) |\, p^k(\mathbf{t}) = Y\}$$

and $\mathit{Headterms}_S(p, Y, I)$ to be the set $\{p(\mathbf{t}) \in \mathit{Headterms}_S(p, Y, I) |\, p^k(\mathbf{t}) = Y$ for $k \in I\}$ (clearly well-defined).

Proposition 59 demonstrates the use of requiring our schemas to be liberal.

Proposition 59 *Let S, T_1, T_2 be predicate-free schemas and assume that each schema ST_i is liberal. Let $v_1, v_2 \in \mathcal{V}$. If $\mathcal{M}[\![ST_1]\!]_e(v_1) = \mathcal{M}[\![ST_2]\!]_e(v_2)$, then $\mathcal{M}[\![T_1]\!]_e(v_1) = \mathcal{M}[\![T_2]\!]_e(v_1)$ holds.*

Proof. Assume $\mathcal{M}[\![ST_1]\!]_e(v_1) = \mathcal{M}[\![ST_2]\!]_e(v_2)$ holds. The conclusion is proved by induction on the number of assignments in T_1. We may assume that T_1 and (similarly) T_2 contain assignments to v_1 and v_2 respectively, since if the last assignment to v_1 in ST_1 occurs in S, then since ST_2 is liberal, this is also the last assignment to $v_1 = v_2$ in ST_2; hence $\mathcal{M}[\![T_1]\!]_e(v_1) = \mathcal{M}[\![T_2]\!]_e(v_2) = v_1 = v_2$.
Let $v_i := f_i(\mathbf{u}_i)$; be the last assignment to v_i in T_i for each i. Clearly $f_1 = f_2$. Let u_1 and u_2 be the first components of $\mathbf{u}_1$ and $\mathbf{u}_2$ respectively, and write $T_i = T_i'\, v_i := f_i(\mathbf{u}_i); T_i''$ for each i. By the inductive hypothesis applied to S and each T_i', the term $\mathcal{M}[\![T_i']\!]_e(u_i)$ is the same for each i; the Proposition then follows from the analogous result for the other components of each $\mathbf{u}_i$. $\square$

Proposition 59 does not hold for non-liberal schemas; for example, if S and T_1 are $v := f()$; (so ST_1 is not liberal) and $T_2 = \Lambda$.

Lemma 60, which will be much used in the rest of this paper, shows that the head of a *pu*-couple for a schema can be changed without changing the tails.

Lemma 60 (changing the head of a couple) *Let S be an LFL schema and let $p \in \mathit{Preds}(S)$ and $u \in \mathcal{V} \cup \{\omega\}$. Suppose there is a pu-couple I for S and a prefix μ <$p = \mathsf{T}$> in S, then there is a pu-couple I' for S such that $\mu = \mathit{head}_S(I')$ and $\{\mathit{tail}_S(k, I) |\, k \in I\} = \{\mathit{tail}_S(k, I') |\, k \in I'\}$. In particular, if there is a pu-couple I for S and S contains an Fp-segment for $F \in \mathcal{F}^*$, then there exists an Fpu-couple I' for S.*

Proof. Write $I = \{i, j\}$ and assume that it is i which maps the predicate term on which i and j differ, to T. Write $\alpha(i) = \mathit{head}_S(I)$ <$p = \mathsf{T}$> and $\alpha(j) = \mathit{head}_S(I)$ <$p = \mathsf{F}$>. First observe that if $\sigma(i)$ and $\sigma(j)$ are prefixes of $\mathit{tail}_S(i, I)$ and $\mathit{tail}_S(j, I)$ respectively, and $x \in \mathcal{V}$, and

$$\mathcal{M}[\![\mathit{schema}(\mu\sigma(i))]\!]_e(x) = \mathcal{M}[\![\mathit{schema}(\mu\sigma(j))]\!]_e(x), \tag{1}$$

then $\mathcal{M}[\![\mathit{schema}(\sigma(i))]\!]_e(x) = \mathcal{M}[\![\mathit{schema}(\sigma(j))]\!]_e(x)$ using Proposition 59, and so

$$\mathcal{M}[\![\mathit{schema}(\alpha(i)\sigma(i))]\!]_e(x) = \mathcal{M}[\![\mathit{schema}(\alpha(j)\sigma(j))]\!]_e(x). \tag{2}$$

We now define the couple $I' = \{i', j'\}$ as follows. Let $\mathbf{y} = \mathbf{refvec}_S(p)$ and $\mathbf{t} = \mathcal{M}[\![schema(\mu)]\!]_e(\mathbf{y})$. Define $p^{i'}(\mathbf{t}) = \mathsf{T}$, $p^{j'}(\mathbf{t}) = \mathsf{F}$. Given any predicate q, vector term $\mathbf{u}$ and $X \in \{\mathsf{T}, \mathsf{F}\}$ such that $q(\mathbf{u}) \neq p(\mathbf{t})$ and $q(\mathbf{u}) = X$ is a consequence of one of the paths $\mu <\!p = \mathsf{T}\!> tail_S(i, I)$ or $\mu <\!p = \mathsf{F}\!> tail_S(j, I)$, we define $q(\mathbf{u}) = q^{j'}(\mathbf{u}) = X$; in the remaining cases we define $q^{i'}(\mathbf{u}) = q^{j'}(\mathbf{u})$ arbitrarily. We need to show that this definition does not contain an inconsistency. Suppose that $q(\mathbf{u}) = \mathsf{T}$ and $q(\mathbf{u}) = \mathsf{F}$ are both consequences of either of the paths $\mu <\!p = \mathsf{T}\!> tail_S(i, I)$ or $\mu <\!p = \mathsf{F}\!> tail_S(j, I)$. Since S is free, $q(\mathbf{u}) = \mathsf{T}$ and $q(\mathbf{u}) = \mathsf{F}$ cannot both be consequences of the same path, so after interchanging i and j if necessary, $q(\mathbf{u}) = \mathsf{T}$ is a consequence of $\mu <\!p = \mathsf{T}\!>$ $tail_S(i, I)$ and $q(\mathbf{u}) = \mathsf{F}$ is a consequence of $\mu <\!p = \mathsf{F}\!> tail_S(j, I)$. If $q(\mathbf{u}) = p(\mathbf{t})$ then no inconsistency occurs; and $q(\mathbf{u}) = \mathsf{T}$ and $q(\mathbf{u}) = \mathsf{F}$ are not consequences of μ, since S is free; thus there are prefixes $\sigma(i) <\!q = \mathsf{T}\!>$ and $\sigma(j) <\!q = \mathsf{F}\!>$ of $tail_S(i, I)$ and $tail_S(j, I)$ respectively such that (1) and hence (2) hold for every $x \in Refset_S(q)$. But this is impossible since then either

$$head_S(I) <\!p = \mathsf{T}\!> \sigma(i) <\!q = \mathsf{T}\!> \notin pre(\pi_S(i, e))$$

or

$$head(I) <\!p = \mathsf{F}\!> \sigma(j) <\!q = \mathsf{F}\!> \notin pre(\pi_S(j, e))$$

holds. Hence the definitions of i' and j' are consistent. Thus to show that $I' = \{i', j'\}$ is a pu-couple for S, it remains only to show that $\mathcal{M}[\![S]\!]_e^{i'}(u) \neq \mathcal{M}[\![S]\!]_e^{j'}(u)$. But if this were false, then clearly $u \in \mathcal{V}$ and (1) and hence (2) would hold for $x = u$ and $\sigma(i) = tail_S(i, I)$ and $\sigma'(j) = tail_S(j, I)$, contradicting the hypothesis that $\{i, j\}$ is a pu-couple for S. $\Box$

Lemma 60 is false for schemas which are not LFL. To see this, consider the free, linear, schema S of Figure 6. Owing to the constant g-assignment, S is not liberal. There is clearly an hpv-couple for S, but there is no $fhpv$-couple for S, although S contains an fhp-segment; indeed the slice of S obtained by deleting the f-assignment is v-equivalent to S.

Definition 61 (grade of a segment) Let S be a linear schema, let $p \in \mathcal{P}$, $X \in \{\mathsf{T}, \mathsf{F}\}$ and let μ be a segment in S. We define $grade(\mu, p, X)$ to be the number of occurrences of $<\!p = X\!>$ in the segment μ.

Note that S is not one of the parameters of *grade*; if μ is also a segment in a linear schema T, then its grade is unchanged.

Definition 62 (grade of an interpretation) Let i be an interpretation and let $q \in \mathcal{P}$ and $X \in \{\mathsf{T}, \mathsf{F}\}$. We define $grade(i, q, X)$ to be the number of predicate terms $q(\mathbf{t})$ for which $q^i(\mathbf{t}) = X$. If I is a p-couple for some $p \in \mathcal{P}$, then $grade(I, q, X)$ is the minimal element of $\{grade(i, q, X) | \, i \in I\}$.

Clearly if a schema S is free then $grade(i, q, X) \geq grade(\pi_S(i, e), q, X)$ always holds.

$while\, q(u)\ do$

$\{$

$u := k(u);$

$w := h(w);$

$if\ p(w) \quad then$

$\{$

$v := g();$

$w := f(w);$

$\}$

$else \quad \Lambda$

$\}$

Fig. 6. Schema S

Definition 63 (reasonable subsets of $\mathcal{P} \times \{\mathsf{T}, \mathsf{F}\}$) A subset of $\mathcal{P} \times \{\mathsf{T}, \mathsf{F}\}$ is said to be *reasonable* if it does not contain a pair $\{(p, \mathsf{T}), (p, \mathsf{F})\}$.

Definition 64 (finite and minimal interpretations for a sum) Let $\Omega \subseteq \mathcal{P} \times \{\mathsf{T}, \mathsf{F}\}$ be finite and reasonable. An interpretation i is said to be Ω-finite if

$$\sum_{(p,X) \in \Omega} grade(i, p, X) < \infty.$$

The interpretation i is said to be Ω-minimal for a set $\mathcal{I} \ni i$ if $\sum_{(p,X)\in\Omega} grade(i, p, X)$ is minimal over all interpretations in $\mathcal{I}$. If $\Omega = \{(p, X)\}$ then we simply write (p, X)-finite and (p, X)-minimal.
We extend these definitions to couples of interpretations in the natural way using Definition 62.

We will show in Lemma 69 that for a reasonable set Ω, and a set $\mathcal{I}$ of qu-couples for S, an Ω-minimal qu-couple for S in the set $\mathcal{I}$ must have an Ω-minimal head. This motivates the definition of the function $\theta_{S,p,X,I,J}$ (Definition 66).

Definition 65 (normalising an interpretation or couple) Let S be a schema, let i be a interpretation and let $p \in Preds(S)$ and $Y \in \{\mathsf{T}, \mathsf{F}\}$. Then we say that i is (S, p, Y)-normalised if $grade(i, p, Y) = grade(\pi_S(i, e), p, Y)$ holds. For any $q \in Preds(S)$ a q-couple $\{i, j\}$ is said to be (S, p, Y)-normalised if for all predicate terms $p(\mathbf{t})$, we have $p(\mathbf{t})^i = p(\mathbf{t})^j = \neg Y$ if $p(\mathbf{t}) = Y$ is not a consequence of either of the paths $\pi_S(i, e)$ or $\pi_S(j, e)$.

Clearly for any interpretation or couple, there is an (S, p, Y)-normalised interpretation or couple defining the same path(s) through S.

Definition 66 (The $\theta_{S,p,X,I,J}$ bijection) Let S be an LFL schema, let $p, q \in \mathit{Preds}(S)$, let $u \in \mathcal{V} \cup \{\omega\}$ and let I, J be *qu*-couples for S which have the same pair of tails in S. Let $X \in \{\mathsf{T}, \mathsf{F}\}$. We define the mapping

$$\theta_{S,p,X,I,J} : \bigcup_{k \in I} \mathit{Tailterms}_S(p, X, k, I) \to \bigcup_{k \in J} \mathit{Tailterms}_S(p, X, k, J)$$

as follows. Let $p(\mathbf{t}) \in \mathit{Tailterms}_S(p, X, k, I)$ for some $k \in I$. Then there is a segment $\mu \lhd p = X \rhd \in \mathit{pre}(\mathit{tail}_S(k, I))$ such that

$$\mathbf{t} = \mathcal{M}[\![\mathit{schema}(\mathit{head}_S(I)\mu)]\!]_e(\mathbf{refvec}_S(p)).$$

Let $\mathbf{t}' = \mathcal{M}[\![\mathit{schema}(\mathit{head}_S(J)\mu)]\!]_e(\mathbf{refvec}_S(p))$ and define $\theta_{S,p,X,I,J}(p(\mathbf{t})) = p(\mathbf{t}')$. By Proposition 59 $p(\mathbf{t}')$ is independent of the particular choice of μ and k, so the function $\theta_{S,p,X,I,J}$ is well-defined.

Observe that $\theta_{S,p,X,I,J} = (\theta_{S,p,X,J,I})^{-1}$; thus $\theta_{S,p,X,I,J}$ is a bijection.

Lemma 67 (same-tail couples have same number of (p, X)-tailterms) *Let S be an LFL schema, let $p, q \in \mathit{Preds}(S)$ and let $I = \{i(1), i(2)\}$ and $J = \{j(1), j(2)\}$ be qu-couples for S for some $u \in \mathcal{V} \cup \{\omega\}$. Let $X \in \{\mathsf{T}, \mathsf{F}\}$ and assume that the couples have the same pairs of tails; that is, $\mathit{tail}_S(i(k), I) = \mathit{tail}_S(j(k), J))$ for each $k \in \{1, 2\}$. Let $Y \in \{\mathsf{T}, \mathsf{F}\}$. Then*

$$|\mathit{Tailterms}_S(p, X, i(k), I)| = |\mathit{Tailterms}_S(p, X, j(k), J)|$$

for each $k \in \{1, 2\}$ and

$$|\bigcap_{k \in \{1,2\}} \mathit{Tailterms}_S(p, X, i(k), I)| = |\bigcap_{k \in \{1,2\}} \mathit{Tailterms}_S(p, X, j(k), J)|.$$

Proof. This follows from the freeness of S and the fact that the bijection $\theta_{S,p,X,I,J}$ maps $\mathit{Tailterms}_S(p, X, i(k), I)$ onto $\mathit{Tailterms}_S(p, X, j(k), J)$ for each $k \in \{1, 2\}$. $\square$

Proposition 68 (computing the (p, X)-grade of a couple) *Let S be an LFL schema, let $p, q \in \mathit{Preds}(S)$ and let $I = \{i, j\}$ be a qu-couple for S for some $u \in \mathcal{V} \cup \{\omega\}$. Let $X \in \{\mathsf{T}, \mathsf{F}\}$ and assume that I is (S, p, X)-mormalised. Then*

$$\mathit{grade}(I, p, X) = |\bigcup_{k \in I} \mathit{Tailterms}_S(p, X, k, I)| + \mathit{grade}(\mathit{head}_S(I), p, X)$$

holds.

Proof. This follows immediately from the freeness of S, since

$$\mathit{Tailterms}_S(p, k, I) \cap \mathit{Headterms}_S(p, I) = \emptyset$$

for each $k \in I$. $\square$

Lemma 69 (Ω-minimal couple has Ω-minimal head) *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S be an LFL schema. Let $q \in \mathit{Preds}(S)$ and let $\mathcal{I}$ be a set of qu-couples for S. Let Ω be a reasonable finite subset of $\mathcal{P} \times \{\mathsf{T}, \mathsf{F}\}$ and let $I \in \mathcal{I}$ be Ω-finite and Ω-minimal for all couples in $\mathcal{I}$. Let J be a qu-couple for S obtained by changing the head of I in S. Assume that $J \in \mathcal{I}$. Then*

$$\sum_{(p,X)\in\Omega} \mathit{grade}(\mathit{head}_S(J), p, X) \geq \sum_{(p,X)\in\Omega} \mathit{grade}(\mathit{head}_S(I), p, X)$$

holds, and J is Ω-minimal for all couples in $\mathcal{I}$ if and only if equality holds and J is (p, X)-normalised for every $(p, X) \in P$.

Proof. This follows immediately from Lemma 67, Lemma 60 and Proposition 68. $\square$

Theorem 75 and Lemmas 72 and 74 are the main results of this Section, whose other results will not be used in other Sections.

Proposition 70 *Let S be a free linear schema and assume that $x \rightsquigarrow_S g\,(r)$ and that there exists an $xgGv$-segment in S for $g \in \mathcal{F}$, $v \in \mathcal{V}$, $x \in \mathcal{F} \cup \mathcal{V}$ and $G \in \mathcal{F}^*$. Then there is an interpretation i such that $x \rightsquigarrow_{\mathcal{M}[\![S]\!]^i_e(v)} g\,(r)$.*

Proof. Since S is free, there is an interpretation i such that $\pi_S(i,e)$ contains the $fgGv$-segment, and so the result follows from the definition of $\mathcal{M}[\![S]\!]^i_e(v)$. $\square$

Proposition 71 *If S is a linear schema and there is an interpretation i such that $x \rightsquigarrow_{\mathcal{M}[\![S]\!]^i_e(v)} g\,(r)$ for some $v \in \mathcal{V}$, $x \in \mathcal{V} \cup \mathcal{F}$ and $g \in \mathcal{F}$ then $x \rightsquigarrow_S g\,(r)$ and there exists an $xgGv$-segment in S for some $G \in \mathcal{F}^*$.*

Proof. This follows from the definition of $\mathcal{M}[\![S]\!]^i_e(v)$. $\square$

Lemma 72 *Let S be a free schema and assume that $p \in whilePreds(S)$. For each predicate $q \in ifPreds(S)$, let $X(q) \in \{\mathsf{T},\mathsf{F}\}$. Let $p' \in Preds(S)$ and assume that $p' = p$ or $p' \searrow_S p$. Then there exists a $p'\omega$-couple $\{i,j\}$ for S (where $\pi_S(j,e)$ terminates) which is $(q,X(q))$-finite for every if predicate q, and is (q,T)-finite for every while predicate $q \neq p$, and satisfies $q^i(\mathbf{t}) = \mathsf{F}$ for all while predicates q such that $(q \neq p) \wedge \neg(q \searrow_S p)$ and satisfies $q^i(\mathbf{t}) = \neg X(q)$ for all if predicates q such that $\neg(q \searrow_S p\,(X(q))$.*

Proof. Let μ be a path in the body of p in S which does not enter the body of any while predicate in $body_S(p)$ or the $X(q)$-part of any if predicate q in $body_S(p)$. Choose a prefix $\tau <p = \mathsf{T}> \in pre(\Pi(S))$ such that if τ passes through a letter $<q = X(q)>$ for any $q \in Preds(S)$, then $q \searrow_S p\ (X(q))$.
Let the interpretation i satisfy

$$\pi_S(i,e) = \tau\,(<p = \mathsf{T}>\ \mu)^\omega$$

and assume that i is $(S,q,X(q)$-normalised for every $q \in ifPreds(S)$ and (S,q,T)-normalised for every $q \in whilePreds(S)$.
The prefix $\tau <p = \mathsf{T}>$ has a prefix $\tau' <p' = X>$, where $X = \mathsf{T}$ if $p' = p$ and $p' \searrow_S p\,(X)$ otherwise. Let the vector term

$$\mathbf{u} = \mathcal{M}[\![schema(\tau')]\!]_e(\mathbf{refvec}_S(p'))$$

and let the interpretation $j = i(p'(\mathbf{u}) = \neg(X))$. We will show that j terminates for S. Since j is (q,T)-finite for every $q \in whilePreds(S) - \{p\}$, and S is free, the path $\pi_S(j,e)$ could only fail to terminate by having infinitely many occurrences of $<p = \mathsf{T}>$. This is impossible if the predicate p' lies immediately below S, since clearly $\tau' \lessdot p' = \neg(X) \gtrdot$ is a prefix of the path $\pi_S(j,e)$, so this path does not contain the letter $\lessdot p' = X \gtrdot$. Alternatively, let p'' be the while predicate immediately above p'. Since there is only one predicate term $p''(\mathbf{t})$ for which $p''(\mathbf{t}) = \mathsf{T}$ is a consequence of j, the path $\pi_S(j,e)$ cannot reenter the body of p'' after $\tau' \lessdot p' = \neg(X) \gtrdot$, and so cannot enter the body or X-part of p'; thus $\pi_S(j,e)$ must terminate. Hence $\{i,j\}$ is a $p'\omega$-couple for S which satisfies the stated conditions. $\square$

Lemma 73 *Let S be an LFL schema and assume that $p \in Preds(S)$. Let $q \in Preds(S)$ and assume that $q \searrow_S g\ (X)$ for some $X \in \{\mathsf{T},\mathsf{F}\}$ and $g \in \mathcal{F}$. Let $u \in \mathcal{V} \cup \omega$. Suppose that there exists a (q,X)-finite pu-couple $\{i(1),i(2)\}$ for S, and S contains a gGp-segment for some $G \in \mathcal{F}^*$. Then there exists a qu-couple J for S such that for all $r \in \mathcal{P}$ and $Z \in \{\mathsf{T},\mathsf{F}\}$ such that $\{i(1),i(2)\}$ is (r,Z)-finite, the new couple J is also (r,Z)-finite.*

Proof. By Lemma 60 we may assume that $\{i(1),i(2)\}$ is a $gGpu$-couple for S. Define the interpretations $j(k) = i(k)(q = \neg X)$ for each $k \in \{1,2\}$. Clearly the interpretations $j(k)$ satisfy every (r,Z)-finiteness condition that each $i(k)$ satisfies. The paths $\pi_S(j(k),e)$ do not pass through g; since the interpretations $j(1)$ and $j(2)$ differ only at a predicate term containing g, we get $\mathcal{M}[\![S]\!]_e^{j(1)}(u) = \mathcal{M}[\![S]\!]_e^{j(2)}(u)$. Hence for at least one $k \in \{1,2\}$ we get $\mathcal{M}[\![S]\!]_e^{i(k)}(u) \neq \mathcal{M}[\![S]\!]_e^{j(k)}(u)$. Since the interpretations $i(k)$ and $j(k)$ differ only at finitely many q-predicate terms, and agree on all predicate terms $r(\mathbf{t})$ for $r \neq q$, there exists a qu-couple J for S whose elements map every predicate term $r(\mathbf{t})$ to either $r^{i(k)}(\mathbf{t})$ or $r^{j(k)}(\mathbf{t})$ and hence satisfy the given finiteness conditions. $\square$

Lemma 74 connects the syntactically defined set $\mathcal{N}_S(u)$ with the semantic definition of a pu-couple for S.

Lemma 74 ($p \in \mathcal{N}_S(u)$ implies there is a pu-couple for S) *Let S be an LFL schema and let $p \in \mathcal{P}$.*

(1) *Let $v \in \mathcal{V}$. Suppose that $p \in \mathcal{N}_S(v)$. Then there exists a pv-couple for S.*
(2) *Suppose that there is a while predicate q and a q-dependence sequence w for p in S. For each $r \in ifPreds(S)$ choose $Z(r) \in \{\mathsf{T},\mathsf{F}\}$ and assume that if w contains a 2-letter subword rx with $r \in ifPreds(S)$, then $r \searrow_S x\ (Z(r))$ holds. Then there exists a $p\omega$-couple $\{i,j\}$ for S which is $(r,Z(r))$-finite for all $r \in ifPreds(S)$ and (q',T)-finite for all $q' \in whilePreds(S) - \{q\}$.*

Proof.

(1) Let pwv be a minimal v-dependence sequence for p. The result follows by induction on the length of w. If w contains no predicates, then $w = gF$ for $F \in \mathcal{F}^*$ and $g \in \mathcal{F}$ with $p \searrow_S g\ (Z)$. Thus there is a gFv-segment in S. Let i be

a (p, Z)-finite interpretation passing through this segment; thus $\mathcal{M}[\![S]\!]^i_e(v)$ is a gF-term. Let $j = i(p = \neg Z)$. Clearly $\mathcal{M}[\![S]\!]^j_e(v)$ is not a gF-term, and so $\mathcal{M}[\![S]\!]^i_e(v) \neq \mathcal{M}[\![S]\!]^j_e(v)$, and since i and j differ only on finitely many p-predicate terms, the result follows. Alternatively, assume that w contains a predicate q; by the minimality of pwv we may write $pwv = pgFqw'v$ with $F \in \mathcal{F}^*$ and $g \in \mathcal{F}$ with $p \searrow_S g\ (Z)$. Here the result follows from Lemma 73 and the inductive hypothesis applied to q.

(2) The result follows by induction on the length of w. If w contains only predicate symbols (and hence $(p = q) \vee (p \searrow_S q)$ holds), then the result follows from Lemma 72. For the general case, we may write $w = pPgGp'w'$ with $P \in \mathcal{P}^*$, $G \in \mathcal{F}^*$ and $g \in \mathcal{F}$. Since the relation $\searrow_S$ is transitive, $p \searrow_S g\ (Z)$ holds for some $Z \in \{\mathsf{T}, \mathsf{F}\}$. By the assumption given, $p \in \mathit{ifPreds}(S) \Rightarrow Z = Z(p)$ holds. By the inductive hypothesis applied to p' and q, there is a $p'\omega$-couple I for S satisfying the required conditions; in particular, $p \in \mathit{ifPreds}(S)$ implies I is (p, Z)-finite. The result now follows from Lemma 73.

□

Theorem 75 **($S \cong_u T$ implies conditions (1),(4),(5),(6) of $S\ \mathit{simil}_u\ T$)** *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be LFL schemas. Assume that $S \cong_u T$. Then $\mathcal{N}_S(u) = \mathcal{N}_T(u)$. If $x \in \mathcal{F} \cup \mathcal{V}$ and $y \in (\{u\} \cap \mathcal{V}) \cup \mathcal{N}_S(u)$, then $x \rightsquigarrow_S y\,(n) \iff x \rightsquigarrow_T y\,(n)$.*

Proof. We first prove that $\mathcal{N}_S(u) = \mathcal{N}_T(u)$. If $p \in \mathcal{N}_S(u) \cap \mathcal{P}$ then by Lemma 74 there is a pu-couple for S, and hence for T by Proposition 57. Thus $p \in \mathcal{N}_T(u)$ using Part (2) of Theorem 42, since otherwise a u-equivalent slice of T not containing p could be obtained. Hence $\mathcal{N}_S(u) \cap \mathcal{P} \subseteq \mathcal{N}_T(u) \cap \mathcal{P}$ and equality similarly holds. If $f \in \mathcal{N}_S(u) \cap \mathcal{F}$ then either $u \in \mathcal{V}$ and there is an fFu-segment in S for $F \in \mathcal{F}^*$ (in which case this also holds in T, so $f \in \mathcal{N}_T(u)$) or there is an $fFpu$-segment in S for $F \in \mathcal{F}^*$ for $p \in \mathcal{N}_S(U) \cap \mathcal{P}$, and so by Lemma 60 there is an fFp-couple for S and hence for T, so $f \in \mathcal{N}_T(u)$. Thus $\mathcal{N}_S(u) \cap \mathcal{F} \subseteq \mathcal{N}_T(u) \cap \mathcal{F}$ and equality similarly holds. Now assume that $x \rightsquigarrow_S y\,(n)$ for $x \in \mathcal{F} \cup \mathcal{V}$ and $y \in \mathcal{N}_S(u) \cup (\{u\} \cap \mathcal{V})$. If S contains an xFp-segment, where $F \in \mathcal{F}^*$, $p \in \mathcal{N}_S(u) \cap \mathcal{P}$ and y is the first letter of Fp, then by Lemma 74, there is a $p(\mathbf{t})u$-couple $\{i, j\}$ for S and hence T by Proposition 57, where $x \rightsquigarrow_{p(\mathbf{t})} y\,(n)$. Hence $\pi_T(i, e)$ passes through $p(\mathbf{t})$ and so $x \rightsquigarrow_T y\,(n)$ holds. If alternatively S contains an xFu-segment for $u \in \mathcal{V}$, where $F \in \mathcal{F}^*$ and y is the first letter of Fu, then $x \rightsquigarrow_T y\,(n)$ follows from Propositions 70 and 71 provided that $y \in \mathcal{F}$; the case with $y = u$ follows similarly.
Thus we have shown that $x \rightsquigarrow_S y\,(n) \Rightarrow x \rightsquigarrow_T y\,(n)$ and the converse similarly holds.
□

The following result will be useful.

Theorem 76 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S be an LFL schema. Let T be any free schema. Assume that $S \cong_u T$. Let $p(\mathbf{t})$ be a predicate term with $p \in \mathcal{N}_S(u)$ and assume that* *there is a path through S passing through $p(\mathbf{t})$. Then there is a $p(\mathbf{t})u$-couple for T; in particular, there is a path through T passing through $p(\mathbf{t})$.*

Proof. The result follows immediately from Proposition 57 and Lemma 74. $\square$

7 Passages of a linear schema

The main result of this section is Theorem 90, whose statement is fairly similar to that of Part (2) of Lemma 74, but which requires a quite different method of proof. No other results in this section will be quoted later. In this section we will show (Theorem 81) that a path through a linear schema can be thought of as passing through a succession of while-free linear schemas which we call *passages* (Definition 78). This allows us to study how a single change in the definition of an interpretation feeds through a schema to eventually affect termination.

Definition 77 (super-while predicates of a schema) Let S be a linear schema and let $p \in \mathit{Preds}(S)$. We say that p is a super-while predicate of S if either $p \searrow_S q$ for some $q \in \mathit{whilePreds}(S)$ or $p \in \mathit{whilePreds}(S)$. We define $\mathit{swhilePreds}(S)$ to be the set of super-while predicates of S.

Definition 78 (schema passages and the $\mathit{nextpred}_S$ partial function) Let S be a linear schema. We will define a set $\mathit{Passages}(S)$ of linear while-free schemas, and $\mathit{super-while}$ predicates $\mathit{firstpred}(S)$ and $\mathit{nextpred}_S(p,Z)$ for $p \in \mathit{swhilePreds}(S)$, and $Z \in \{\mathsf{T},\mathsf{F}\}$, as follows. Write $S = S_1 \dots S_n$ where each S_i is indecomposable, then define the *initial* passage $\mathit{init}(S) = S_1 \dots S_r$, where the integer r is maximal such that $S_1 \dots S_r$ is while-free. If $r < n$ then S_{r+1} is an if or while schema and we define $\mathit{firstpred}(S)$ to be its guard; if $r = n$ then $\mathit{firstpred}(S)$ is undefined.
Having defined $\mathit{init}(S)$ and $\mathit{firstpred}(S)$ for all linear schemas S, let $p \in \mathit{swhilePreds}(S)$, and let $Z \in \{\mathsf{T},\mathsf{F}\}$; we recursively define $\mathit{passage}_S(p,Z)$ and $\mathit{nextpred}_S(p,Z) \in \mathit{swhilePreds}(S)$ as follows.
Assume that $\mathit{passage}_T(p,Z)$ and $\mathit{nextpred}_T(p,Z)$ have been defined (or declared undefined) for all linear schemas T satisfying $|T| < |S|$ and let M be the main subschema of S lying immediately above p in S and write $M = S_1 \dots S_n$ where each S_i is indecomposable. Assume that $p \in \mathit{Preds}(S_r)$. The schema S_r is clearly either an if schema or the while schema whose body is guarded by $p \in \mathit{whilePreds}(S)$. We consider three cases separately.

- Assume that the schema S_r is a while schema, and thus its body is guarded by $p \in \mathit{whilePreds}(S)$. We define

$$\mathit{passage}_S(p,X) = \begin{cases} \mathit{init}(S_{r+1} \dots S_n) & X = \mathsf{F} \\ \mathit{init}(\mathit{body}_S(p)) & X = \mathsf{T} \end{cases}$$

and $\mathit{nextpred}_S(p,X) =$

$$\begin{cases} \mathit{firstpred}(S_{r+1}\dots S_n) & X=\mathsf{F},\ \mathit{firstpred}(S_{r+1}\dots S_n)\text{ defined.}\\ \text{guard of } M & X=\mathsf{F},\ \mathit{firstpred}(S_{r+1}\dots S_n)\text{ undefined, } M\neq S\\ \text{undefined} & X=\mathsf{F},\ \mathit{firstpred}(S_{r+1}\dots S_n)\text{ undefined, } M=S\\ \mathit{firstpred}(\mathit{body}_S(p)) & X=\mathsf{T},\ \mathit{firstpred}(\mathit{body}_S(p))\text{ defined}\\ p & X=\mathsf{T},\ \mathit{firstpred}(\mathit{body}_S(p))\text{ undefined.}\end{cases}$$

- If S_r is an if schema guarded by $p' \neq p$, and if $p' \searrow_S p\,(X)$, then we define

$$\mathit{passage}_S(p,Z) = \mathit{passage}_{\mathit{part}_S^X(p')S_{r+1}\dots S_n}(p,Z)$$

  and $\mathit{nextpred}_S(p,Z) = \mathit{nextpred}_{\mathit{part}_S^X(p')S_{r+1}\dots S_n}(p,Z)$.
- If S_r is an if schema guarded by p, then

$$\mathit{passage}_S(p,Z) = \mathit{init}(\mathit{part}_S^Z(p)S_{r+1}\dots S_n)$$

  and $\mathit{nextpred}_S(p,Z) = \mathit{firstpred}(\mathit{part}_S^Z(p)S_{r+1}\dots S_n)$.

We call the elements of $\mathit{Passages}(S)$ and the schema $\mathit{init}(S)$ the *passages* of S.

A passage of S need not be a subschema of S; for example, if $S = (\,\mathit{if}\ p(x)\ \mathit{then}\ T'\ \mathit{else}\ T'')T'''$, where T'' and T''', but not T', are while-free, then $\mathit{passage}_S(p,\mathsf{F}) = T''T'''$.

Lemma 79 *Let S be a linear schema. Then $\mathit{passage}_S(p,Y) = \mathit{passage}_S(q,Z)$ implies $\mathit{nextpred}_S(p,Y) = \mathit{nextpred}_S(q,Z)$. Also, if $x \in \mathit{Funcs}(S) \cup \mathit{Preds}(S)$ is not a super-while predicate of S, then x lies in exactly one passage of S.*

Proof. This follows by induction on $|S|$. $\square$

Definition 80 (extending the definition of $\mathit{nextpred}_S$) Let S be a linear schema and let P be a passage of S. Then we define $\mathit{nextpred}_S(P) = \mathit{firstpred}(S)$ if $P = \mathit{init}(S)$ and $\mathit{nextpred}_S(P) = \mathit{nextpred}_S(q,Z)$ if $P = \mathit{passage}_S(q,Z)$. If x is a symbol in P, we also define $\mathit{nextpred}_S(x) = \mathit{nextpred}_S(P)$. By Lemma 79, $\mathit{nextpred}_S$ is well-defined as a partial function.

The motivation for this definition is given by the following result.

Theorem 81 *Let S be a linear schema.*

(1) *Let $\mu \in \mathit{pre}(\Pi(S))$. Write*

$$\mu = \mu(0) \lessdot q_1 = X_1 \gtrdot \mu(1) \lessdot q_2 = X_2 \gtrdot \dots \mu(n)$$

where the letters $\lessdot q_r = X_r \gtrdot$ are all the occurrences of elements of $\mathit{swhilePreds}(S)$ in μ. Then each segment $\mu(r) \in \Pi(\mathit{passage}_S(q_r, X_r))$ for $r \leq n-1$ and $\mu(n) \in \mathit{pre}(\Pi(\mathit{passage}_S(q_n, X_n)))$, and $\mu(0) \in \Pi(\mathit{init}(S))$ if $n \geq 1$. Also each $q_{r+1} = \mathit{nextpred}_S(q_r, X_r)$, and $q_1 = \mathit{firstpred}(S)$.

(2) *Let* $\mu(0) \in \Pi(init(S))$ *and define* $q_1 = firstpred(S)$, $q_r \in swhilePreds(S)$ *for* $2 \leq r \leq n$ *such that each* $q_{r+1} = nextpred_S(q_r, X_r)$. *Let* $X_r \in \{\mathsf{T}, \mathsf{F}\}$ *for each* $r \in \{1, \ldots, n\}$. *Let* $\mu(r) \in \Pi(passage_S(q_r, X_r))$ *for* $1 \leq r \leq n$. *Then*

$$\mu(0) \lessdot q_1 = X_1 \gtrdot \mu(1) \lessdot q_2 = X_2 \gtrdot \ldots \mu(n) \in pre(\Pi(S)).$$

Proof. Both results follow by induction on $|S|$, using Definition 9 and by expressing S in one of the forms AB, *if* $p(\mathbf{y})$ *then* A *else* B, *while* $p(\mathbf{y})$ *do* A or as an assignment, and using the inductive hypothesis applied to A and B. $\square$

Observe that if S is a linear schema and P is a passage of S with $q \in ifPreds(P)$ then $q \searrow_S x\,(Z) \iff q \searrow_P x\,(Z)$ holds for all $x \in Symbols(S)$ and $Z \in \{\mathsf{T}, \mathsf{F}\}$. Also $f \rightsquigarrow_P x \iff (f \rightsquigarrow_S x \;\wedge (\neg back_S(q, f, x)) \forall q \in whilePreds(S))$ holds if P contains both $f \in \mathcal{F}$ and $x \in \mathcal{F} \cup \mathcal{P}$.

The definition of liberality that we have used up to now, using the natural state, is too narrow for our purposes here; we need to consider paths defined using initial states apart from e.

Definition 82 (liberality with respect to a state) Let S be a schema and let $c : \mathcal{V} \to Terms(\mathcal{F}, \mathcal{V})$ be a state. Then we say that S is c-liberal if given any interpretation i such that $\pi_S(i, c)$ passes through a function symbol f, the path $\pi_S(i, c)$ defines a different f-term (with respect to the initial state c) each time it passes through f, and all such terms are different from every element of $\{c(v) | \, v \in \mathcal{V}\}$.

Proposition 83 *Let* S *be an LFL schema and let* $\mu \lessdot q = Z \gtrdot \in pre(\Pi(S))$. *Let the state* $c = \mathcal{M}[\![schema(\mu)]\!]_e$. *Then the schema* $passage_S(q, Z)$ *is c-liberal.*

Proof. This follows immediately from the fact that S is LFL. $\square$

Lemma 84 *Let* S *be a linear while-free schema and let* $v \in \mathcal{V}$. *For each* $q \in ifPreds(S)$, *let* $Y(q) \in \{\mathsf{T}, \mathsf{F}\}$.

(1) *Let* c_1, c_2 *be any states which differ on a variable in* $Inv_S(v)$, *and assume that* S *is* c_k*-liberal for each* $k \in \{1, 2\}$. *Then there is an interpretation* i *such that* $\mathcal{M}[\![S]\!]^i_{c_1}(v) \neq \mathcal{M}[\![S]\!]^i_{c_2}(v)$ *and for each* $q \in ifPreds(S)$, *if* $\pi_S(i, c_1)$ *passes through* $\lessdot q = Y(q) \gtrdot$, *then* $\pi_S(i, c_2)$ *does not pass through* $\lessdot q = \neg Y(q) \gtrdot$.

(2) *Let* c *be any state such that* S *is* c*-liberal and let* $p \in \mathcal{N}_S(v)$. *Then there is a* p*-couple* $\{i, j\}$ *such that* $\mathcal{M}[\![S]\!]^i_c(v) \neq \mathcal{M}[\![S]\!]^j_c(v)$ *and for each* $q \in ifPreds(S)$, *if the path* $\pi_S(i, c)$ *passes through* $\lessdot q = Y(q) \gtrdot$, *then* $\pi_S(j, c)$ *does not pass through* $\lessdot q = \neg Y(q) \gtrdot$.

Proof.

(1) This follows by induction on $|S|$. If S is an assignment, then the result is immediate. If $S = AB$ nontrivially, then the result follows by the inductive hypothesis applied to A and B. Lastly, assume that $S =$ *if* $p(\mathbf{x})$ *then* A *else* B. If c_1 and c_2 differ on a variable in $Inv_A(v)$ then by the inductive hypothesis applied to A, there is an interpretation j such that $\mathcal{M}[\![A]\!]^j_{c_1}(v) \neq \mathcal{M}[\![A]\!]^j_{c_2}(v)$ and j satisfies the additional hypothesis on if predicates, so $i = j(p = \mathsf{T})$ gives the result. The case where c_1 and c_2 differ on a variable in $Inv_B(v)$ can be treated similarly. On the other hand, if c_1 and c_2 agree on every element of $Inv_A(v) \cup Inv_B(v)$, then $p \in \mathcal{N}_S(v)$ and $c_1(\mathbf{x}) \neq c_2(\mathbf{x})$ follows, and either A or B contains an assignment to v. We may assume the former; $f \rightsquigarrow_A v$, say. We will assume $Y(p) = \mathsf{F}$; otherwise interchange c_1 and c_2. Since S is linear and $c_1(\mathbf{x}) \neq c_2(\mathbf{x})$, there is an interpretation i such that $\pi_S(i, c_1)$ ends in an fv-segment, and $\pi_S(i, c_2)$ enters B, and the two paths satisfy the conditions on $ifPreds(S)$. If $\pi_S(i, c_2)$ passes through an assignment to v then i clearly satisfies the required conditions. If not, then $v \rightsquigarrow_B v$ and so $v \in Inv_B(v)$ holds, thus $\mathcal{M}[\![S]\!]^i_{c_2}(v) = c_2(v) = c_1(v) \neq \mathcal{M}[\![S]\!]^i_{c_1}(v)$, where the inequality follows since S is c_1-liberal.

(2) This also follows by induction on $|S|$, using a similar classification of schemas to that used in proving the first part of this Lemma.

□

Definition 85 (passage sequences) Let S be a linear schema and let $x \in Symbols(S) - swhilePreds(S)$ and $y \in Symbols(S)$.
A y-passage sequence for x in S is a word

$$w = xq_1Z_1q_2Z_2 \ldots q_mZ_my$$

for $Z_i \in \{\mathsf{T}, \mathsf{F}\}$ and $q_i \in swhilePreds(S)$ such that there is a passage P_0 in S containing x, and if we define the passages $P_i = passage_S(q_i, Z_i)$ for each $i < m$, then one of the following holds;

- $m = 0$ (that is, $w = xy$) and either $y \in Symbols(P_0) \wedge x \in \mathcal{N}_{P_0}(y)$, or $y = nextpred_S(x) \wedge x \in \mathcal{N}_P(v)$ for some $v \in Refset_S(y)$, or
- $m > 0$ and $q_1 = nextpred_S(x)$, each $q_{i+1} = nextpred_S(P_i)$, and there are variables $v_1, \ldots, v_m$ such that $x \in \mathcal{N}_{P_0}(v_1)$ and each $v_i \in Inv_{P_i}(v_{i+1})$ for $i < m$, and either $y \in Symbols(P_m) \wedge v_m \in Inv_{P_m}(y)$ or $y = nextpred_S(P_m) \wedge v_m \in Inv_{P_m}(v)$ for some $v \in Refset_S(y)$.

Lemma 86 (joining of passage sequences at a symbol) *Let S be a linear schema and let $xq_1Z_1q_2Z_2 \ldots q_mZ_my$ and $yq'_1Z'_1q'_2Z'_2 \ldots q'_nZ'_nz$ be passage sequences in S for $x, y \in Symbols(S) - swhilePreds(S)$ and $z \in Symbols(S)$. Then*

$$xq_1Z_1q_2Z_2 \ldots q_mZ_mq'_1Z'_1q'_2Z'_2 \ldots q'_nZ'_nz$$

is a z-passage sequence for x in S.

Proof. This follows immediately from the preceding definition. ☐

Lemma 87 (dependence sequences imply passage sequences) *Let S be a linear schema and let $x \in \mathit{Symbols}(S) - \mathit{swhilePreds}(S)$. Let $y \in \mathit{Symbols}(S)$. Suppose there is a y-dependence sequence xwy for x in S such that the word w does not contain any super-while predicates, then there is a y-passage sequence for x in S.*

Proof. The result follows by induction on the length of w. We may assume that $x \neq y$ and that xwy is minimal and so contains no repeated symbols. Let P be a passage containing x. Let xw' be the longest prefix of xwy such that if fz is a two-letter segment of xw' with $f \in \mathcal{F}$, then there is an fz-segment in S which is also a segment of P. If $w' = wy$ then xy is a y-passage sequence for x in S. If not, then xw' ends in a function symbol, f say. Clearly $f \neq y$. Write $xwy = xW_1fzW_2$, where z is the first letter of xwy immediately after f. By the inductive hypothesis applied to xW_1f and zW_2, there is an f-passage sequence $xp_1Y_1 \ldots p_mY_mf$ for x in S and a y-passage sequence $zq_1Z_1 \ldots q_nZ_ny$ for z in S.
Let

$$\mu(1) < r(1) = X(1) > \mu(2) \ldots < r(k) = X(k) > \mu(k+1)$$

be an fz-segment in S, written such that the letters $< r(i) = X(i) >$ are all the occurrences of super-while predicates in the segment. By the choice of w', we have $k \geq 1$. Let f assign to v in S; thus by Part (1) of Theorem 81, we get $f \rightsquigarrow_{passage_S(p_m,y_m)} v$,

$$v \rightsquigarrow_{passage_S(r(j),X(j))} v$$

for each $j < k$, and

$$v \rightsquigarrow_{passage_S(r(k),X(k))} z.$$

Thus $fr(1)X(1) \ldots r(k)X(k)z$ is a z-passage sequence for f in S. Hence

$$xp_1Y_1 \ldots p_mY_mr(1)X(1) \ldots r(k)X(k)zq_1Z_1 \ldots q_nZ_ny$$

is a y-passage sequence for x in S, by Lemma 86. ☐

Lemma 88 *Let S be an LFL schema and for every $q \in \mathit{ifPreds}(S)$ let $Y(q) \in \{\mathsf{T}, \mathsf{F}\}$. Let $\sigma_1, \sigma_2 \in \mathit{pre}(\Pi(S))$ and assume that there exists $m \geq 0$ and $Z_0, \ldots, Z_{m-1} \in \{\mathsf{T}, \mathsf{F}\}$ and that for each $k \in \{1, 2\}$ there exists $X_k \in \{\mathsf{T}, \mathsf{F}\}$ such that*

$$\sigma_k = \mu < q_0 = Z_0 > \mu(0,k) < q_1 = Z_1 > \mu(1,k) \ldots \quad \ldots \mu(m-1,k) < q_m = X_k >$$

such that each $q_r \in \mathit{swhilePreds}(S)$ and $\mu(r,k) \in \Pi(\mathit{passage}_S(q_r, Z_r))$ and $q_{r+1} = \mathit{nextpred}_S(q_r, Z_r)$. Suppose that $\sigma_2\tau \in \mathit{pre}(\Pi(S))$ for some segment τ in S and the following conditions hold for every $q \in \mathit{ifPreds}(S)$;

(1) *the segment τ does not pass through $< q = Y(q) >$;*

(2) *for each* $r < m$*, if* $\mu(r,2)$ *passes through* $<q = Y(q)>$*, then* $\mu(r,1)$ *does not pass through* $<q = \neg Y(q)>$*;*
(3) *for any* $s < m$ *and* $r < s$*, if* $\mu(r,1)$ *passes through* $<q = Y(q)>$ *and* $\mu(r,2)$ *passes through* $<q = \neg Y(q)>$*, then* $\mu(s,2)$ *does not pass through* $<q = Y(q)>$*.*

Let $x \in \mathcal{F} \cup \mathcal{P}$*. Suppose that the prefixes* σ_1 *and* $\sigma_2\tau$ *both pass through the* x*-term* $x(\mathbf{t})$ *and* $x(\mathbf{t})$ *occurs in* $\mu(r,1)$ *along* σ_1*. Then* $x(\mathbf{t})$ *occurs in* $\mu(s,2)$ *along* $\sigma_2\tau$*, for some* $s \leq r$*.*

Proof. This follows by induction on the sum of the lengths of the prefixes of σ_1 and $\sigma_2\tau$ ending in the mentioned occurrences of x. Assume that the conclusion of the Lemma is false; thus $x(\mathbf{t})$ occurs either along τ or along $\mu(s,2)$ for some $s > r$. We will deduce a contradiction. We consider two cases.

- Suppose first that σ_2 also passes through x within the segment $\mu(r,2)$. Since S is free and liberal, σ_2 also passes through some $y \in \mathcal{F}$ between the two mentioned occurrences of x, such that the prefix of $\sigma_2\tau$ up to the second occurrence of x ends in a yx-segment. Thus the prefix of σ_1 ending in $x(\mathbf{t})$ also ends in a yx-segment, and so the relevant occurrence of y is in $\mu(r',1)$ for some $r' \leq r$. We will show that the result follows from the inductive hypothesis applied to the two occurrences of y, which clearly define the same term along the two prefixes. If $r' < r$, or the relevant occurrence of y in $\sigma_2\tau$ is along τ or in $\mu(s',2)$ for $s' > r$, then this is immediate. Thus we may assume that $r' = r$ and that y occurs in $\mu(s',2)$ for $s' \leq r$. Clearly $s' = r$ since y occurs between the two occurrences of x. Since $r' = r$, $y \rightsquigarrow_{passage_S(q_r,Z_r)} x$ holds, so $\mu(r,2)$ cannot pass through y after passing through x, giving a contradiction and so proving the result.
- Alternatively, suppose that σ_2 does not pass through x within the segment $\mu(r,2)$. Then there is an if predicate q such that $q \searrow_{passage_S(q_r,Z_r)} x\,(Z)$ and $\mu(r,2)$ passes through $<q = \neg Z>$ and $\mu(r,1)$ passes through $<q = Z>$. Thus $Z = Y(q)$ by Condition (2) of the hypotheses. Hence by Condition (1) of the hypotheses, $x(\mathbf{t})$ occurs along $\mu(s,2)$ for some $s > r$; but this contradicts Condition (3) of the hypotheses, thus proving the Lemma.

□

Lemma 89 *Let* S *be a linear while-free schema, let* c_1, c_2 *be any states, let* κ_1 *and* κ_2 *be any interpretations and assume that the path* $\pi_S(\kappa_1, c_1)$ *passes through a predicate term* $p(\mathbf{t})$ *starting at* c_1*, whereas* $\pi_S(\kappa_2, c_2)$ *does not pass through* $p(\mathbf{t})$ *starting at* c_2*. Then either* c_1, c_2 *differ on a variable in* $Inv_S(p)$ *or* κ_1 *and* κ_2 *differ on a predicate in* $\mathcal{N}_S(p)$*.*

Proof. This is similar to Part (2) of Theorem 42 and uses induction on $|S|$. □

Theorem 90 *Let* S *be an LFL schema and let* $p \in ifPreds(S) \cap \mathcal{N}_S(\omega)$*. For each*

$q \in ifPreds(S)$, let $Y(q) \in \{\mathsf{T}, \mathsf{F}\}$. Then there is a $p\omega$ couple $\{i(1), i(2)\}$ for S which is $(q, Y(q))$-finite for every $q \in ifPreds(S)$ and (q, T)-finite for every $q \in whilePreds(S)$ except one, for which $body_S(q)$ is while-free.

Proof. If $p \in swhilePreds(S)$ then the result follows from Lemma 72, so we may assume $p \notin swhilePreds(S)$. Thus since $p \in \mathcal{N}_S(\omega)$, there is an ω-dependence sequence pw for p in S, with $|w| \geq 1$. Let w' be the shortest prefix of w containing a symbol in $swhilePreds(S)$. Applying Lemma 87 to pw' shows that there are predicates $q_1, \ldots, q_m \in swhilePreds(S)$ such that $pq_1Z_1q_2 \ldots Z_{m-1}q_m$ is a q_m-passage sequence for p in S. We may assume that m is minimal for all $q' \in swhilePreds(S)$ such that a q'-passage sequence for p in S exists. By the definition of a passage sequence there are variables $v_1, \ldots, v_m$ such that $v_m \in Refset_S(q_m)$ and $v_r \in Inv_{passage_S(q_r, Z_r)}(v_{r+1})$ for $1 \leq r < m$ and $p \in \mathcal{N}_{passage_S(q_0, Z_0)}(v_1)$ for some passage $passage_S(q_0, Z_0)$ containing p. We will define the p-couple $\{i(1), i(2)\}$ such that for each $k \in \{1, 2\}$ we have

$$\pi_S(i(k), e) = \sigma_k,$$

where

$$\sigma_k = \mu <q_0 = Z_0> \mu(0, k) <q_1 = Z_1> \ldots \mu(m-1, k) <q_m = X_k> \rho(k)$$

for $\mu(r, k) \in passage_S(q_r, Z_r)$ and such that each $q_{r+1} = nextpred_S(q_r, Z_r)$, and $\mathcal{M}[\![schema(\tau(r, 1))]\!]_e(v_r) \neq \mathcal{M}[\![schema(\tau(r, 2))]\!]_e(v_r)$ (where $\tau(r, k)$ is the prefix of $\pi_S(i(k), e)$ preceding the given occurrence of q_r), $X_1 \neq X_2$, and the occurrence of p at which the paths deviate is along $\mu(0, k)$ in $passage_S(q_0, Z_0)$. The path σ_2 will be the non-terminating one. We will be using Lemma 88 for $\tau = \rho(2)$ and will thus need to ensure that the segments $\mu(r, k)$ satisfy the hypotheses of that Lemma. The construction of the paths σ_k occurs in several stages.

- We first choose μ to be any prefix in S such that $\mu <q_0 = \mathsf{T}> \in pre(\Pi(S))$. We also define the state $c_0^1 = c_0^2 = \mathcal{M}[\![schema(\mu)]\!]_e$.
- We then define each segment $\mu(0, k)$ as follows. By Proposition 83 and Part (2) of Lemma 84 applied to the while-free linear schema $passage_S(q_0, Z_0)$, there is a p-couple $\{\kappa_0^1, \kappa_0^2\}$ such that

  $$\mathcal{M}[\![passage_S(q_0, Z_0)]\!]_{c_0^1}^{\kappa_0^1}(v_1) \neq \mathcal{M}[\![passage_S(q_0, Z_0)]\!]_{c_0^2}^{\kappa_0^2}(v_1),$$

  and such that for every $q \in ifPreds(passage_S(q_0, Z_0))$, if the path $\pi_{passage_S(q_0, Z_0)}(\kappa_0^2, c_0^2)$ passes through $<q = Y(q)>$, then $\pi_{passage_S(q_0, Z_0)}(\kappa_0^1, c_0^1)$ does not pass through $<q = \neg Y(q)>$. We define

  $$\mu(0, k) = \pi_{passage_S(q_0, Z_0)}(\kappa_0^k, c_0^k).$$

  We also define the states $c_1^k = \mathcal{M}[\![passage_S(q_0, Z_0)]\!]_{c_0^k}^{\kappa_0^k}$.
- We now recursively define interpretations κ_s for $1 \leq s < m$, the segments $\mu(s, k) = \pi_{passage_S(q_s, Z_s)}(\kappa_s, c_s^k)$ for $s < m$ and $k \in \{1, 2\}$ and define $c_{s+1}^k = \mathcal{M}[\![passage_S(q_s, Z_s)]\!]_{c_s^k}^{\kappa_s}$,

such that

$$c^1_{s+1}(v_{s+1}) \neq c^2_{s+1}(v_{s+1})$$

holds, and that the resulting segments $\mu(s,k)$ satisfy Condition (2) of Lemma 88 for $r = s$ and all $q \in \mathit{ifPreds}(\mathit{passage}_S(q_s, Z_s))$. Assume that this has been done for all $r < s$. By Proposition 83 and Part (1) of Lemma 84 applied to the while-free linear schema $\mathit{passage}_S(q_s, Z_s)$ and the variable v_{s+1}, an interpretation κ_s, states c^k_{s+1} and paths $\mu(s,k)$ exist satisfying the conditions above.

- We now modify each interpretation κ_s in several steps, in order that Condition (3) of Lemma 88 is also satisfied, and to ensure that κ_s is consistent with the previously defined interpretations. Again, we do this in order of increasing s.

  Define $\mathit{ifPreds}(\mathit{passage}_S(q_s, Z_s)) = \{p_1, \ldots, p_a\}$, with the predicates ordered so that $p_l \ll_{\mathit{passage}_S(q_s,Z_s)} p_{l'} \Rightarrow l < l'$. For each p_l, with l in increasing order, we modify κ_s with the following steps. Assume that for each l, each path $\mu(s,k)$ passes through $p_l(\mathbf{t}_{l,k})$ if it passes through p_l.

(1) Suppose first that $p_l(\mathbf{t}_{l,1}) = Z$ is a consequence of $\mu(s,1)$, whereas $p_l(\mathbf{t}_{l,1}) = \neg Z$ is a consequence of $\mu(r,2)$ for some $r < s$. Note that by the freeness of S this implies $\mathbf{t}_{l,1} \neq \mathbf{t}_{l,2}$. We then replace κ_s by $\kappa_s(p_l(\mathbf{t}_{l,1}) = \neg Z)$. (We will consider later the analogous situation with 1 and 2 interchanged.)

(2) Suppose next that $\mu(s,2)$ passes through $\lessdot p_l = Y(p_l)\gtrdot$, whereas for some $r < s$, $\mu(r,1)$ and $\mu(r,2)$ pass through $\lessdot p_l = Y(p_l)\gtrdot$ and $\lessdot p_l = \neg Y(p_l)\gtrdot$ respectively. In order that Condition (3) of Lemma 88 be satisfied, we replace κ_s by $\kappa_s(p_l(\mathbf{t}_{l,2}) = \neg Y(p_l))$. Note that this modification does not change the path $\mu(s,1)$ unless $\mathbf{t}_{l,1} = \mathbf{t}_{l,2}$, so Step (1) does not become necessary as a result of this modification.

(3) As a consequence of the execution of Steps (1) or (2), it may be that for some $l' \geq l$, Condition (2) of Lemma 88 for $r = s$ and $q = p_{l'}$ is no longer satisfied. If so, we replace κ_s by $\kappa_s(p_{l'}(\mathbf{t}_{l',2}) = \neg Y(p_{l'}))$. Again, if $l' = l$ then Step (1) need not be executed again.

We now need to prove that these modifications to κ_s do not change the terms $c^k_{s+1}(v_{s+1})$, as defined above, so that each κ_t for $t > s$ still satisfies the conditions that it was originally chosen to satisfy. Suppose this is not the case. If Step (1) or (2) is executed, then for the given value of $r < s$, $\mathit{passage}_S(q_r, Z_r)$ and $\mathit{passage}_S(q_s, Z_s)$ both contain p_l, hence $\mathit{passage}_S(q_r, Z_r) = \mathit{passage}_S(q_s, Z_s)$. By Lemma 89 and (in the case of Step (1)) the freeness of S, $r = 0 \Rightarrow p \in \mathcal{N}_{\mathit{passage}_S(q_r,Z_r)}(p_l)$ holds. If $r \geq 1$ then by induction on $r - b$ and Part (2) of Theorem 42, there are variables $\{w_1, \ldots, w_r\}$ such that the states c^k_b differ on w_b and each $w_b \in \mathit{Inv}_{\mathit{passage}_S(q_b,Z_b)}(w_{b+1})$ for $b < r$ and $p \in \mathcal{N}_{\mathit{passage}_S(q_0,Z_0)}(w_1)$ and $w_r \in \mathit{Inv}_{\mathit{passage}_S(q_r,Z_r)}(p_1)$. Thus in either case $pq_1 \ldots q_r Z_r p_l$ is a p_l-passage sequence for p in S.

If an execution of Step (1) or (2) changes a term $c^k_{s+1}(v_{s+1})$, then $p_l \in \mathcal{N}_{\mathit{passage}_S(q_s,Z_s)}(v_{s+1})$ by Part (2) of Theorem 42; if an execution of Step (3) changes a term $c^k_{s+1}(v_{s+1})$, then $p_{l'} \in \mathcal{N}_{\mathit{passage}_S(q_s,Z_s)}(v_{s+1})$ by Part (2) of Theorem 42, and since Step (3) was necessary, $p_l \in \mathcal{N}_{\mathit{passage}_S(q_s,Z_s)}(p_{l'})$ and so $p_l \in \mathcal{N}_{\mathit{passage}_S(q_s,Z_s)}(v_{s+1})$ holds. Hence $r \geq 1 \Rightarrow v_r \in \mathit{Inv}_{\mathit{passage}_S(q_r,Z_r)}(v_{s+1})$ and $r = 0 \Rightarrow p \in \mathcal{N}_{\mathit{passage}_S(q_r,Z_r)}(v_{s+1})$ and so by Lemma 86,

$$pq_1 \ldots q_r Z_r q_{s+1} \ldots Z_{m-1} q_m$$

is a q_m-passage sequence for p in S, contradicting the minimality of m.

- We now define the segments $<q_m = X_k>\ \rho(k)$ for each $k \in \{1, 2\}$. For $k = 2$ this segment will be infinite. If $q_m \in \mathit{whilePreds}(S)$ then we define $X_2 = \mathsf{T}$ and $X_1 = \mathsf{F}$; and if $q_m \in \mathit{ifPreds}(S)$ then we require that the X_2-part of q_m contain a while predicate and $X_2 \neq X_1$. The segment $\rho(2)$ is defined as follows. If it meets an if predicate $q \in \mathit{swhilePreds}(S)$, then it enters a part of q containing a while predicate. If it meets a while predicate q, it passes through $<q = \mathsf{T}>$. And if it meets an if predicate $q \notin \mathit{swhilePreds}(S)$, it passes through $<q = \neg Y(q)>$. Notice that $\rho(2)$ only passes finitely often through any super-while if predicate. The segment $\rho(1)$ is defined as follows. If it meets a while predicate q, it passes through $<q = \mathsf{F}>$; subject to that condition, it may take any path consistent with the prefix of σ_2 preceding $\rho(2)$. This clearly ensures that it terminates.

The path σ_2 is clearly $(q, Y(q))$-finite for every $q \in \mathit{ifPreds}(S)$, and there is only one while predicate through which $\rho(2)$ passes infinitely often. Thus to prove the existence of the interpretations $i(k)$ satisfying $\pi_S(i(k), e) = \sigma_k$, it only remains to show that there is no predicate term $q(\mathbf{t})$ such that $q(\mathbf{t}) = X$ is a consequence of σ_1 and $q(\mathbf{t}) = \neg X$ is a consequence of σ_2. We will assume otherwise and deduce a contradiction.

Note that the segments $\rho(k)$ do not pass through any common predicate, and so $q(\mathbf{t})$ occurs before $\rho(k)$ for at least one $k \in \{1, 2\}$. Also, by the freeness of S, the 'bad' occurrences of $q(\mathbf{t})$ cannot be in μ in either path. We consider three cases.

- Assume $q \notin \mathit{swhilePreds}(S)$. Assume $q(\mathbf{t}) = X$ is a consequence of a segment $\mu(r, 1)$ whereas $q(\mathbf{t}) = \neg X$ is a consequence of $\mu(s, 2)$ for some $r < s$ or of $\rho(2)$ (by Step (1) in the construction of κ_s, the same situation with 1 and 2 interchanged cannot happen). Lemma 88, with $\rho(2)$ used as the segment τ of that Lemma, shows that this is impossible, since the segments $\mu(s, k)$ were constructed so as to satisfy the hypotheses of Lemma 88.
- Assume $q = q_r$ for some $r < m$. Thus $\mathbf{t} = c_r^k(\mathbf{refvec}_S(q_r))$ for some $k \in \{1, 2\}$, and so by the freeness of S, we get

$$c_r^1(\mathbf{refvec}_S(q_r)) \neq c_r^2(\mathbf{refvec}_S(q_r))$$

  for one such r. By induction on $r - b$ and Part (2) of Theorem 42, there are variables $\{w_1, \ldots, w_r\}$ with $w_r \in \mathit{Refset}_S(q_r)$ and such that the states c_b^k differ on w_b and each $w_b \in \mathit{Inv}_{\mathit{passage}_S(q_b, Z_b)}(w_{b+1})$ for each $b < r$ and $p \in \mathcal{N}_{\mathit{passage}_S(q_0, Z_0)}(w_1)$. Thus $pq_1 \ldots q_{r-1} Z_{r-1} q_r$ is a q_r-passage sequence for p in S, contradicting the minimality of m.
- Assume $q = q_m$. If $q_m \in \mathit{ifPreds}(S)$, then neither segment $\rho(k)$ passes through q, so $q_m \in \mathit{whilePreds}(S)$ must hold. Since $\rho(1)$ does not pass through q_m in this case, $\rho(2)$ must pass through $q_m(\mathbf{t})$. By the freeness of S, $\rho(2)$ must pass through some $f \in \mathcal{F}$ defining a term $f(\mathbf{t}')$ which is one of the components of $\mathbf{t}$. Thus σ_1 also passes through $f(\mathbf{t}')$ before $\rho(1)$, contradicting Lemma 88.

□

Theorem 96, in which we show that u-equivalent LFL schemas have the same sets of u-needed if and while predicates, is the main result of this Section. Definition 91, Lemma 93 and Corollary 95 will also be quoted in later Sections.

Definition 91 (preupdated and postupdated symbols) Let S be a linear schema, let $p \in \mathit{whilePreds}(S)$ and $x \in \mathit{Symbols}(\mathit{body}_S(p))$ and assume $\mathit{above}_{\mathit{body}_S(p)}(x) = x$. We say that x is preupdated in S if every path through $\mathit{body}_S(p)$ passing through x also passes through some $g \in \mathcal{F}$ satisfying $g \rightsquigarrow_{\mathit{body}_S(p)} x$, and x is postupdated in S if every such path passes through some $g \in \mathcal{F}$ satisfying $\mathit{back}_S(p, g, x)$.

Proposition 92 *Let S be an LFL schema, let $p \in \mathit{whilePreds}(S)$ and $x \in \mathit{Symbols}(\mathit{body}_S(p))$ and assume $\mathit{above}_{\mathit{body}_S(p)}(x) = x$. Then x is either preupdated or postupdated in S.*

Proof. Let $\rho = \sigma l \sigma' \in \Pi(\mathit{body}_S(p))$ with $l \in \mathit{alphabet}(S)$ such that $\mathit{symbol}(l) = x$. If suffices to show that ρ passes through some $g \in \mathcal{F}$ satisfying either $g \rightsquigarrow_{\mathit{body}_S(p)} x$ or $\mathit{back}_S(p, g, x)$. Since S is free, there is an interpretation defining a path through S of which $l\sigma' \lessdot p = \mathsf{T} \gtrdot \sigma l$ is a segment. The existence of an appropriate $g \in \mathcal{F}$ follows from the freeness of S if $x \in \mathcal{P}$ and the liberality of S if $x \in \mathcal{F}$. $\square$

Lemma 93 *Let S be an LFL schema and let $p \in \mathit{whilePreds}(S)$. Let R be the set of predicates lying immediately below p in S. For each predicate $r \in R$, let $X(r) \in \{\mathsf{T}, \mathsf{F}\}$ and assume that $r \in \mathit{whilePreds}(S) \Rightarrow X(r) = \mathsf{T}$. Let $x \in \mathcal{P} \cup \mathcal{F}$ lie immediately below p in S and assume that x does not lie in the $X(r)$-part of any if predicate $r \in R$.*

(1) *Then $y \rightsquigarrow_{S(p)} x$ for some $y \in \mathcal{F}$ and there is a yx-segment in $S(p)$ which does not contain any letter $\lessdot r = X(r) \gtrdot$ for $r \in R$.*
(2) *There is a $gGgFx$-segment in the while schema $S(p)$ for $g \in \mathcal{F}$ and $G, F \in \mathcal{F}^*$, which does not contain any letter $\lessdot r = X(r) \gtrdot$ for $r \in R$.*

Proof.

(1) Let $\sigma l \sigma' \in \Pi(\mathit{body}_S(p))$ with $l \in \mathit{alphabet}(S)$ such that $\mathit{symbol}(l) = x$ and σ and σ' do not contain any letter $\lessdot r = X(r) \gtrdot$ for $r \in R$. If σl ends in a yx-segment for some $y \in \mathcal{F}$ then we are done; otherwise x is not preupdated in S, and hence by Proposition 92, x is postupdated in S and so the segment $\sigma' \lessdot p = \mathsf{T} \gtrdot \sigma l$ in $S(p)$ ends in a in a yx-segment for some $y \in \mathcal{F}$, as required.
(2) By Part (1) of this Lemma, and by induction on n, for every $n > 0$ there is an Hx-segment in the while schema $S(p)$ satisfying the condition given, with $|H| = n$. Choosing n to be greater than the number of function symbols in S gives the stated result.

$\square$

Lemma 94 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. Let $p \in$ whilePreds(S) and assume that there is a gGg-segment in the while schema $S(p)$ for $g \in \mathcal{F} \cap \mathcal{N}_S(u)$ and $G \in \mathcal{F}^*$, whose symbols all lie immediately below p in S. Then $p \in \mathcal{N}_S(u) \cap$ whilePreds(T) and g lies immediately below p in T.*

Proof. Since $g \in \mathcal{N}_S(u)$, there exists $H \in \mathcal{F}^*$ such that S contains either a gHq-segment for $q \in \mathcal{P} \cap \mathcal{N}_S(u)$ or a gHu-segment with $u \in \mathcal{V}$. We may assume the first case; the second may be reduced to the first by using Lemma 27.
Let $\bar{g} \in \mathit{alphabet}(S)$ be the assignment to g in S and let $\bar{g}\,\mu\,\bar{g}\mu_H < q = \mathsf{F} >$ be a $gGgHq$-segment in S such that the symbols of the segment $\bar{g}\,\mu\,\bar{g}$ lie immediately below p in S. Let $\tau \in \mathit{pre}(\Pi(S))$ be such that $\tau\bar{g} \in \mathit{pre}(\Pi(S))$. Thus $\tau(\bar{g}\mu)^n\bar{g}\mu_H < q = \mathsf{F} >$ is a $(gG)^n gHq$-prefix for every $n > 0$. Write $\sigma_n = \tau(\bar{g}\mu)^n\bar{g}\mu_H$. Since $q \in \mathcal{P} \cap \mathcal{N}_S(u)$, we claim there is a qu-couple $\{i, j\}$ for S which is $(\mathcal{P} - \{q'\}, \mathsf{T})$-finite for some $q' \in$ *whilePreds*(S) if $u = \omega$ and $(\mathcal{P}, \mathsf{T})$-finite if $u \in \mathcal{V}$. If $u \in \mathcal{V}$ then this follows from Part (1) of Lemma 74 and if $u = \omega$ then this follows from Part (2) of Lemma 74. Define

$$\mathcal{P}_u = \begin{cases} \mathcal{P} - \{q'\} & u = \omega \\ \mathcal{P} & u \in \mathcal{V}. \end{cases}$$

Hence by Lemma 60, for each $n > 0$ there is a $(\mathcal{P}_u, \mathsf{T})$-finite $(gG)^n gHqu$-couple for S whose common prefix in S is σ_n. For each $n > 0$, let $\{i_n, j_n\}$ be a $(gG)^n gHqu$-couple for S such that i_n is $(\mathcal{P}_u, \mathsf{T})$-minimal. Observe that $\mathit{grade}_S((\bar{g}\mu), r, \mathsf{T}) = 0$ for all $r \in$ *whilePreds*$(S) - \{p\}$. Thus by Lemma 60, the element

$$B(r, n) = \mathit{grade}(r, i_n, \mathsf{T})$$

is independent of n for all $r \in \mathcal{P} - \{p\}$, and finite for all $r \in \mathcal{P}_u$. Also, if $p \in \mathcal{P}_u$ then $\{B(p, n) |\, n > 0\}$ is an unbounded set of integers. On the other hand each set $\{i_n, j_n\}$ is also a $(gG)^n gHqu$-couple for T, by Proposition 57, and so similar results can be shown using T. Thus g lies immediately below some $p' \in$ *whilePreds*(T) and $B(r, n)$ is independent of n for all $r \in \mathcal{P} - \{p'\}$; also, if $p' \in \mathcal{P}_u$ then $\{B(p', n) |\, n > 0\}$ is an unbounded set of integers. If $p' \neq p$ then choosing $r \in \{p, p'\} \cap \mathcal{P}_u$ gives a contradiction, thus proving the Lemma. $\square$

Corollary 95 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. Let $p \in$ whilePreds(S) and let the symbol $x \in \mathcal{N}_S(u)$ satisfy either $x = p$ or x lies immediately below p in S. Then for $g \in \mathcal{F}$ and $G, H \in \mathcal{F}^*$, there is a $gGgHx$-segment in the while schema $S(p)$, all of whose symbols apart possibly from x lie immediately below p in S and such that g lies immediately below p in T.*

Proof. This follows immediately from Lemmas 93, Part (2) and Lemma 94 and if $x = p$, the fact that by the freeness of S there exists $y \in \mathcal{F}$ such that $y \rightsquigarrow_S p$ and y lies immediately below p in S. $\square$

Theorem 96 ($S \cong_u T$ implies conditions (2),(3) of u-similarity) *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. Then*

$$\mathcal{N}_S(u) \cap \mathit{ifPreds}(S) = \mathcal{N}_T(u) \cap \mathit{ifPreds}(T)$$

and

$$\mathcal{N}_S(u) \cap \mathit{whilePreds}(S) = \mathcal{N}_T(u) \cap \mathit{whilePreds}(T)$$

hold.

Proof. Let $p \in \mathit{whilePreds}(S) \cap \mathcal{N}_S(u)$. By Corollary 95, $p \in \mathit{whilePreds}(T) \cap \mathcal{N}_S(u) = \mathit{whilePreds}(T) \cap \mathcal{N}_T(u)$. Thus

$$\mathit{whilePreds}(S) \cap \mathcal{N}_S(u) \subseteq \mathit{whilePreds}(T) \cap \mathcal{N}_T(u)$$

and equality holds by interchanging S and T. By Theorem 75 we get $\mathcal{N}_S(u) \cap \mathit{ifPreds}(S) = \mathcal{N}_T(u) \cap \mathit{ifPreds}(T)$. $\square$

Theorem 103 and Theorem 105 are the main results of this section. Here we obtain information about the relations given in the title of this Section by studying (p, X)-minimal couples for suitably chosen (p, X), and by using Lemma 69 to infer information about their heads.

Lemma 97 (grade of prefix) *Let S be a linear schema and let $y \in \mathit{Symbols}(S)$. Let $p \in \mathit{Preds}(S)$ and let $X \in \{\mathsf{T}, \mathsf{F}\}$ and assume*

$$p \in \mathit{whilePreds}(S) \Rightarrow X = \mathsf{T}$$

holds. Then the following hold.

(1) *If $p \searrow_S y\ (X)$ then every y-prefix μ in S satisfies $\mathit{grade}(\mu, p, X) \geq 1$, and there exists such a prefix such that equality holds.*
(2) *If $\neg(p \searrow_S y\ (X))$ holds then there exists a y-prefix μ such that $\mathit{grade}(\mu, p, X) = 0$.*

Proof. Both parts are straightforward to prove. □

Lemma 98 (grade of segment) *Let S be a linear schema and assume that $f \rightsquigarrow_S x$ for some $f \in \mathcal{F}$ and $x \in \mathcal{P} \cup \mathcal{F}$. Then the following hold:*

(1) *If $p \in \mathit{whilePreds}(S)$ and $x \notin \mathit{Symbols}(\mathit{body}_S(p))$ then there exists a fx-segment ν in S such that $\mathit{grade}(\nu, p, \mathsf{T}) = 0$.*
(2) *If $p \in \mathit{whilePreds}(S)$ and either $\mathit{back}_S(p, f, x)$ or $f \notin \mathit{Funcs}(\mathit{body}_S(p)) \wedge x \in \mathit{Symbols}(\mathit{body}_S(p))$ then every fx-segment ν in S satisfies $\mathit{grade}(\nu, p, \mathsf{T}) \geq 1$, and there exists such a segment for which equality holds.*
(3) *If $p \in \mathit{whilePreds}(S)$ and $f \rightsquigarrow_{\mathit{body}_S(p)} x$ then there exists a fx-segment ν in S such that $\mathit{grade}(\nu, p, \mathsf{T}) = 0$.*
(4) *If $p \in \mathit{ifPreds}(S)$, $X \in \{\mathsf{T}, \mathsf{F}\}$ and $\mathit{outif}_S(p, X, f, x)$ and $p \searrow_S x\ (X)$, then every fx-segment ν in S satisfies $\mathit{grade}(\nu, p, X) \geq 1$ and there exists such a segment for which equality holds.*
(5) *If $p \in \mathit{ifPreds}(S)$, $X \in \{\mathsf{T}, \mathsf{F}\}$ and $\mathit{outif}_S(p, X, f, x) \wedge \neg(p \searrow_S x\ (X))$ holds, then there exists an fx-segment ν in S satisfying $\mathit{grade}(\nu, p, X) = 0$.*

Proof. All parts are straightforward to prove. □

Definition 99 (the index_S function) Let S be an LFL schema, let $u \in \mathcal{V} \cup \{\omega\}$ and let $q \in \mathit{Preds}(S) \cap \mathcal{N}_S(u)$. For any word $G \in \mathcal{F}^*$ such that S contains a Gq-segment, and $X \in \{\mathsf{T}, \mathsf{F}\}$ and $r \in \mathit{Preds}(S)$, let

$$\mathit{index}_S(Gqu, r, X) \in \mathbb{N} \cup \{\infty\}$$

be the minimal element of

$$\{grade(I, X, r) \mid I \text{ is a } Gqu\text{-couple for } S\},$$

provided this set is nonempty; otherwise $index_S(Gqu, r, X)$ is undefined.

Observe that if $S \cong_u T$ then $index_S(Gqu, r, X) = index_T(Gqu, r, X)$ by Proposition 57, provided one side is defined.

Lemma 100 *Let S be an LFL schema and $u \in \mathcal{V} \cup \{\omega\}$ and let $p, q \in Preds(S)$ and assume $q \in \mathcal{N}_S(u)$. Let $X \in \{\mathsf{T}, \mathsf{F}\}$. Let $f \in \mathcal{F}$ and $F \in \mathcal{F}^*$, and assume that S contains an fFq-segment and let $\mu(F), \mu(fF)$ be prefixes such that $\mu(F) <q = Z>$ and $\mu(fF) <q = Z>$ are (p, X)-minimal Fq- and fFq-prefixes in S for $Z \in \{\mathsf{T}, \mathsf{F}\}$. Then*

$$index_S(fFqu, p, X) - index_S(Fqu, p, X) = grade(\mu(fF), p, X) - grade(\mu(F), p, X)$$

holds provided all expressions are finite.

Proof. Assuming that $index_S(fFqu, p, X)$ and $index_S(Fqu, p, X)$ are defined, by Lemma 69 and Lemma 60 there are (p, X)-minimal Fq- and fFq-couples I and J respectively with $head_S(I) = \mu(F)$ and $head_S(J) = \mu(fF)$. We must have

$$|\bigcup_{k \in I} Tailterms_S(p, X, k, I)| = |\bigcup_{k \in J} Tailterms_S(p, X, k, J)|,$$

since otherwise we could construct a new couple with the tails of one element of $\{I, J\}$ and the head of the other, contradicting the (p, X)-minimality of I or J. Thus the result follows from Proposition 68. $\square$

Lemma 101 *Let S be an LFL schema and assume that there is a (p, X)-finite $fFqu$-couple for S for some $u \in \mathcal{V} \cup \{\omega\}$, $f \in \mathcal{F}$, $F \in \mathcal{F}^*$, $X \in \{\mathsf{T}, \mathsf{F}\}$ and $p, q \in \mathcal{P}$. Let x be the first letter in the word Fq. Then the following hold, provided that the index function defines a natural number in each case.*

(1) *If $p \in whilePreds(S)$, $X = \mathsf{T}$ and $outwhile_S(p, f, x) \vee back_S(p, f, x)$ then*

$$index_S(Fqu, \mathsf{T}, p) + 1 = index_S(fFqu, \mathsf{T}, p)$$

holds.

(2) *If $p \in whilePreds(S)$, $X = \mathsf{T}$ and either $f \rightsquigarrow_{body_S(p)} x$ or f does not lie in $body_S(p)$, then $index_S(Fqu, \mathsf{T}, p) = index_S(fFqu, \mathsf{T}, p)$.*

(3) *If $p \in ifPreds(S)$ and $outif_S(p, X, f, x) \vee thru_S(p, X, f, x)$, then*

$$index_S(Fqu, X, p) + 1 = index_S(fFqu, X, p)$$

holds.

(4) *If* $p \in \mathit{ifPreds}(S)$ *and either* $f \rightsquigarrow_{\mathit{part}_S^X(p)} x$ *or there exists an* fx*-segment in* S *not containing the letter* $\lessdot p = X \gtrdot$*, then*

$$\mathit{index}_S(Fqu, X, p) = \mathit{index}_S(fFqu, X, p).$$

Proof. For notational convenience we will assume $x \neq q$. Write $Fq = xF'$ and let $\mu(fF)$ be a segment in S such that $\mu(fF) \lessdot q = Z \gtrdot$ is a (p, X)-minimal fFq-prefix in S. Thus we may write $\mu(fF) = \mu_1 \bar{f} \mu_2 \bar{x} \mu_2$, where $\bar{f}$ and $\bar{x}$ are the assignments in S to f and x, $\mu_1 \bar{f}$ is a (p, X)-minimal f-prefix in S, $\bar{f} \mu_2 \bar{x}$ is a (p, X)-minimal fx-segment in S and $\bar{x} \mu_2 \lessdot q = Z \gtrdot$ is a (p, X)-minimal F'-segment in S. Let $\tau \bar{x}$ be a (p, X)-minimal x-prefix in S. Thus by Lemma 100, $\mathit{index}_S(fFq, v, p, X) - \mathit{index}_S(Fq, v, p, X) = \mathit{grade}(\mu_1 \bar{f} \mu_2 \bar{x} \mu_2, p, X) - \mathit{grade}(\tau \bar{x} \mu_2, p, X)$ and so the results follow from Lemma 97 applied to the prefixes $\tau \bar{f}$ and $\mu_1 \bar{x}$ and Lemma 98 applied to the fx-segment $\bar{f} \mu_2 \bar{x}$. $\square$

Lemma 102 *Let* $u \in \mathcal{V} \cup \{\omega\}$ *and let* S, T *be* u*-equivalent LFL schemas. Let* $p \in \mathit{whilePreds}(S)$ *and assume that there is a predicate* $q \in \mathcal{N}_S(u)$ *such that there is a* (p, T)*-finite* qu*-couple for* S *and that* S *contains an* fFq*-segment for* $f \in \mathcal{F}$ *and* $F \in \mathcal{F}^*$*. Let* x *be the first letter of* Fq*. Then*

$$(\mathit{outwhile}_S(p, f, x) \vee \mathit{back}_S(p, f, x)) \iff (\mathit{outwhile}_T(p, f, x) \vee \mathit{back}_T(p, f, x))$$

holds.

Proof. Clearly $\mathit{index}_S(Gq, u, p, \mathsf{T})$ is finite for all $G \in \mathcal{F}^*$. If

$$\mathit{outwhile}_S(p, f, x) \vee \mathit{back}_S(p, f, x)$$

holds then

$$\mathit{index}_S(Fq, u, \mathsf{T}, p) + 1 = \mathit{index}_S(fFq, u, \mathsf{T}, p)$$

and thus

$$\mathit{index}_T(Fq, u, \mathsf{T}, p) + 1 = \mathit{index}_T(fFq, u, \mathsf{T}, p)$$

holds. Since the conclusion of Part (2) in Lemma 101 is thus false for T we get $f \in \mathit{Funcs}(\mathit{body}_T(p))$ and $\neg(f \rightsquigarrow_{\mathit{body}_T(p)} x)$; and so $\mathit{outwhile}_T(p, f, x) \vee \mathit{back}_T(p, f, x)$ holds. The converse follows similarly. $\square$

Theorem 103 *Let* $u \in \mathcal{V} \cup \{\omega\}$ *and let* S, T *be* u*-equivalent LFL schemas.Let* $f \in \mathcal{F}$ *and* $x \in \mathcal{N}_S(u) \cup (\{u\} \cap \mathcal{V})$*. Let* $p \in \mathit{whilePreds}(S)$*. Then*

$$(\mathit{outwhile}_S(p, f, x) \vee \mathit{back}_S(p, f, x))$$

$$\iff$$

$$(\mathit{outwhile}_T(p, f, x) \vee \mathit{back}_T(p, f, x))$$

holds provided that at least one of the following conditions holds.

(1) $u \in \mathcal{V}$.
(2) $u = \omega$ *and there exists* $p' \in \mathit{whilePreds}(S) - \{p\}$ *such that* $x \in \mathcal{N}_S(p')$.
(3) $u = \omega$ *and* $\mathit{body}_S(p)$ *is not while-free.*

Proof.

(1) We may assume that $x \in \mathit{Symbols}(S)$ and there is a predicate $q \in \mathcal{N}_S(u)$ such that either $x = q$ or S contains an xFq-segment for some $F \in \mathcal{F}^*$; if this is false, then S contains an F-segment for some F beginning with x and ending with u, and so Lemma 27 may be used to eliminate this case. Thus the conclusion follows from Lemma 102 and Lemma 74, Part (1).
(2) The conclusion follows from Lemma 102 and Lemma 74, Part (2).
(3) Clearly for some $p' \in \mathit{whilePreds}(S)$ we have $x \in \mathcal{N}_S(p')$. If $p' \neq p$ then the conclusion follows from Part (2) of this Theorem. If $p = p'$, let q be a while predicate in the body of p. Then $x \in \mathcal{N}_S(q)$ and $q \neq p$ holds, so again the conclusion follows from Part (2) of this Theorem.

□

We will later prove that the three conditions given in Theorem 103 are unnecessary (Theorem 127).

Corollary 104 *Let* S, T *be u-equivalent LFL schemas for* $u \in \mathcal{V} \cup \{\omega\}$ *and let* $p \in \mathit{whilePreds}(S) \cap \mathcal{N}_S(u)$ *and* $f \in \mathcal{F}$. *Then*

$$\mathit{outwhile}_S(p, f, p) \iff \mathit{outwhile}_T(p, f, p)$$

holds.

Proof. Assume that $\mathit{outwhile}_S(p, f, p)$ holds. By Theorem 75, to show $\mathit{outwhile}_T(p, f, p)$ it suffices to prove that $f \in \mathit{Funcs}(\mathit{body}_T(p))$. If $u \in \mathcal{V}$ or there exists $p' \in \mathit{whilePreds}(S) - \{p\}$ such that $p \in \mathcal{N}_S(p')$, or there is a while predicate in the body of p in S, then the result follows from Theorem 103. Thus we may assume that these conditions are all false. Hence $u = \omega$ and there is no while predicate in the body of p in S, so f lies immediately below p in S. By Lemma 93, Part (2), there is a $gGgHf$-segment in the body of p in S, for $g \in \mathcal{F}$ and $G, H \in \mathcal{F}^*$. Thus if ab is a 2-letter subword of $gGgHf$, then

$$\neg(\mathit{outwhile}_S(p', a, b) \vee \mathit{back}_S(p', a, b))$$

holds for every $p' \in \mathit{whilePreds}(S) - \{p\}$. By Theorem 75, T also contains a $gGgHfp$-segment; and if ab is a 2-letter subword of $gGgHfp$, then by Theorem 103, Part (2),

$$\neg(\mathit{outwhile}_T(p', a, b) \vee \mathit{back}_T(p', a, b))$$

holds for every $p' \in \mathit{whilePreds}(S) - \{p\}$. Thus g cannot lie immediately below a while predicate $p' \neq p$ in T; but g must lie in the body of a while predicate in T, otherwise the gGg-segment in T could not exist. Hence $g \in \mathit{Funcs}(\mathit{body}_T(p))$. Assume that $f \notin \mathit{Funcs}(\mathit{body}_T(p))$. Thus there is a $gHfp$-segment in T which starts in the body of p, leaves the while schema $T(p)$, and then returns to p. Thus p lies immediately below a while predicate p' in T, and

$$\mathit{outwhile}_T(p', a, b) \vee \mathit{back}_T(p', a, b)$$

holds for some 2-letter subword ab of $gGgHfp$, giving a contradiction. Thus

$$\mathit{outwhile}_S(p, f, p) \Rightarrow \mathit{outwhile}_T(p, f, p)$$

follows and the converse holds similarly. $\square$

Theorem 105 (u-equivalence implies Condition (10) of u-similarity) *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. Suppose that $f \rightsquigarrow_S x$ for some $f \in \mathcal{F}$ and $x \in \mathcal{N}_S(u) \cup (\{u\} \cap \mathcal{V})$. Let $p \in \mathit{ifPreds}(S)$ and $Y \in \{\mathsf{T}, \mathsf{F}\}$. Then $(\mathit{outif}_S(p, Y, f, x) \vee \mathit{thru}_S(p, Y, f, x)) \iff (\mathit{outif}_T(p, Y, f, x) \vee \mathit{thru}_T(p, Y, f, x))$.*

Proof. This is similar to Theorem 103, using Parts (3) and (4) of Lemma 101. If $u = \omega$ then Theorem 90 must be used; otherwise Part (1) of Lemma 74 may be used. $\square$

Theorem 109 and Lemma 117 are the main results of this section. In order to strengthen Theorems 103 and 105, we need to refine the notion of the grade of a prefix and an interpretation.

Definition 106 (external and internal predicate terms) Let S be a linear schema and let $p \in \mathit{whilePreds}(S)$ and let $p(\mathbf{t})$ be a predicate term. Then we say that $p(\mathbf{t})$ is S-external if no component of $\mathbf{t}$ is an f-term for any $f \in \mathcal{F}$ lying in the body of p in S. We also say that $p(\mathbf{t})$ is S-internal if it is not S-external.

Definition 107 (external and internal grade of a prefix of a schema) Let S be a linear schema, let $p \in \mathcal{P}$, $X \in \{\mathsf{T}, \mathsf{F}\}$ and let $\mu \in \mathit{pre}(\Pi(S))$. We define

$$\mathit{grade}_S(\mu, p, X, EXT) \in \mathbb{N} \cup \{\infty\}$$

to be the number of S-external predicate terms $p(\mathbf{t})$ for which μ has a prefix

$$\mu' \lessdot p = X \gtrdot$$

such that

$$\mathcal{M}[\![\mathit{schema}(\mu')]\!]_e(\mathbf{refvec}_S(p)) = \mathbf{t}.$$

We define $\mathit{grade}_S(\mu, X, INT)$ similarly using S-internal instead of S-external.

Definition 108 (external and internal grade of interpretations)
Let i be an interpretation, let S be an LFL schema and let $p \in \mathit{whilePreds}(S)$ and $X \in \{\mathsf{T}, \mathsf{F}\}$. Then $\mathit{grade}_S(i, p, X, EXT) \in \mathbb{N} \cup \{\infty\}$ is the number of S-external predicate terms $p(\mathbf{t})$ for which $p^i(\mathbf{t}) = X$. We define $\mathit{grade}_S(i, p, X, INT)$ similarly using S-internal instead of S-external. If I is a pu-couple for some $u \in \mathcal{V} \cup \{\omega\}$, then $\mathit{grade}_S(I, p, X, INT)$ is the minimal element of $\{\mathit{grade}_S(i, p, X, INT) | \, i \in I\}$.

Observe that the schema S is a parameter of the external or internal grade (as distinct from the standard grade, as in Definitions 61 and 62, in which S is not a parameter); a predicate term $p(\mathbf{t})$ may be external with respect to one schema, but not with respect to another. However Theorem 109 qualifies this.

Theorem 109 (u-equivalence implies same external predicate terms) *Let S, T be u-equivalent LFL schemas for $u \in \mathcal{V} \cup \{\omega\}$ and let $p \in \mathit{whilePreds}(S) \cap \mathcal{N}_S(u)$. Then a predicate term $p(\mathbf{t})$ is S-external if and only if $p(\mathbf{t})$ is T-external.*

Proof. This follows from Corollary 104. □

Definition 110 (external and internal terms of a couple) Let S be an LFL schema, let $u \in \mathcal{V} \cup \{\omega\}$, and let $p \in \mathit{whilePreds}(S)$ and $q \in \mathit{Preds}(S)$. Let I be a *qu*-couple for S and let $k \in I$ and $Y \in \{\mathsf{T}, \mathsf{F}\}$. Then we define

$$\mathit{Tailterms}_S(p, k, INT, I)$$

to be the set of S-internal predicate terms in $\mathit{Tailterms}_S(p, k, I)$ and $\mathit{Tailterms}_S(p, Y, INT, k, I)$ to be the set of S-internal predicate terms in

$$\mathit{Tailterms}_S(p, Y, k, I).$$

We define $\mathit{Tailterms}_S(p, Y, EXT, k, I)$, $\mathit{Headterms}_S(p, Y, INT, k, I)$ and $\mathit{Headterms}_S(p, Y, EXT, k, I)$ similarly.

Proposition 111 (computing the internal (p, X)-grade of a couple) *Let S be an LFL schema, let $q \in \mathit{Preds}(S)$ and $p \in \mathit{whilePreds}(S)$, let $X \in \{\mathsf{T}, \mathsf{F}\}$, let $u \in \mathcal{V} \cup \{\omega\}$ and let $I = \{i, j\}$ be an (S, p, X)-normalised qu-couple for S. Let $m = \mathit{grade}_S(I, p, X, INT)$. Then*

$$m = \mathit{grade}_S(\mathit{head}_S(I), p, X, INT) + |\bigcup_{k \in I} \mathit{Tailterms}_S(p, X, INT, k, I)\,|$$

holds provided both sides are finite. A similar statement holds for external predicate terms.

Proof. This follows immediately from the freeness of S. $\square$

Proposition 112 is easily shown.

Proposition 112 (identifying internal predicate terms)
Let S be a linear schema and let $p \in \mathit{whilePreds}(S)$, $X \in \{\mathsf{T}, \mathsf{F}\}$ and

$$\mu \ \lessdot p = X > \in \ \mathit{pre}(\Pi(S)).$$

Then the predicate term

$$p(\mathcal{M}[\![\mathit{schema}(\mu)]\!]_e(\mathbf{refvec}_S(p)))$$

is S-internal if and only if $\mu \ \lessdot p = X >$ is an fp-prefix for some $f \in \mathit{Funcs}(\mathit{body}_S(p))$. $\square$

Lemma 113 *Let S be a free linear schema, let $p \in \mathit{whilePreds}(S)$ and let*

$$\rho_1\sigma,\ \rho_2\sigma \in \mathit{pre}(\Pi(S)).$$

Assume that ρ_2 does not pass through $\lessdot p = \mathsf{T} >$.

(1) *If the prefix ρ_1 also does not pass through $\lessdot p = \mathsf{T} >$, then*

$$\mathit{grade}_S(\rho_1\sigma, p, \mathsf{F}, INT) = \mathit{grade}_S(\rho_2\sigma, p, \mathsf{F}, INT)$$

holds.

(2) *If the prefix ρ_1 contains at least one occurrence of $<\!\!p = \mathsf{F}\!\!>$ after an occurrence of $<\!\!p = \mathsf{T}\!\!>$, then*

$$grade_S(\rho_1\sigma, p, \mathsf{F}, INT) > grade_S(\rho_2\sigma, p, \mathsf{F}, INT)$$

holds.

(3) *If every occurrence of $<\!\!p = \mathsf{F}\!\!>$ in ρ_1 is immediately preceded by a path through $body_S(p)$, then*

$$grade_S(\rho_1\sigma, p, \mathsf{F}, EXT) \leq grade_S(\rho_2\sigma, p, \mathsf{F}, EXT).$$

If in addition ρ_2 contains at least one occurrence of $<\!\!p = \mathsf{F}\!\!>$, then the inequality is strict.

Proof. This follows easily from Proposition 112 and the fact that since S is free, any path in $\Pi(body_S(p))$ ends in an fv-segment for some $f \in Funcs(body_S(p))$ and $v \in Refset_S(p)$. $\square$

Lemma 114 *Let S be an LFL schema, let $u \in \mathcal{V} \cup \{\omega\}$, let $p \in whilePreds(S)$ and $q \in Preds(S)$ and let I be a (p, F)-finite, (S, p, F)-normalised qu-couple for S. Suppose that $head_S(I) = \rho\sigma$, for a prefix ρ which does not pass through $<\!\!p = \mathsf{T}\!\!>$.*

(1) *Let J' be a (p, F)-finite, (S, p, F)-normalised qu-couple for S obtained from I by replacing $head_S(I)$ by a prefix $\rho'\sigma$ such that ρ' also does not pass through $<\!\!p = \mathsf{T}\!\!>$. Then*

$$grade_S(I, p, INT, \mathsf{F}) = grade_S(J', p, INT, \mathsf{F})$$

holds.

(2) *Let J'' be a (p, F)-finite, (S, p, F)-normalised qu-couple for S obtained from I by replacing $head_S(I)$ by a prefix $\rho''\sigma$ such that ρ'' contains at least one occurrence of $<\!\!p = \mathsf{F}\!\!>$ after an occurrence of $<\!\!p = \mathsf{T}\!\!>$, then*

$$grade_S(I, p, INT, \mathsf{F}) < grade_S(J'', p, INT, \mathsf{F})$$

holds.

(3) *Assume that J''' is a (p, F)-finite, (S, p, F)-normalised qu-couple for S obtained from I by replacing $head_S(I)$ by a prefix $\rho'''\sigma$ such that every occurrence of $<\!\!p = \mathsf{F}\!\!>$ in ρ''' is immediately preceded by a path through $body_S(p)$, then*

$$grade_S(J''', p, EXT, \mathsf{F}) \leq grade_S(I, p, EXT, \mathsf{F})$$

holds, with strict inequality if ρ contains at least one occurrence of $<\!\!p = \mathsf{F}\!\!>$.

Proof. In Case (1), it suffices just to show $grade_S(I, p, INT, \mathsf{F}) \leq grade_S(J', p, INT, \mathsf{F})$, since the other inequality follows by interchanging I and J'.

The statements

$$grade_S(head_S(I), p, \mathsf{F}, INT) \leq grade_S(head_S(J'), p, \mathsf{F}, INT),$$

$$grade_S(head_S(I), p, \mathsf{F}, INT) < grade_S(head_S(J''), p, \mathsf{F}, INT)$$

and

$$grade_S(head_S(J'''), p, \mathsf{F}, EXT) \leq grade_S(head_S(I), p, \mathsf{F}, EXT)$$

(with strict inequality in the last case under the conditions given) follow immediately from the appropriate part of Lemma 113. By Proposition 111, it suffices to show that

$$|\bigcup_{k \in I} Tailterms_S(p, \mathsf{F}, INT, k, I)| \leq |\bigcup_{k \in I} Tailterms_S(p, \mathsf{F}, INT, k, J')|,$$

$$|\bigcup_{k \in I} Tailterms_S(p, \mathsf{F}, INT, k, I)| \leq |\bigcup_{k \in I} Tailterms_S(p, \mathsf{F}, INT, k, J'')|,$$

and

$$|\bigcup_{k \in I} Tailterms_S(p, \mathsf{F}, EXT, k, J''')| \leq |\bigcup_{k \in I} Tailterms_S(p, \mathsf{F}, EXT, k, I)|$$

hold. Observe that by Lemma 67, the last inequality is implied by

$$|\bigcup_{k \in I} Tailterms_S(p, \mathsf{F}, INT, k, I)| \leq |\bigcup_{k \in I} Tailterms_S(p, \mathsf{F}, INT, k, J''')|.$$

Let $p(\mathbf{t}) \in Tailterms_S(p, \mathsf{F}, INT, k, I)$; then

$$\theta_{S,p,\mathsf{F},I,J}(p(\mathbf{t})) \in Tailterms_S(p, \mathsf{F}, INT, k, J)$$

holds for all $J \in \{J', J'', J'''\}$, using Proposition 112, the definition of $\theta_{S,p,\mathsf{F},I,J}$, and the fact that the prefix ρ does not pass through $<p = \mathsf{T}>$. Thus the results follow from the injectivity of $\theta_{S,p,\mathsf{F},I,J}$. $\square$

Proposition 115 is easily shown.

Proposition 115 *Let S be a linear schema and let $q \in whilePreds(S)$ be such that $body_S(q)$ is while-free and let μ be a path through S which is (p, T)-finite for every $p \in whilePreds(S) - \{q\}$. Then μ is (p, F)-finite for every $p \in whilePreds(S)$.* $\square$

Corollary 116 *Let S be an LFL schema, $u \in \mathcal{V} \cup \{\omega\}$ and let $q \in Preds(S) \cup \mathcal{N}_S(u)$. For each $p \in ifPreds(S)$ let $Y(p) \in \{\mathsf{T}, \mathsf{F}\}$. Let $R \subseteq whilePreds(S)$. Then there is a qu-couple for S which is $(p, Y(p)$-finite for every $p \in ifPreds(S)$, is (p, F)-finite for every $p \in R$ and is (p, T)-finite for every $p \in whilePreds(S) - R$ except at most one, whose body in S is while-free.*

Proof. If $u \in \mathcal{V}$ then this follows from Part (1) of Lemma 74, since the paths defined by a qu-couple for S are finite. If $u = \omega$ then the result follows from Proposition 115 and Theorem 90 (if $q \in \mathit{ifPreds}(S)$) or Lemma 72 (if $q \in \mathit{whilePreds}(S)$). □

Lemma 117 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. Suppose that $f \rightsquigarrow_S x$ for some $f \in \mathcal{F}$ and $x \in \mathcal{N}_S(u)$. Let $p \in \mathit{whilePreds}(S)$. If*

$$x \notin \mathit{Symbols}(\mathit{body}_S(p)) \cup \mathit{Symbols}(\mathit{body}_T(p)),$$

then

$$\mathit{outwhile}_S(p, f, x) \iff \mathit{outwhile}_T(p, f, x)$$

holds.

Proof. We may assume that $p \neq x$; otherwise the Lemma follows from Corollary 104. Also we may assume that $u = \omega$, otherwise the Lemma follows from Part (1) of Theorem 103. We will assume that $\mathit{outwhile}_S(p, f, x) \wedge \neg \mathit{outwhile}_T(p, f, x)$ holds and deduce a contradiction; the converse may be proved similarly. Since $x \in \mathcal{N}_S(\omega)$, there is an Fq-segment in S and T for some $F \in \mathcal{F}^*$ and $q \in \mathcal{P} \cap \mathcal{N}_S(\omega)$, such that x is the first letter of Fq. By Corollary 116 there is a (p, F)-finite Fq-couple for T. Let I be an Fq-couple for T for which $\mathit{grade}_T(I, p, \mathsf{F}, \mathit{INT})$ is minimal, and such that $\mathit{grade}(I, p, \mathsf{F})$ is minimal for all such Fq-couples for T.
Write $\mathit{head}_T(I) = \rho\sigma$, where σ $<q = \mathsf{T}>$ is an Fq-segment. By Lemma 60 there is a (T, p, F)-normalised fFq-couple J' for T obtained by replacing $\mathit{head}_T(I)$ by a prefix $\rho'\sigma$ such that $\mathit{head}_T(J')$ $<q = \mathsf{T}>$ is an fFq-segment in T; since $\mathit{outwhile}_T(p, f, x)$ is false, we may assume that ρ' does not pass through $<p = \mathsf{T}>$. By Lemma 114, Parts (1) and (2), we get

$$\mathit{grade}_T(J', p, \mathsf{F}, \mathit{INT}) \leq \mathit{grade}_T(J, p, \mathsf{F}, \mathit{INT}).$$

We now get a contradiction by looking at S. Assume that $\mathit{outwhile}_S(p, f, x)$ holds. Clearly $\mathit{head}_S(J')$ $<q = \mathsf{T}>$ is an fFq-segment, so we may write $\mathit{head}_S(J') = \nu\tau$, where τ $<q = \mathsf{T}>$ is an Fq-segment and thus begins with x. Since $\mathit{outwhile}_S(p, f, x)$ holds and $p \neq x$, the prefix ν passes through f (and hence $<p = \mathsf{T}>$) before passing through $<p = \mathsf{F}>$. By Lemma 60 there is an (S, p, F)-normalised Fq-couple J'' for S obtained from J' by replacing ν with a prefix which does not pass through $<p = \mathsf{T}>$. Hence

$$\mathit{grade}_S(J'', p, \mathsf{F}, \mathit{INT}) < \mathit{grade}_S(J', p, \mathsf{F}, \mathit{INT})$$

holds by Lemma 114, Part (2). By Theorem 109 and the first inequality above we get $\mathit{grade}_S(J'', p, \mathsf{F}, \mathit{INT}) < \mathit{grade}_S(J, p, \mathsf{F}, \mathit{INT})$, contradicting the minimality condition on J. This proves the Lemma. □

Theorem 126, in which we show that the set of symbols lying below a u-needed while predicate is preserved by u-equivalence for LFL schemas, and Theorem 118, giving a similar result for while predicates lying under predicates if $u = \omega$, are the main results of this section.

Theorem 118 (ω-equivalence implies condition (7) of ω-similarity) *Let S and T be ω-equivalent LFL schemas. Then*

$$p' \searrow_S p\,(Z) \iff p' \searrow_T p\,(Z)$$

holds if $p \in \mathit{whilePreds}(S)$ and $p' \in \mathit{Preds}(S)$ and $Z \in \{\mathsf{T}, \mathsf{F}\}$.

Proof. Assume that $\neg(p' \searrow_S p\,(Z))$ holds. We may assume that $p' \neq p$, else the conclusion is obvious. By Lemma 72 there is a $p\omega$-couple $\{i, j\}$ for S (and hence T) such that $\pi_S(j, e)$ terminates and $p'^i(\mathbf{t}) = \neg Z$ for all vector terms $\mathbf{t}$, and i is (q, T)-finite for every $q \in \mathit{whilePreds}(S) - \{p'\}$. If $p' \searrow_T p\,(Z)$ holds, then these conditions and Theorem 96 imply that $\pi_T(i, e)$ terminates, giving a contradiction. Thus $\neg(p' \searrow_T p\,(Z))$ holds, and the converse follows similarly. $\square$

As the two v-equivalent LFL schemas below show, Theorem 118 cannot be generalised to the case where a variable v replaces ω, even if $p \in \mathcal{N}_S(v)$ holds.

while $p(u)$ *do* $u := k(u)$;
if $p'(w)$ *then* $v := f(u)$;

if $p'(w)$ *then*
{
while $p(u)$ *do* $u := k(u)$;
$v := f(u)$;
}

Lemma 119 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. Let $p, q \in \mathit{whilePreds}(S) \cap \mathcal{N}_S(u)$ and assume that q lies immediately below p in S.*

(1) *Assume that p lies in the body of a while predicate r in T. Then q also lies in $\mathit{body}_T(r)$; in particular, $q \neq r$.*

(2) *If* $outwhile_S(q,f,x))$ *holds for some symbol* $x \in \mathcal{N}_S(u)$ *lying in both* $body_S(p)$ *and* $body_T(p)$*, then* q *lies in* $body_T(p)$*.*

Proof. We may assume that $u \in \mathcal{V}$, since if $u = \omega$ then both parts of this lemma follow from Theorem 118. Observe first that by Corollary 95, for some $F \in \mathcal{F}^*$ and $g \in Funcs(body_S(p)) \cap Funcs(body_T(p))$ there is a gFq-segment in S and hence T whose symbols all lie immediately below p in S.

(1) Clearly $r \neq p$. Thus $outwhile_S(r,a,b) \vee back_S(r,a,b)$ is false for any subword ab of gFq. Thus if p lies in $body_T(r)$, then also q lies in $body_T(r)$, since otherwise $outwhile_T(r,a,b)$ would hold for some subword ab of gFq, contradicting Theorem 103, Part (1).

(2) Assume that the conclusion is false. Suppose that p lies immediately below T; then the existence of a gFq-segment in T implies $p \ll_T above_T(q)$. But since $q \in \mathcal{N}_S(u)$, Lemma 74 and Lemma 60 imply that there is a qu-couple $\{i,j\}$ for T (and hence S) satisfying $p^i(\mathbf{t}) = p^j(\mathbf{t}) = \mathsf{F}$ for all $\mathbf{t}$, clearly contradicting $q \in Preds(body_S(p))$. Thus p lies immediately below a while predicate r in T, whose body in T also contains q, by Part (1) of the present Lemma. If $\neg(p \ll_{body_T(r)} above_{body_T(r)}(q))$ holds, then $back_T(r,a,b)$ would hold for some subword ab of gFq, which is impossible by an argument similar to that used in the proof of Part (1) of the present Lemma. Hence $p \ll_{body_T(r)} above_{body_T(r)}(q)$ holds and so $back_T(r,f,x)$ follows. But then $outwhile_S(r,f,x) \vee back_S(r,f,x)$ holds by Theorem 103, parts (1) and since $above_{body_S(p)}(f) = q$ holds by the hypotheses, the predicate r lies in $body_S(q)$. Since p lies immediately below r in T, by Corollary 95 there is a $g'F'p$-segment in T for $F' \in \mathcal{F}^*$ and $g' \in Funcs(body_S(r) \cap Funcs(body_T(r)$ whose symbols all lie immediately below r in T. Thus $outwhile_T(q,a,b) \vee back_T(q,a,b)$ is false for any subword ab of $g'F'p$. This statement also holds in S, by Theorem 103, part (1) and this contradicts the fact that r lies in $body_S(q)$ whereas p does not. Thus q lies in $body_T(p)$ as required.

□

Corollary 120 *Let* $u \in \mathcal{V} \cup \{\omega\}$ *and let* S,T *be u-equivalent LFL schemas. Let* $r_1,\ldots,r_n \in whilePreds(S) \cap \mathcal{N}_S(u)$*. Assume that each* r_i *lies immediately below* r_{i+1} *in* S*. If a symbol* x *lies in each* $body_T(r_i)$*, then each* r_i *lies in* $body_T(r_{i+1})$*.*

Proof. Let $i < n$. Since $body_T(r_i)$ and $body_T(r_{i+1})$ have a common symbol, one element of the set $\{r_i, r_{i+1}\}$ lies in the body of the other in T. Since $above_{body_S(r_{i+1})}(r_i) = r_i$ holds, we get $r_{i+1} \searrow_T r_i$ from Lemma 119, Part (1). □

Corollary 121 *Let* $u \in \mathcal{V} \cup \{\omega\}$ *and let* S,T *be u-equivalent LFL schemas. Let* $z \in \mathcal{N}_S(u)$*. Assume that* $p \searrow_S z \iff p \searrow_T z$ *for all* $p \in whilePreds(S)$*. Then the while predicates containing* z *in their bodies are nested in the same order in* S *as in* T*.*

Proof. This follows immediately from Corollary 120. $\square$

Lemma 122 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. Suppose that $f \rightsquigarrow_S x$ for $f \in \mathcal{F}$ and $x \in \mathcal{N}_S(u)$. Assume that $r \searrow_S x \iff r \searrow_T x$ for all $r \in \mathit{whilePreds}(S)$. Then $r \searrow_S f \iff r \searrow_T f$ for all $r \in \mathit{whilePreds}(S)$.*

Proof. Let $r \in \mathit{whilePreds}(S)$. Suppose that $r \searrow_S f$. We will show that $r \searrow_T f$. The result then will follow by interchanging S and T. We consider two cases separately. Assume first that the symbol x does not lie in the body of r in S (or T, by hypothesis). Here $\mathit{outwhile}_S(r, f, x)$ holds and so by Lemma 117 $\mathit{outwhile}_T(r, f, x)$ holds. Thus $r \searrow_T f$ holds.
Now assume instead that the symbol $x \in \mathit{Symbols}(\mathit{body}_S(r))$ and hence

$$x \in \mathit{Symbols}(\mathit{body}_T(r)).$$

By Corollary 120, we may assume that r lies immediately above $\{f, x\}$ in S. Suppose first that $\mathit{above}_{\mathit{body}_S(r)}(f) = q \neq f$. By the assumption just made, $\mathit{body}_S(q)$ does not also contain x. Thus $\mathit{body}_T(q)$ contains f by the first case applied to q, and $\mathit{body}_T(r)$ contains q and hence f by Lemma 119, Part (2). Thus we may assume that $\mathit{above}_{\mathit{body}_S(r)}(f) = f$. By Corollary 95 there is a yFf-segment in $\mathit{body}_S(r)$ for $F \in \mathcal{F}^*$ and $y \in \mathit{Funcs}(\mathit{body}_S(r)) \cap \mathit{Funcs}(\mathit{body}_T(r))$, such that the symbols in the word yFf all lie immediately below r in S. Thus the word $yFfx$ does not contain a subword ab for which $\mathit{outwhile}_S(r', a, b) \vee \mathit{back}_S(r', a, b)$ holds for any while predicate $r' \neq r$. On the other hand x and y both lie in $\mathit{body}_T(r)$, so if f does not, then $yFfx$ contains a subword ab for which $\mathit{outwhile}_T(r', a, b) \vee \mathit{back}_T(r', a, b)$ holds if r' lies immediately above r in T. This contradicts Theorem 103 applied to T and r', since r' contains a while predicate in its body in T. $\square$

Lemma 123 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. Let $Y \in \{\mathsf{T}, \mathsf{F}\}$. Suppose that an if predicate $q \in \mathcal{N}_S(u)$ lies immediately below a while predicate p in S. Then $\neg(q \searrow_T p\,(Y))$ holds.*

Proof. Since q lies immediately below p in S, by Corollary 95 there is a yFq-segment in $S(p)$ for $F \in \mathcal{F}^*$ and $y \in \mathit{Funcs}(\mathit{body}_S(p)) \cap \mathit{Funcs}(\mathit{body}_T(p))$ such that all symbols in the word yFq lie immediately below p in S. Thus the word yFq does not contain a subword ab for which $\mathit{outwhile}_S(r, a, b) \vee \mathit{back}_S(r, a, b)$ holds for any while predicate $r \neq p$. On the other hand if $q \searrow_T p\,(Y)$ holds, then the yFq-segment in T begins and ends at the if schema $T(q)$ and so $\mathit{outwhile}_T(r, a, b) \vee \mathit{back}_T(r, a, b)$ must hold for a subword ab of yFq and the while predicate r lying immediately above q in T, contradicting Theorem 103, parts (1) or (3). $\square$

Lemma 124 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. Let $X \in \{\mathsf{T}, \mathsf{F}\}$. Suppose that $\mathit{outif}_S(q, X, f, x) \vee \mathit{thru}_S(q, X, f, x)$ for $f \in \mathcal{F}$ and $x \in \mathcal{N}_S(u) \cup$*

$(\{u\} \cap \mathcal{V})$ *and* $q \in \mathit{ifPreds}(S)$. *Then there exists* $f' \in \mathcal{F}$ *and* $Y \in \{\mathsf{T}, \mathsf{F}\}$ *such that* $\mathit{outif}_T(q, Y, f', x)$.

Proof. This follows immediately from Theorem 105 and Proposition 31. $\square$

Lemma 125 *Let* $u \in \mathcal{V} \cup \{\omega\}$ *and let* S, T *be* u*-equivalent LFL schemas. Let* $X \in \{\mathsf{T}, \mathsf{F}\}$*,* $f \in \mathcal{F}$*,* $x \in \mathcal{N}_S(u) \cup (\{u\} \cap \mathcal{V})$ *and* $q \in \mathit{ifPreds}(S)$*. Suppose that* $\mathit{outif}_S(q, X, f, x) \vee \mathit{thru}_S(q, X, f, x)$ *holds and assume that all members of the set* $\{x\} \cup \{g \in \mathcal{F} | \, g \rightsquigarrow_S x\}$ *lie in the bodies of the same set of while predicates in* S *as in* T*. Then* q *lies immediately below a while predicate* p *in* S *if and only if* q *lies immediately below* p *in* T*.*

Proof. We will assume that q lies immediately below p in S, and prove that q lies immediately below p in T; the converse follows immediately from Theorem 105.
In order to show this, by Lemma 124 we may assume that $\mathit{outif}_S(q, X, f, x)$ holds. We first show that $p \searrow_T q$ holds, and to do this we consider two cases separately.
Suppose that $p \searrow_S x$. Observe that $\mathit{outif}_T(q, X, f, x) \vee \mathit{thru}_T(q, X, f, x)$ holds by Theorem 105. If $\mathit{thru}_T(q, X, f, x)$ holds, then since f and x both lie in $\mathit{body}_S(p)$ and hence $\mathit{body}_T(p)$, the predicate q also does. If $\mathit{outif}_T(q, X, f, x)$ holds, then $q \searrow_T f\,(X)$ holds and since $p \searrow_T f$ holds, either $p \searrow_T q$, or $q \searrow_T p\,(X)$ holds. By Lemma 123 the latter is impossible, so the former holds.
Now assume instead that $\neg(p \searrow_S x)$ holds. Thus $\neg(p \searrow_T x) \wedge (p \searrow_T f)$ holds by the hypotheses. Assume that $\neg(p \searrow_T q)$ holds; then since $q \searrow_T p\,(X)$ is false by Lemma 123, clearly $\neg(\mathit{outif}_T(q, X, f, x))$ holds, and so $\mathit{thru}_T(q, X, f, x)$ follows from Theorem 105. Thus $\mathit{outif}_T(q, \neg X, f', x)$ for some $f' \in \mathcal{F}$. Thus $q \searrow_T f'$ and so $\neg(p \searrow_T f')$ and hence $\neg(p \searrow_S f')$ hold. But $\mathit{outif}_T(q, \neg X, f', x)$ also implies $\mathit{outif}_S(q, \neg X, f', x) \vee \mathit{thru}_S(q, \neg X, f', x)$ by Theorem 105, contradicting the fact that $\mathit{body}_S(p)$ contains q, but not f' or x. Thus $p \searrow_T q$ holds.
Having shown $p \searrow_T q$, we now show that $\mathit{above}_{\mathit{body}_T(p)}(q) = q$. If this is false, then q lies immediately below a while predicate p' in T with $p \searrow_T p'$. By Lemma 124 there exists $f' \in \mathcal{F}$ satisfying the same hypotheses given in the present Lemma for f. Thus $p' \searrow_S q$ holds by using the argument above with f, p and S replaced by f', p' and T. Since q lies immediately below p in S, we get $p' \searrow_S p$. However this reversal of the nesting of p and p' contradicts Lemma 119 applied to p' and q in T, using p for r, thus proving $\mathit{above}_{\mathit{body}_T(p)}(q) = q$. $\square$

Theorem 126 (u-equivalence implies Condition (8) of u-similarity) *Let* $u \in \mathcal{V} \cup \{\omega\}$ *and let* S, T *be* u*-equivalent LFL schemas. Let* $x \in \mathcal{N}_S(u)$*. Then* $p \searrow_S x \iff p \searrow_T x$ *holds for all* $p \in \mathit{whilePreds}(S)$*.*

Proof. Let xw be a u-dependence sequence for x in S. We will prove the result using induction on $|w|$. If $|w| = 0$ then $u = \omega$ and $x \in \mathit{whilePreds}(S)$ and so the result follows from Theorem 118. Thus we may assume that $|w| \geq 1$.

If $x \in \mathcal{F}$ then either the first letter of w is $y \in \mathcal{N}_S(u)$, in which case the Theorem follows from Lemma 122 and the inductive hypothesis, or it is $u \in \mathcal{V}$, in which case the Theorem follows from Part (1) of Theorem 103.

If $x \in \mathit{whilePreds}(S)$, then $x \searrow_S z$ for some $z \in \mathcal{N}_S(u)$ occurring in w. By the inductive hypothesis applied to z, we have $x \searrow_T z$. If also $p \searrow_S x$ for some $p \in \mathit{whilePreds}(S)$ then $p \searrow_S z$ and so $p \searrow_T f$ hold by the inductive hypothesis applied to z. By Corollary 121 we get $p \searrow_T x$. If on the other hand $\neg(p \searrow_S x)$ holds, we must consider two cases. If $\neg(p \searrow_S z)$ holds, then $\neg(p \searrow_T z)$ holds by the inductive hypothesis applied to z; but since $x \searrow_T z$ we get $\neg(p \searrow_T x)$. If $p \searrow_S z$ holds, then since $\neg(p \searrow_S x) \wedge (x \searrow_S z)$ holds we get $x \searrow_S p$. By Corollary 121 and the inductive hypothesis applied to z, we get $x \searrow_T p$, and so $\neg(p \searrow_T x)$ holds.

Lastly we consider the case where $x \in \mathit{ifPreds}(S)$. Suppose that $x \searrow_S f$ holds for some $f \in \mathcal{F}$ occurring in w, and f is the last such function symbol in w. Let z be the symbol after f in w. Clearly $f \rightsquigarrow_S z$ holds. By Lemma 125 and the inductive hypothesis applied to any $f' \in \mathcal{F}$ satisfying $f' \rightsquigarrow_S z$, a while predicate p lies immediately above x in S if and only if p lies immediately above x in T. Thus the general result follows from Corollary 121 applied to f.

If no such such function symbol f occurs in w, then $u = \omega$ and $x \searrow_S q$ holds for some $q \in \mathit{whilePreds}(S)$, and so the result follows from Theorem 118. $\square$

Observe that Theorem 126 implies that if S and T are u-equivalent LFL schemas, and $q \in \mathit{whilePreds}(S) \cap \mathcal{N}_S(u)$, then $\mathit{above}_S(x) = \mathit{above}_T(x)$ for all $x \in \mathit{body}_S(q) \cap \mathcal{N}_S(u)$.

Theorem 127 shows that the conclusion of Theorem 103 holds even without the three conditions listed in that Theorem; thus the relation $back_S(p, f, x)$ for u-needed symbols is always preserved by u-equivalence.

Theorem 127 (u-equivalence implies Condition (9) of u-similarity) *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. Let $p \in Symbols(whilePreds(S))$ and $x \in \mathcal{N}_S(u)$. Then*

$$back_S(p, f, x) \iff back_T(p, f, x)$$

holds.

Proof. We may assume that $u = \omega$ and that $body_S(p)$ is while-free; otherwise the result follows from Theorem 103 and Theorem 126. By Theorem 126 we get $p \in Symbols(whilePreds(T))$. Suppose that $\neg(back_S(p, f, x))$ holds, and hence $f \rightsquigarrow_{body_S(p)} x$. We will show that $\neg(back_T(p, f, x))$ holds; and then the converse will follow similarly.
Assume that $f \rightsquigarrow_S x\,(n)$. Let $\sigma \in preSymbols((body_S(p)))$ end in an fx-segment. Let Q be the set of all if predicates q such that σ passes through $<q = X(q)>$ for some $X(q) \in \{\mathsf{T}, \mathsf{F}\}$. Since $x \in \mathcal{N}_S(\omega)$, there is a word w whose first letter is x and whose last letter is a predicate in $\mathcal{N}_S(\omega)$ such that there is a w-segment in S. Let Ω be the set of all pairs $(q, \neg X(q))$ for $q \in Q$. Since $body_S(p)$ is while-free, Ω is finite and reasonable (Definition 63). By Theorem 90 (if w ends in an if predicate) or Lemma 72 (if w ends in a while predicate) there is a $w\omega$-couple $I = \{i, j\}$ in S which is Ω-finite. Choose $I = \{i, j\}$ to be Ω-minimal. To simplify notation we will assume that $x \in \mathcal{P}$ and so $w = p$. Define

$$\pi_S(k, e) = head_S(I) <x = Y_k> tail_S(k, I)$$

for $k \in \{i, j\}$ and $Y_i \neq Y_j$. We now show that for one $k \in \{i, j\}$ the prefix

$$head_S(I) <x = Y_k>$$

ends in σ and hence $\{i, j\}$ is an $fx\omega$-couple. If this were false, then $head_S(I) <x = Y_k>$ would have to enter the $\neg X(q)$-part of some $q \in Q$, contradicting Lemma 60 and the minimality hypothesis on $\{i, j\}$.
Thus we have shown that an Ω-finite $x\omega$-couple for S (and hence T) exists and that an Ω-minimal $x\omega$-couple for S must be an $fx\omega$-couple. We now get a contradiction by looking at T. Suppose that $back_T(p, f, x)$ holds. Let $I' = \{i', j'\}$ be an Ω-minimal $x\omega$-couple for T. Write

$$head_T(I') = \mu' <p = \mathsf{T}> \mu'',$$

where $\mu'' \in pre(body_S(p))$. Clearly μ'' does not pass through f. By Lemma 60 we can obtain another $x\omega$-couple for T by replacing μ' by a prefix that does not enter the body of p in T. By Theorem 126, such a prefix does not pass through any element of $\{f\} \cup Q$; hence this replacement does not increase the sum $\sum_{q \in Q} grade(I', q, \neg(X(q)))$ and the new $x\omega$-couple is not an $fx\omega$-couple, giving a contradiction. $\square$

13 u-equivalence implies Condition (11) of $S\, simil_u\, T$

In order to prove that u-equivalence of LFL schemas S and T implies Condition (11) of u-similarity, we proceed as follows; assuming that $S \cong_u T$ but S and T do not satisfy this condition, then (with symbols as given in Definition 38) we prove the existence of a word $F \in x(\mathcal{F} \cup \mathcal{P})^* \cap \mathcal{F}^*\mathcal{P}$ such that there is an Fu-couple I for S (and hence T, by Proposition 57) such that the prefix of $head_S(I)$ preceding x passes through f and then through f', and does not subsequently pass through a function symbol assigning to $assign_S(f) = assign_S(f')$, and $head_T(I)$ satisfies the same condition with S and T, and f and f', interchanged. This gives a contradiction since I is then an $f'Fu$-couple for S and an fFu-couple for T.

It is worth showing that Condition (11) is not redundant. To see this, for each $j \in \{1, 2\}$ let S_j be the LFL schema

$while\ p_j(v_j)\ do$

$\{$

$u := f_j(v_j);$

$v_j := g_j(v_j);$

$\}$

Then the LFL schemas S_1S_2 and S_2S_1 satisfy every condition of u-similarity *except* Condition (11), but are not u-equivalent; consider for example the interpretation i which maps $p_1(v_1)$ and $p_2(v_2)$ to T and maps every other predicate term to F; clearly $\mathcal{M}[\![S_1S_2]\!]^i_e(u) = f_2(v_2)$ whereas $\mathcal{M}[\![S_2S_1]\!]^i_e(u) = f_1(v_1)$.

Definition 128 (unitary and superunitary symbols) Let S be a linear schema and let σ be a segment in S. Let x be a symbol in S. Then we say that x is unitary in σ (with respect to S) if either x occurs not more than once in σ, or $x \in whilePreds(S)$ and $<x = \mathsf{T}>$ occurs not more than once in σ. We say that x is superunitary in σ (with respect to S) if x is unitary and every while predicate containing x in its body in S is unitary in σ. The segment σ is said to be unitary with respect to S if every symbol in S is unitary in σ with respect to S.

Observe that if $p \in whilePreds(S)$ is superunitary in a prefix σ in S, and p occurs more than once in σ, then $\sigma = \sigma' <p = \mathsf{T}> \sigma'' <p = \mathsf{F}> \sigma'''$ for segments $\sigma', \sigma'', \sigma'''$ not passing through p.

Definition 129 (The set $Predpairs_S(g, x)$) Let S be a linear schema and assume that $g \rightsquigarrow_S x$ for $g \in \mathcal{F} \cup \mathcal{V}$ and $x \in \mathit{Symbols}(S) \cup \mathcal{V}$. We define the set

$$\mathit{Predpairs}_S(g, x) \subseteq \mathit{ifPreds}(S) \times \{\mathsf{T}, \mathsf{F}\}$$

to contain (q, Z) if and only if $\mathit{outif}_S(q, Z, g, x) \vee \mathit{thru}_S(q, Z, g, x)$ holds.

Observe that if LFL schemas S and T satisfy $S \cong_u T$ and $x \in \mathcal{N}_S(u) \cup (\{u\} \cap \mathcal{V})$, then $\mathit{Predpairs}_S(g, x) = \mathit{Predpairs}_T(g, x)$ holds for all $g \in \mathcal{F}$.

Lemma 130 *Let S be a linear schema and let $p \in \mathit{whilePreds}(S)$ and $y \in \mathit{Symbols}(\mathit{body}_S(p))$.*

(1) *Let T be a linear schema, let $u \in \mathcal{V} \cup \{\omega\}$ and assume that S and T satisfy conditions (1)–(10) of u-similarity. Assume $y \in \mathcal{N}_S(u)$. Let $n \leq \mathit{arity}(y)$. If v and w are the nth variables referenced by y in S and T respectively, then $v \rightsquigarrow_{\mathit{body}_S(p)} y \iff w \rightsquigarrow_{\mathit{body}_T(p)} y$ holds.*

(2) *Assume that $\mathit{back}_S(p, g, y)$ holds for some $g \in \mathcal{F}$. Let $v = \mathit{assign}_S(g)$. Then $g \rightsquigarrow_{\mathit{body}_S(p)} v$ and $v \rightsquigarrow_{\mathit{body}_S(p)} y$ hold. Also, if $p \searrow_S q \searrow_S g\,(Z)$ holds for some $q \in \mathit{ifPreds}(S)$, then $\mathit{outif}_S(q, Z, g, x)$ holds.*

(3) *Let $g \in \mathcal{F}$ with $v = \mathit{assign}_S(g)$ and let $q \in \mathit{ifPreds}(S)$; then if $\mathit{back}_S(p, g, y) \wedge \mathit{outif}_S(q, Z, g, y)$ holds then $\mathit{outif}_{\mathit{body}_S(p)}(q, Z, g, v)$ holds, and if*

$$\mathit{back}_S(p, g, y) \wedge \mathit{thru}_S(q, Z, g, y)$$

holds then either $\mathit{thru}_{\mathit{body}_S(p)}(q, Z, g, v)$ or $\mathit{thru}_{\mathit{body}_S(p)}(q, Z, v, y)$ holds, but not both.

Proof.

(1) This follows from Condition (4) of u-similarity and the fact that $v \rightsquigarrow_{\mathit{body}_S(p)} y$ holds if and only if there exists $f \in (\mathcal{F} - \mathit{Funcs}(\mathit{body}_S(p)) \cup \mathcal{V}$ such that $f \rightsquigarrow_{\mathit{body}_S(p)} y$, and the corresponding statement in T.

(2,3) These follow immediately from the definitions of the relations.

$\square$

Observe that the converse of the first statement of Part (2) of Lemma 130 does not hold; that is, $g \rightsquigarrow_{\mathit{body}_S(p)} v \wedge v \rightsquigarrow_{\mathit{body}_S(p)} y$ does not imply $\mathit{back}_S(p, g, y)$; for example, if S is the linear schema in Figure 7.

Lemma 131 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. Let $p \in \mathit{whilePreds}(S)$. Suppose $\mathit{outwhile}_S(p, g, x) \vee \mathit{back}_S(p, g, x)$ holds for some $x \in \mathcal{N}_S(u) \cup (\{u\} \cap \mathcal{V})$. Then*

$$\mathit{Predpairs}_{\mathit{body}_S(p)}(g, \mathit{assign}_S(g)) = \mathit{Predpairs}_{\mathit{body}_T(p)}(g, \mathit{assign}_T(g))$$

holds.

$$
\begin{array}{l}
\textit{while}\, p(u)\, \textit{do} \\
\qquad \{ \\
\qquad \textit{if}\, q(u) \quad \textit{then}\;\; v := g(); \\
\qquad \qquad\qquad\; \textit{else}\;\; \Lambda \\
\qquad w := y(v); \\
\qquad \}
\end{array}
$$

Fig. 7. If $w \neq v$ then $g \rightsquigarrow_{body_S(p)} v \wedge v \rightsquigarrow_{body_S(p)} y \wedge \neg back_S(p, g, y)$ holds here

Proof. If $outwhile_S(p, g, x)$ holds then

$$Predpairs_{body_S(p)}(g, assign_S(g)) = Predpairs_S(g, x) \cap ifPreds(body_S(p))$$

holds; and if $back_S(p, g, x)$ holds then

$$Predpairs_{body_S(p)}(g, assign_S(g)) = Predpairs_S(g, x) - Predpairs_{body_S(p)}(assign_S(g), x)$$

follows from Parts (2) and (3) of Lemma 130. By Conditions (8) and (9) of $S\, simil_u\, T$, the right-hand-sides in each case are unchanged by replacing S by T. $\square$

Lemma 132 *Let S be a linear schema and assume that $f, f' \rightsquigarrow_S x$ for some $f, f' \in \mathcal{F}$ and $x \in Symbols(S) \cup \mathcal{V}$ and assume that $assign_S(f) = assign_S(f')$. Suppose that $f \ll_S f' \ll_S x$ holds. Assume that for every $q \in ifPreds(S)$, the function symbols f and f' are not q-competing for x. Then there is a unitary $f'x$-prefix ν in S such that for all $(q, Z) \in Predpairs_S(f, x) \cup Predpairs_S(f', x)$ the prefix ν does not pass through $<q = \neg Z>$, and for all $q \in whilePreds(S)$ such that $q \searrow_S f \vee q \searrow_S f'$, either ν passes once through $<q = \mathsf{T}>$ or ν does not pass through q.*

Proof. We may assume that $x \in Symbols(S)$, since if instead $x \in \mathcal{V}$ then we can replace S by a linear schema $T = S\, y := h(x)$ and replace x by $h \in Symbols(T)$.
Since $f' \rightsquigarrow_S x$ holds, there is a unitary $f'x$-prefix μ in S. Clearly any unitary $f'x$-prefix satisfies the conditions given for $Predpairs_S(f', x)$ and for any $q \in whilePreds(S)$ such that $q \searrow_S f'$, since it passes through f', hence it suffices to find a unitary $f'x$-prefix satisfying the conditions given for those predicates q for which $q \searrow_S f$.
We say that any $q \in Preds(S)$ is bad for μ if either $(q, Z) \in Predpairs_S(f, x)$ for some $Z \in \{\mathsf{T}, \mathsf{F}\}$ or $q \in whilePreds(S) \wedge q \searrow_S f$, and q does not satisfy the appropriate condition.
We prove the result on induction on the number of predicates q which are bad for μ. Let q be such a predicate. There are two cases to consider.

- Let $q \in whilePreds(S)$ and suppose $q \searrow_S f$. Since $f \ll_S f'$ holds, $\neg(q \searrow_S f')$ and so $q \ll_S f'$ follows. Thus μ passes through q but (since q is bad for μ) does not enter the body of q, and so the letter $<q = \mathsf{F}>$ occurs in μ before f'. We can replace

$<q = \mathsf{F}>$ in μ by $<q = \mathsf{T}>\ \sigma\ <q = \mathsf{F}>$ for some unitary $\sigma \in \Pi(\mathit{body}_S(q))$, chosen such that it does not pass the 'wrong way' through any predicates in $\mathit{body}_S(q)$. Let ρ be the resulting $f'x$-prefix. Clearly q is good (that is, not bad) for ρ and by the choice of σ, every $q' \in \mathit{Preds}(S)$ which is good for μ is also good for ρ and so the result follows from the inductive hypothesis applied to ρ.

- Now suppose that $(q, Z) \in \mathit{Predpairs}_S(f, x)$ for some $Z \in \{\mathsf{T}, \mathsf{F}\}$, and so μ passes through $<q = \neg Z>$. Thus $\neg(q \searrow_S x)$ holds since either $\mathit{thru}_S(q, Z, f, x)$ or $\mathit{outif}_S(q, Z, f, x) \wedge f \ll_S x$ holds. Thus $\neg(q \searrow_S f'(\neg Z))$ holds, since otherwise $\mathit{outif}_S(q, \neg Z, f', x)$ would hold, contradicting the fact that f and f' are not q-competing for x. Hence μ has a segment $<q = \neg Z>\ \sigma$ for some $\sigma \in \Pi(\mathit{part}_S^{\neg Z}(q)$, and f' and x do not occur in σ. We will show that we can replace this segment in μ by $<q = Z>\ \sigma'$ for some unitary $\sigma' \in \Pi(\mathit{part}_S^{Z}(q))$ chosen so as to avoid introducing bad predicates in $\mathit{part}_S^{Z}(q)$, thus getting an $f'x$-prefix ν to which the inductive hypothesis can be applied. If σ occurs before f' in μ, or σ occurs after x in μ, then this is obvious. If on the other hand f' occurs before q and x occurs after σ, then since $f \ll_S f'$ holds, $\neg\mathit{outif}_S(q, Z, f, x)$ follows and so $\mathit{thru}_S(q, Z, f, x)$ holds. Thus by Proposition 31, an $f'x$-segment cannot enter and leave $\mathit{part}_S^{\neg Z}(q)$, thus giving a contradiction. $\square$

Lemma 133 *Let S be a linear schema and let $x \in \mathit{Symbols}(S)$.*

(1) *Let $\mu \in \Pi(S)$ not pass through x. Then there is a predicate q and $Z \in \{\mathsf{T}, \mathsf{F}\}$ such that $q \searrow_S x\,(Z)$ and μ passes through $<q = \neg Z>$ at least once, but μ never passes through $<q = Z>$.*

(2) *Let $y \in \mathit{Symbols}(S)$ satisfy $x \ll_S y$. Suppose $\mu \in \mathit{pre}(\Pi(S))$ ends in y and does not pass through x. Then there is a predicate q and $Z \in \{\mathsf{T}, \mathsf{F}\}$ satisfying the same conditions as in Part (1) of this Lemma.*

Proof.

(1) By induction on $|S|$, there exists $q \in \mathit{Preds}(S)$ such that $q \searrow_S x$ and the path μ passes through q. Choose $q \in \mathit{Preds}(S)$ of 'maximal depth in S' such that both these conditions hold; thus $q \searrow_S x$ and μ passes through q, but for every $q' \in \mathit{Preds}(S)$ satisfying $q \searrow_S q' \searrow_S x$, the path μ does not pass through q'. Suppose that $q \searrow_S x\,(Z)$ holds. If μ does not pass through $<q = Z>$, then we are done. If on the other hand μ passes through $<q = Z>$, then since μ does not pass through x, there exists $q' \in \mathit{Preds}(S)$ satisfying $q \searrow_S q' \searrow_S x$ and μ passes through q', giving a contradiction.

(2) Let $\mu\mu' \in \Pi(S)$. By Part (1) of this Lemma applied to $\mu\mu'$, there exists $q \in \mathit{Preds}(S)$ and $Z \in \{\mathsf{T}, \mathsf{F}\}$ satisfying the required conditions provided that μ' does not pass through q. But clearly either $q \ll_S x$ or $\mathit{above}_S(x) = \mathit{above}_S(q)$ holds, and so $q \ll_S y$ holds, so μ' does not pass through q, giving the result. $\square$

Lemma 134 *Let S be a linear schema, let $q \in \mathit{whilePreds}(S)$ and assume $f \ll_{\mathit{body}_S(q)}$*

x for $f \in Funcs(body_S(q))$ and $x \in Symbols(body_S(q))$. Then

$$\neg(back_S(q, f, x))$$

holds.

Proof. Let $v = assign_S(f)$. Assume that $back_S(q, f, x)$ holds. Thus there is a vx-segment σ in $body_S(q)$, which clearly does not pass through f. Since $f \ll_{body_S(q)} x$ holds, by Lemma 133, Part (2), there is a predicate p in $body_S(q)$ such that $p \searrow_{body_S(q)} f\,(X)$ for $X \in \{\mathsf{T}, \mathsf{F}\}$ and σ passes through $<p = \neg X>$ but not through $<p = X>$. Thus $\neg(p \searrow_{body_S(q)} x\,(X))$ holds; and since $f \rightsquigarrow_S x$ holds, there is an fv-segment τ in the schema $S(p)$. Write $\sigma = \sigma'\sigma''$, such that the last letter of σ' is the last letter of σ lying in $S(p)$. Thus $\tau\sigma''$ is a segment in $S(q)$ starting at f and ending at x. Since $\neg(f \rightsquigarrow_{body_S(p)} x)$ holds, $\tau\sigma''$ passes through an assignment to v after f; and since σ is a vx-segment, no such assignment exists in σ''. Thus the assignment to v must be in τ, contradicting the choice of τ. □

Lemma 135 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. Let $f, f' \in Funcs(S)$ and assume that $f, f' \rightsquigarrow_S x\,(n)$ for some $x \in \mathcal{N}_S(u)$ and $n \leq arity(x)$. Let $\bar{S}$, $\bar{T}$ be the main subschemas lying immediately above $\{f, f'\}$ in S and T respectively and assume that $x \in Symbols(\bar{S})$ and $f \ll_{\bar{S}} f' \wedge f' \ll_{\bar{T}} f$ hold. Then the following hold.*

(1) *$above_{\bar{S}}(x) \notin \{above_{\bar{S}}(f), above_{\bar{S}}(f')\}$.*
(2) *If $\bar{S} = body_S(q)$ for $q \in whilePreds(S)$ then $back_S(q, f, x) \iff back_S(q, f', x)$ holds.*

Proof.

(1) Assume first that $above_{\bar{S}}(x) = above_{\bar{S}}(f)$. Thus $x \ll_{\bar{S}} f'$ holds, and so $\neg(f' < <_{\bar{S}} x)$ holds. Since $f' \rightsquigarrow_S x$ holds, we must have $\bar{S} = body_S(q)$ for some $q \in whilePreds(S)$ such that $back_S(q, f', x)$ and hence $back_T(q, f', x)$ holds by Condition (9) of $S\,simil_u\,T$. But by Theorem 126 this contradicts $above_{\bar{S}}(x) = above_{\bar{S}}(f)$.
If instead $above_{\bar{S}}(x) = above_{\bar{S}}(f')$, then interchange (S, f) and (T, f') to get a contradiction.
(2) By Part (1) of this Lemma and its analogue in T, Theorem 126 and Lemma 134, $g \rightsquigarrow_K x \iff g \ll_K x$ holds for each $g \in \{f, f'\}$ and $K \in \{\bar{S}, \bar{T}\}$. Thus since $\ll_{\bar{S}}$ is transitive, $f' \rightsquigarrow_{\bar{S}} x \Rightarrow f \rightsquigarrow_{\bar{S}} x$ and similarly $f \rightsquigarrow_{\bar{T}} x \Rightarrow f' \rightsquigarrow_{\bar{T}} x$ follow from $f \ll_{\bar{S}} f'$ and $f' \ll_{\bar{T}} f$ respectively. Thus $back_S(q, f, x) \Rightarrow back_S(q, f', x)$ and $back_T(q, f', x) \Rightarrow back_T(q, f, x)$. Hence Part (2) of this Lemma follows from Condition (9) of u-similarity.

□

Lemma 136 (syntactic consequence of $\ll_S$ relation) *Let S be an LFL schema and let $f, f' \in Funcs(S)$ and assume that $f, f' \rightsquigarrow_S x(n)$ for some $x \in Symbols(S)$ and $n \leq arity(x)$. Let $\bar{S}$ be the main subschema lying immediately above $\{f, f'\}$ in S and assume that $f \ll_{\bar{S}} f'$ and if $x \in Symbols(\bar{S})$ then*

$$above_{\bar{S}}(x) \notin \{above_{\bar{S}}(f), above_{\bar{S}}(f')\}$$

and $f \ll_{\bar{S}} x \iff f' \ll_{\bar{S}} x$ hold. Assume that for every $q \in ifPreds(\bar{S})$, the function symbols f and f' are not q-competing for x. Also let H be the set of while predicates q in $\bar{S}$ satisfying $q \searrow_{\bar{S}} f \vee q \searrow_{\bar{S}} f'$. Let $g \in \{f, f'\}$. Then the following conditions hold if and only if $g = f'$.

(1) *If $f' \rightsquigarrow_{\bar{S}} x$ then there exists a gx-prefix $\mu \in pre(\Pi(\bar{S}))$ such that for all $(q, Z) \in Predpairs_{\bar{S}}(f, x) \cup Predpairs_{\bar{S}}(f', x)$ we have $grade(\mu, q, \neg Z) = 0$ and for all $q \in H$, μ passes through $<q = \mathsf{T}>$ if it passes through q.*
(2) *If $\neg(f' \rightsquigarrow_{\bar{S}} x)$ then there exists a $g\,assign_S(g)$-prefix $\mu \in \Pi(\bar{S})$ such that for all $(q, Z) \in Predpairs_{\bar{S}}(f, assign_S(f)) \cup Predpairs_{\bar{S}}(f', assign_S(f'))$ we have $grade(\mu, q, \neg Z) = 0$ and for all $q \in H$, μ passes through $<q = \mathsf{T}>$ if it passes through q.*

Furthermore, if $g = f'$ then μ may be chosen to be unitary.

Proof. First observe that if $f' \rightsquigarrow_{\bar{S}} x$, then $f \rightsquigarrow_{\bar{S}} x$ holds by Lemma 134; and if $\neg(f' \rightsquigarrow_{\bar{S}} x)$ instead, then either $x \notin Symbols(\bar{S})$ or $\bar{S} = body_S(r)$ for $r \in whilePreds(S)$ and $back_S(r, h, x)$ for each $h \in \{f, f'\}$. Hence $h \rightsquigarrow_{\bar{S}} assign_S(f) = assign_S(f')$ for each $h \in \{f, f'\}$. We consider the cases $g = f'$ and $g = f$ separately.

- Assume $g = f'$. The results in both cases (1) and (2) follow from Lemma 132 applied to $\bar{S}$.
- Now assume $g = f$.
  We first prove the result for Case (1); thus we assume that $f' \rightsquigarrow_{\bar{S}} x$ holds. Suppose that there exists an fx-prefix μ in $\bar{S}$ satisfying the conditions given. We will deduce a contradiction. Thus since $f \ll_{\bar{S}} f'$ holds, we will get a contradiction if we show that μ passes through f'. Suppose that this does not hold; then by Lemma 133, Part (2), there exists $q \in Preds(\bar{S})$ such that $q \searrow_{\bar{S}} f'(Z)$ holds and μ passes through $<q = \neg Z>$, but not through $<q = Z>$. If $q \in whilePreds(\bar{S})$ then $q \in H$, contradicting immediately the conditions on μ; and if $q \in ifPreds(\bar{S})$ then clearly $\neg(q \searrow_{\bar{S}} x(Z))$ holds, so $(q, Z) \in Predpairs_{\bar{S}}(f', x)$, again contradicting immediately the conditions on μ.
  The result for Case (2) is similar except that Part (1) of Lemma 133 must be used.

□

Theorem 137 (u-equivalence implies condition (11) of u-similarity) *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. If $f, f' \in Funcs(S)$ and*

$f, f' \rightsquigarrow_S x$ for $x \in \mathcal{N}_S(u) \cup (\{u\} \cap \mathcal{V})$ and $assign_S(f) = assign_S(f')$, and $\bar{S}$, $\bar{T}$ are the main subschemas of S and T respectively lying immediately above $\{f, f'\}$, then either $\neg(f \ll_{\bar{S}} f' \wedge f' \ll_{\bar{T}} f)$ holds, or there exists $q \in ifPreds(S)$ such that f and f' are q-competing for x in S.

Proof. Assume that there is no $q \in ifPreds(S)$ such that f and f' are q-competing for x in S, and that $f \ll_{\bar{S}} f' \wedge f' \ll_{\bar{T}} f$ holds. We will obtain a contradiction.
We may also assume, using Lemma 27, that $x \in \mathcal{N}_S(u)$ and there is a word $F \in x(\mathcal{F} \cup \mathcal{P})^* \cap \mathcal{F}^*\mathcal{P}$ such that there is an Fu-couple for S.
If $S \neq \bar{S}$, we define $\bar{S} = body_S(r)$. We now consider two cases separately.

- Suppose that $f' \rightsquigarrow_{\bar{S}} x$ ($\iff$ $f' \rightsquigarrow_{\bar{T}} x$) holds. Let $\mathcal{I}$ be the set of all fFu- and $f'Fu$-couples for S which are (q, T)-finite for all $q \in whilePreds(S) \cap \mathcal{N}_S(u)$ if $u \in \mathcal{V}$ and let H be the set of while predicates q in $\bar{S}$ satisfying $q \searrow_{\bar{S}} f \vee q \searrow_{\bar{S}} f'$. Define the set
$$\Omega = Predpairs_{\bar{S}}(f, x) \cup Predpairs_{\bar{S}}(f', x).$$
Since there is no $q \in ifPreds(S)$ such that f and f' are q-competing for x in S, the set Ω is reasonable. By Corollary 116, the set $\mathcal{I}$ contains an Fu-couple which is (r, T)-finite if $H \neq \emptyset \wedge \bar{S} = body_S(r)$, is $(q, \neg Z)$-finite for every $(q, Z) \in \Omega$, and is (q, F)-finite for every $q \in H$. Let $\mathcal{J} \subseteq \mathcal{I}$ be the set of all such couples. Let $\mathcal{J}'$ be the set of couples $I \in \mathcal{J}$ for which
$$\sum_{(q,Z)\in\Omega} grade(I, q, \neg Z)$$
is minimal, let $\mathcal{J}''$ be the set of all $I \in \mathcal{J}'$ for which $grade(I, r, \mathsf{T})$ is minimal if $H \neq \emptyset \wedge S \neq \bar{S} = body_S(r)$ (else let $\mathcal{J}'' = \mathcal{J}'$), and let $\mathcal{J}'''$ be the set of all $I \in \mathcal{J}''$ for which the sum
$$\sum_{q\in H} grade_S(I, q, \mathsf{F}, EXT)$$
is minimal.
Let $I \in \mathcal{J}'''$. We will assume that I is an fFu-couple and show that this leads to a contradiction. If we had assumed that I is an $f'Fu$-couple, we could similarly have obtained a contradiction by interchanging S and T and f and f', since by Proposition 57, the sets Ω and H are unchanged if S is replaced by T.
Assume, thus, that I is an fFu-couple. Observe that by Lemma 135, the hypotheses of Lemma 136 are satisfied. To simplify the notation we will assume that $x \in \mathcal{F}$. Thus $head_S(I)$ has a segment μ which is an fx-prefix in $\bar{S}$, and such that if μ in $head_S(I)$ is replaced by an $f'x$-prefix ν in $\bar{S}$, then an $f'Fu$-couple $J \in \mathcal{I}$ results.
By Lemma 136, Part (1), either μ passes through a letter $<q = \neg Z>$ such that $(q, Z) \in \Omega$ or μ passes through a letter $<q = \mathsf{F}>$ such that $q \in H$ and μ does not pass through $<q = \mathsf{T}>$, and by this Lemma, we may choose ν such that it is unitary and does not satisfy these conditions.
Assume that J is $(S, q, \neg Z)$-normalised for every $(q, Z) \in \Omega$,
(S, r, T)-normalised if $\bar{S} \neq S$ and (S, q, F)-normalised for every $q \in H$. Thus $J \in \mathcal{J}$; and so $J \in \mathcal{J}''$ by Lemma 69.

If μ passes through a letter $\lessdot q = \neg Z \gtrdot$ such that $(q, Z) \in \Omega$, then Lemma 69 applied to I and J contradicts $I \in \mathcal{J}'$. Hence there is some $q \in H$ violating the conclusion of Part (1) of Lemma 136 for μ. Let ρ be the prefix of $\mathit{head}_S(I)$ preceding μ. If $S \neq \bar{S}$ and ρ passes through $\bar{S}$, then Lemma 69 contradicts $I \in \mathcal{J}''$, since ρ could be replaced by a prefix not passing through $\bar{S}$. Thus ρ does not enter the body of q and so $\mathit{grade}_S(\rho\mu, q, \mathsf{F}, EXT) > 0$, and so Part (3) of Lemma 114 applied to I and J contradicts $I \in \mathcal{J}'''$.

- Suppose that $\neg(f' \rightsquigarrow_{\bar{S}} x)$ holds. Thus either $x \notin \mathit{Symbols}(\bar{S})$ or $\bar{S} = \mathit{body}_S(r) \wedge \mathit{back}_S(r, h, x)$ holds for each $h \in \{f, f'\}$; and the same statement with S replaced by T is true. The proof here is the same as in the preceding case except that we must define $\Omega = \mathit{Predpairs}_{\bar{S}}(f, \mathit{assign}_S(f)) \cup \mathit{Predpairs}_{\bar{S}}(f', \mathit{assign}_S(f'))$ (by Lemma 131, this definition is unchanged if S is replaced by T) and the segment μ is an f $\mathit{assign}_S(f)$-prefix in $\bar{S}$, and Part (2) of Lemma 136 must be used.

□

Theorem 147 is the main result of this section.

As we have mentioned in Section 3, Condition (12) of u-similarity is probably redundant; that is, we conjecture that it is a consequence of the other u-similarity conditions. However Condition (13) is not. To see this, let S_1 be the LFL schema

$$
\begin{array}{lll}
\textit{if } \hat{q}(z) & \textit{then} & v := f(z,w); \\
\textit{if } q(z) & \textit{then} & u := g(z,v); \\
z := h(z); & &
\end{array}
$$

and let S_2 be the LFL schema

$$
\begin{array}{lll}
\textit{if } q(z) & \textit{then} & \\
& & \{ \\
& & \quad \textit{if } \hat{q}(z) \quad \textit{then } v := f(z,w); \\
& & \quad u := g(z,v); \\
& & \} \\
z := h(z); & &
\end{array}
$$

For each $j \in \{1,2\}$ let T_j be the LFL schema $\textit{while}\, p(z)\, \textit{do}\, S_j$. Then the schemas T_1 and T_2 satisfy every condition of u-similarity except Condition (13), since

$$\textit{outif}_{S_2}(q, \mathsf{T}, f, v)$$

and

$$\neg(\textit{outif}_{S_1}(q, \mathsf{T}, f, v) \vee \textit{thru}_{S_1}(q, \mathsf{T}, f, v))$$

hold, and are not u-equivalent; consider for example the interpretation i which satisfies

$$p^i(z) = p^i(h(z)) = \hat{q}^i(z) = q^i(h(z)) = \mathsf{T}$$

and maps all other predicate terms to F. For both T_1 and T_2, the path $\pi_S(i,e)$ passes twice through the body of p; passing through the false part of q the first time and through the true part of q the second time. For $\hat{q}$ it is the other way round. Thus $\mathcal{M}[\![T_1]\!]^i_e(u) = g(h(z), f(z,w))$ whereas $\mathcal{M}[\![T_2]\!]^i_e(u) = g(h(z), v)$.

Definition 138 (The u-correspondence relation for prefixes) Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be linear schemas satisfying conditions (1)-(11) of u-similarity. Let $\sigma \in \mathit{pre}(\Pi(S))$ and $\tau \in \mathit{pre}(\Pi(T))$. We define the *correspondence* relation as follows. Assume that σ and τ end in the same symbol $x \in \mathcal{N}_S(u)$, one element of the set $\{\sigma, \tau\}$ is unitary and x is superunitary in the other, and if $x \in \mathit{Preds}(S)$, then σ and τ end in the same actual letter (that is, either $<\!x = \mathsf{T}\!>$ or $<\!x = \mathsf{F}\!>$). Also assume that if $x \in \mathit{whilePreds}(S)$, then x occurs the same number of times in each element of $\{\sigma, \tau\}$. Then we say that (σ, τ) is a u-corresponding pair (for (S, T)).
We also say that a pair (σ, τ) is a *disagreeing* u-correspondence (for (S, T)) if one element of the set $\{\sigma, \tau\}$ ends in $<\!p = Z\!>$ for $p \in \mathit{Preds}(S) \cap \mathcal{N}_S(u)$ and $Z \in \{\mathsf{T}, \mathsf{F}\}$, the other element ends in $<\!p = \neg Z\!>$, and replacing Z and $\neg Z$ by F in the last letters of σ and τ gives a u-corresponding pair for (S, T).

Lemma 139 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be linear schemas satisfying conditions (1)-(11) of u-similarity. Let (σ, τ) be a u-corresponding pair for (S, T). Assume there do not exist $\sigma' \in \mathit{pre}(\sigma)$ and $\tau' \in \mathit{pre}(\tau)$ such that (σ', τ') is a disagreeing u-correspondence for (S, T).*

(1) *Let $y \in \mathcal{N}_S(u)$ occur in both σ and τ. Then y is superunitary in both σ and τ.*
(2) *Let $x \in \mathcal{N}_S(u)$ be the last letter of σ and τ. Then*

$$\mathcal{M}[\![\mathit{schema}(\sigma)]\!]_e(x(\mathbf{refvec}_S(x))) = \mathcal{M}[\![\mathit{schema}(\tau)]\!]_e(x(\mathbf{refvec}_T(x)))$$

holds.

Proof.

(1) By the symmetry of the hypotheses, we may assume that y is superunitary in σ, but not in τ. We will deduce a contradiction. Let the sequence $y_1, \ldots, y_n = y$ be such that each y_i for $i < n$ is a while predicate and each y_i lies immediately below y_{i-1} in S and y_1 lies immediately below S. Clearly $n \geq 2$. Suppose that y_i is not superunitary in τ, for minimal i. Thus y_i is not unitary in τ, and $y_i \in \mathit{whilePreds}(T)$ holds by the minimality of i. Let $\hat{\tau}$ be the prefix of τ ending in the second occurrence of y_i. We may write

$$\hat{\tau} = \tau' <\!y_i = \mathsf{T}\!> \tau'' <\!y_i = \mathsf{T}\!>$$

by the minimality of i. Since y and hence y_i occurs in σ, we may write

$$\sigma = \sigma' <\!y_i = Z\!> \bar{\sigma}$$

such that y_i does not occur in σ'. Since $(\sigma' <\!y_i = Z\!>, \tau' <\!y_i = \mathsf{T}\!>)$ is not allowed to be a disagreeing u-correspondence, we have $Z = \mathsf{T}$; and since y and hence y_i is superunitary in σ, the letter $<\!y_i = \mathsf{T}\!>$ does not occur in $\bar{\sigma}$. Clearly $(x \neq y_i) \wedge \neg(y_i \searrow_T x)$ holds by the hypotheses, and σ ends in x, hence $<\!y_i = \mathsf{F}\!>$ occurs in $\bar{\sigma}$. But this implies that σ has a prefix $\hat{\sigma}$ such that $(\hat{\sigma}, \hat{\tau})$ is a disagreeing u-correspondence, contradicting the hypotheses.

(2) This follows by induction on the length of σ. For any $r \leq \mathit{arity}(x)$, let $F(S,r)$ be the set of elements $f \in \mathcal{F}$ such that $f \rightsquigarrow_S x\,(r)$ and σ passes through f and $f \neq x$, and define $F(T,r)$ similarly with respect to τ. We will show that $F(S,r) = F(T,r)$, and that the last element of $F(S,r)$ to occur along ξ is the same for each $\xi \in \{\sigma, \tau\}$.

Suppose that $f \in F(S,r)$. We will show that $f \in F(T,r)$; the opposite implication follows similarly. Let $\hat{T}$ be the main subschema of T lying immediately above $\{f,x\}$. Assume first that $\neg(f \ll_{\hat{T}} x)$ holds. Thus $\hat{T} = \mathit{body}_T(p)$ and $\mathit{back}_T(p,f,x)$ holds for some $p \in \mathit{whilePreds}(T)$. Hence $\mathit{back}_S(p,f,x)$ and so $\neg(f \ll_{\mathit{body}_S(p)} x)$ by Lemma 134, contradicting the fact that x is superunitary in σ. Hence $f \ll_{\hat{T}} x$ holds; and so if $f \notin F(T,r)$, then by Part (2) of Lemma 133 applied to $\hat{T}$, we get $q \searrow_{\hat{T}} f\;(X)$ for some predicate q and $X \in \{\mathsf{T}, \mathsf{F}\}$, and τ passes through $<q = \neg X>$ but not through $<q = X>$. Thus $\neg(q \searrow_T x\;(X))$ holds; hence $\mathit{outif}_T(q,X,f,x)$ or $\mathit{outwhile}_T(q,f,x)$ holds. Thus

$$\mathit{outif}_S(q,X,f,x) \vee \mathit{thru}_S(q,X,f,x) \vee \mathit{outwhile}_S(q,f,x)$$

holds by conditions (8) and (10) of u-similarity and so σ passes through $<q = X>$ and hence q is superunitary in both σ and τ by Part (1) of this Lemma. Let $\sigma' \in \mathit{pre}(\sigma)$ end at $<q = X>$ and define $\tau' \in \mathit{pre}(\tau)$ similarly; then the pair (σ', τ') is a disagreeing u-correspondence, contradicting the hypotheses. Thus we conclude that $F(S,r) = F(T,r)$ and hence all elements of $F(S,r)$ are superunitary in both σ and τ, by Part (1) of this Lemma.

Now assume that f and g are the last elements of $F(S,r) = F(T,r)$ to occur on σ and τ respectively. Suppose $f \neq g$. We will show that this leads to a contradiction. Thus f occurs before g in σ, but after g in τ. Let $\bar{S}$ and $\bar{T}$ be the main subschemas of S and T lying immediately above $\{f,g\}$. Since f and g are superunitary in both σ and τ, we get $f \ll_{\bar{S}} g$ and $g \ll_{\bar{T}} f$; thus by Condition (11) of u-similarity there is an if predicate q such that f and g are q-competing for x. Thus there exists $X \in \{\mathsf{T}, \mathsf{F}\}$ such that σ passes through $<q = X>$ and τ passes through $<q = \neg X>$, again giving a disagreeing u-correspondence since q is superunitary in both σ and τ by Part (1) of this Lemma. Thus we have shown $f = g$.

We now show that (2) follows from this. Let $r \leq \mathit{arity}(x)$. We may assume that $F(S,r) \neq \emptyset$. Observe that if $f \in F(S,r)$ and σ has a prefix σ' ending in f and does not later pass through an element of $F(S,r)$, then if we write $\mathcal{M}[\![\mathit{schema}(\sigma)]\!]_e(x(\mathbf{refvec}_S(x))) = x(\mathbf{t})$, then the rth component of $\mathbf{t}$ is

$$\mathcal{M}[\![\mathit{schema}(\sigma')]\!]_e(f(\mathbf{refvec}_S(f))),$$

and a similar statement holds for a prefix τ' of τ. Thus (2) follows from the inductive hypothesis applied to (σ', τ'), which is a u-corresponding pair by Part (1) of this Lemma.

□

Corollary 140 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas and let i be an interpretation. Let $\sigma \in pre(\pi_S(i,e))$ and $\tau \in pre(\pi_T(i,e))$.*

(1) *Then (σ, τ) is not a disagreeing u-correspondence for (S,T).*
(2) *If (σ, τ) is a u-correspondence for (S,T) and σ and τ both end in $x \in \mathcal{N}_S(u)$, then*

$$\mathcal{M}[\![schema(\sigma)]\!]_e(x(\mathbf{refvec}_S(x))) = \mathcal{M}[\![schema(\tau)]\!]_e(x(\mathbf{refvec}_T(x)))$$

holds.

Proof. We prove the conditions by induction on $|\sigma|$. Thus (2) for the pair (σ, τ) follows from the inductive hypothesis and Part (2) of Lemma 139 applied to (σ, τ), so it remains to prove (1) for the pair (σ, τ). We will assume (1) is false and deduce a contradiction.
Let $\sigma = \sigma' \lessdot x = Z(S) \gtrdot$ and $\tau = \tau' \lessdot x = Z(T) \gtrdot$ and let $\bar{\sigma}$ and $\bar{\tau}$ be obtained from σ and τ by replacing $Z(S)$ and $Z(T)$ by F in the last letter. Suppose that the final occurrence of x defines the predicate terms $x(\mathbf{s})$ and $x(\mathbf{t})$ in σ and τ respectively. Let the interpretation $j = i(x(\mathbf{s}) = x(\mathbf{t}) = \mathsf{F})$.
We now show that τ' does not pass through $x(\mathbf{s})$. If this is false, then since clearly $(\bar{\sigma}, \bar{\tau})$ is a u-correspondence for (S,T), we have $x \in \mathit{whilePreds}(S)$ and σ' and τ' have prefixes σ'' and τ'' which end in x (defining the predicate term $x(\mathbf{s})$ in the case of τ'') and u-correspond. By the minimality assumption on σ, applying (2) to the pair (σ'', τ'') shows that σ'' also passes through $x(\mathbf{s})$, contradicting the freeness of S.
Thus we have shown that τ' and (similarly) σ' do not pass through $x(\mathbf{s})$ and $x(\mathbf{t})$ respectively. Hence $\bar{\sigma} \in pre(\pi_S(j,e))$ and $\bar{\tau} \in pre(\pi_T(j,e))$. Thus $x(\mathbf{s}) = x(\mathbf{t})$ follows from the inductive hypothesis and Part (2) of Lemma 139 applied to $(\bar{\sigma}, \bar{\tau})$, thus showing that $Z(S) = Z(T)$ and giving a contradiction. $\square$

Lemma 141 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas and let i be an interpretation.*

(1) *Suppose that $back_S(p, g, y)$ holds for some $g \in \mathcal{F}$ and $p \in \mathit{whilePreds}(S)$ with $above_{body_S(p)}(g) = g$, and $y \in \mathcal{N}_S(u)$. Assume that*

$$\sigma = \sigma' \lessdot p = \mathsf{T} \gtrdot \sigma'' \in pre(\pi_S(i,e)),$$

where $\sigma'' \in \Pi(body_S(p))$, and define $\tau = \tau' \lessdot p = \mathsf{T} \gtrdot \tau'' \in pre(\pi_T(i,e))$ such that $(\sigma' \lessdot p = \mathsf{T} \gtrdot, \tau' \lessdot p = \mathsf{T} \gtrdot)$ is a u-correspondence for (S,T) and $\tau'' \in \Pi(body_T(p))$. If either σ or τ is unitary, and σ'' ends in a g $assign_S(g)$-segment, then τ'' passes through g.
(2) *Assume that $u = \omega$ and let*

$$\sigma = \sigma' \lessdot p = \mathsf{T} \gtrdot \sigma'' \in pre(\pi_S(i,e))$$

and

$$\tau = \tau' <p = \mathsf{T}> \tau'' \in \mathit{pre}(\pi_T(i,e))$$

where $(\sigma' <p = \mathsf{T}>, \tau' <p = \mathsf{T}>)$ *is a* u*-correspondence for* (S,T) *and* $\sigma'' \in \Pi(\mathit{body}_S(p))$ *is unitary. Assume* $\tau'' \in \mathit{pre}(\Pi(\mathit{body}_T(p)))$*. Then* τ'' *is unitary; and if* τ'' *passes through a while predicate, then* σ'' *also passes through this while predicate.*

(3) *Assume that* $u = \omega$ *and* $\pi_S(i,e)$ *has a unitary prefix ending at* $p \in \mathit{whilePreds}(S)$*. If* $\pi_T(i,e)$ *passes through* p*, then* p *is superunitary in the prefix of* $\pi_T(i,e)$ *ending at the first occurrence of* p*.*

Proof.

(1) If τ'' does not pass through g, then $p \searrow_T q \searrow_T g\,(Z)$ for some $q \in \mathit{ifPreds}(T)$ such that $\mathit{above}_{\mathit{body}_S(p)}(q) = q$ and τ'' passes through $<q = \neg Z>$. Thus $\mathit{outif}_T(q,Z,g,y)$ holds by Part (2) of Lemma 130 and so

$$\mathit{outif}_S(q,Z,g,y) \vee \mathit{thru}_S(q,Z,g,y)$$

holds. We will show that σ'' passes through $<q = Z>$, whereupon Part (2) of Corollary 140 gives a contradiction since q is superunitary in the prefixes of σ and τ ending at these occurrences of q. If σ'' does not pass through $<q = Z>$, then

$$\neg \mathit{outif}_S(q,Z,g,y) \wedge \neg \mathit{thru}_{\mathit{body}_S(p)}(q,Z,g,\mathit{assign}_S(g))$$

holds and so $\mathit{thru}_S(q,Z,g,y)$ holds. Thus by Part (3) of Lemma 130,

$$\mathit{thru}_{\mathit{body}_S(p)}(q,Z,\mathit{assign}_S(g),y)$$

holds and so by Part (1) of Lemma 130, $\mathit{thru}_{\mathit{body}_T(p)}(q,Z,\mathit{assign}_T(g),y)$ holds. But we have shown that $\mathit{outif}_T(q,Z,g,y)$ holds. Let $\mu\mu'$ be any $\mathit{assign}_T(g)\,y$-segment in $\mathit{body}_T(p)$, where μ contains all the symbols lying under q. Let μ''' be a $g\,\mathit{assign}_T(g)$-segment in $\mathit{part}^Z_T(q)$; thus $\mu'''\mu''$ is a gy-segment in $\mathit{body}_T(p)$, contradicting $\mathit{back}_T(p,g,y)$.

(2) This is by induction on the length of τ''. Assume, thus, that the assertions are true for every strict prefix of τ''. Suppose first that τ'' ends in $q \in \mathit{whilePreds}(S)$, and σ'' does not pass through q. By the inductive hypothesis, if $p \searrow_S p' \searrow_S q$ for $p' \in \mathit{whilePreds}(S)$, then σ'' passes through p'. Thus by Part (1) of Lemma 133, there is an if predicate p' such that

$$p \searrow_S p' \searrow_S q\,(Z) \wedge (\mathit{above}_{\mathit{part}^Z_S(p')}(q) = q)$$

and σ'' passes through $<p' = \neg Z>$ but not through $<p' = Z>$. Hence also $p' \searrow_S q\,(Z)$ holds by condition (7) of ω-similarity and so the prefixes of $\pi_S(i,e)$ and $\pi_T(i,e)$ ending at p' within σ'' and τ'' are ω-disagreeing, contradicting Part (1) of Corollary 140.

Assume now that τ'' is not unitary. Thus by the inductive hypothesis we may write $\tau'' = \mu <q = \mathsf{T}> \mu' <q = \mathsf{T}>$ such that $\mu <q = \mathsf{T}> \mu'$ is unitary and μ' is a path in the body of the while predicate q. By the inductive hypothesis applied to $\mu \lessdot q = \mathsf{T} \gtrdot$, σ'' passes through q and so σ'' has a prefix $\nu \lessdot q = \mathsf{T} \gtrdot \nu' \lessdot q = \mathsf{F} \gtrdot$, again giving rise to an ω-disagreement.

(3) This is by induction on the number of while predicates containing p in their bodies. Let τ be the shortest element of $\mathit{pre}(\pi_T(i,e))$ ending in p. Clearly p is unitary in τ, so if p is not superunitary in τ, then $q \searrow_S p$ holds for a while predicate q which is not superunitary in τ, such that all while predicates q' satisfying $q' \searrow_S q$ *are* superunitary in τ, and so q is not unitary in τ. Thus τ has a prefix

$$\tau' \lessdot q = \mathsf{T} \gtrdot \tau'' \lessdot q = Z \gtrdot$$

with $Z \in \{\mathsf{T}, \mathsf{F}\}$ and $\tau'' \in \Pi(\mathit{body}_T(q))$ such that q is superunitary in $\tau' \lessdot q = \mathsf{T} \gtrdot$ by the inductive hypothesis. Since p only occurs at the end of τ, τ'' does not pass through p.

By the hypotheses there is a unitary prefix $\sigma' \lessdot q = \mathsf{T} \gtrdot \sigma'' \in \mathit{pre}(\pi_S(i,e))$ such that $\sigma'' \in \mathit{pre}(\Pi(\mathit{body}_S(p))$ ends in p, thus contradicting Part (2) of this Lemma.

□

Lemma 142 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be linear schemas satisfying conditions (1)–(10) of u-similarity. Let $q \in \mathit{ifPreds}(S)$ and $x \in \mathcal{N}_S(u)$. Let $X \in \{\mathsf{T}, \mathsf{F}\}$. If $q \searrow_S x\,(X)$ then $q \searrow_T x\,(\neg X)$ is false.*

Proof. Assume $q \searrow_S x\,(X)$. The Lemma follows by induction on $\mathit{depnum}_S\,(x,u)$. If $\mathit{depnum}_S\,(x,u) \leq 2$ then the Lemma follows from Conditions (7) and (10) of u-similarity.

Thus we may assume $\mathit{depnum}_S\,(x,u) \geq 3$ and so by Proposition 37, either $x \rightsquigarrow_S y$ or $x \searrow_S y$ for $y \in \mathcal{N}_S(u)$ with $\mathit{depnum}_S\,(y,u) < \mathit{depnum}_S\,(x,u)$.

Assume first that $x \rightsquigarrow_S y$ holds. If $\mathit{outif}_S(q,X,x,y)$ then the Lemma follows from Condition (10) of u-similarity; so assume that $x \rightsquigarrow_{\mathit{part}_S^X(q)} y$ holds. Thus $\neg(q \searrow_T y\,(\neg X))$ by the inductive hypothesis applied to y and so the Lemma again follows from Condition (10) of u-similarity.

We now consider the case where $x \searrow_S y\,(X')$ for $X' \in \{\mathsf{T}, \mathsf{F}\}$. By the inductive hypothesis applied to y, $\neg(q \searrow_T y\,(\neg X))$ holds. Thus if $x \in \mathit{whilePreds}(S)$, then $\neg(q \searrow_T x\,(\neg X))$ follows from Condition (8) of u-similarity. Hence we may assume $x \in \mathit{ifPreds}(S)$. By Proposition 37, $y \rightsquigarrow_S z$ for $z \in \mathcal{N}_S(u)$ with $\neg(x \searrow_S z\,(X'))$, and so $\mathit{outif}_S(x,X',y,z)$ holds. By Condition (10) of u-similarity, either $\mathit{outif}_T(x,X',y,z)$ or $\mathit{thru}_T(x,X',y,z)$ holds. The former case would contradict $\neg(q \searrow_T y\,(\neg X))$, so $\mathit{thru}_T(x,X',y,z)$ follows. By Proposition 31, there exists $y' \in \mathcal{F}$ such that

$$\mathit{outif}_T(x, \neg X', y', z)$$

holds and $\mathit{depnum}_S\,(y',u) < \mathit{depnum}_S\,(x,u)$. Assume that $q \searrow_T x\,(\neg X)$ and so $q \searrow_T y'\,(\neg X)$ holds. We will deduce a contradiction. By the inductive hypothesis applied to z, either $\neg(q \searrow_S z\,(X))$ or $\neg(q \searrow_T z\,(\neg X))$ holds. If $\neg(q \searrow_T z\,(\neg X))$, holds, then $outif_T(q, \neg X, y', z)$ follows, and so by Condition (10) of u-similarity, $outif_S(q, \neg X, y', z) \vee thru_S(q, \neg X, y', z)$ and hence $\neg(q \searrow_S y'(X))$ holds. Also

$$outif_S(x, \neg X', y', z) \vee thru_S(x, \neg X', y', z)$$

(and thus clearly the latter) similarly holds. Hence $thru_S(q, \neg X, y', z) \wedge thru_S(q, X, y', z)$ holds. But then every $y'z$-segment in S must pass twice through q, which is clearly impossible. If instead $\neg(q \searrow_S z(X))$ holds, then a contradiction is similarly obtained by interchanging S and T, y and y', X and X' and X and $\neg X$, thus completing the proof. $\square$

Lemma 143 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas, let $p \in whilePreds(S)$ and assume that $v \rightsquigarrow_{body_S(p)} x(r)$ for some*

$$x \in Symbols(body_S(p)) \cap \mathcal{N}_S(u).$$

Then there is a unitary vx-segment σ in $body_S(p)$ which does not enter the body of any while predicate whose body does not contain x, and such that for all $q \in ifPreds(body_T(p))$ such that $q \searrow_T x(Z)$, σ does not pass through $<q = \neg Z>$.

Proof. Clearly a unitary vx-segment σ in $body_S(p)$ exists which satisfies the condition on while predicates. Suppose that there exists
$q \in ifPreds(body_T(p))$ such that $q \searrow_T x(Z)$ and σ passes through $<q = \neg Z>$. By Lemma 142, and the fact that σ is unitary, $\neg(q \searrow_S x)$ holds. We will alter σ by rerouting it through $part_S^Z(q)$ instead of $part_S^{\neg Z}(q)$.
Suppose this does not work; that is, $g \rightsquigarrow_{body_S(p)} x(r)$ for some $g \in \mathcal{F}$ such that $q \searrow_S g(Z)$. Clearly $outif_S(q, Z, g, x) \vee thru_S(q, Z, g, x)$ holds, and the same condition holds in T. But $q \searrow_T x(Z)$ and so $outif_T(q, Z, g, x)$ holds and thus $back_T(p', g, x)$ holds for some $p' \in whilePreds(T)$. Hence $back_S(p', g, x)$ holds, contradicting the fact that σ is unitary. Thus the altered prefix is still a vx-segment which can be chosen to satisfy the other conditions required.
We repeat this step for every 'bad' pair (q, Z), in order of increasing length of the prefix of σ preceding q, to obtain the desired vx-segment in $body_S(p)$. $\square$

Lemma 144 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas, let $p \in whilePreds(S) \cap \mathcal{N}_S(u)$ and let $(\sigma <p = \mathsf{T}>, \tau <p = \mathsf{T}>)$ be a u-corresponding pair for (S, T). Assume that σ and τ end with $<p = \mathsf{T}>$. Let i be an interpretation and assume that $\sigma \in pre(\pi_S(i, e))$ and $\tau \in pre(\pi_T(i, e))$. Let $x \in \mathcal{N}_S(u) \cap Symbols(body_S(p))$ and assume that $v \rightsquigarrow_{body_S(p)} x(r)$ and $w \rightsquigarrow_{body_T(p)} x(r)$ for some $v, w \in \mathcal{V}$ and $r \leq arity(x)$. Then $\mathcal{M}[\![schema(\sigma)]\!]^i_e(v) = \mathcal{M}[\![schema(\tau)]\!]^i_e(w)$.*

Proof. Suppose that there exist prefixes

$$\sigma' \in pre(\Pi(body_S(p))), \quad \tau' \in pre(\Pi(body_T(p)))$$

ending in x such that $(\sigma <p = \mathsf{T}> \sigma', \tau <p = \mathsf{T}> \tau')$ is a u-corresponding pair for (S, T) and $\sigma <p = \mathsf{T}> \sigma' \in pre(\pi_S(i, e))$ and $\tau <p = \mathsf{T}> \tau' \in pre(\pi_T(i, e))$, and σ' does not pass through any $g \in \mathcal{F}$ such that $g \rightsquigarrow_S x\,(r)$, then the result follows from Part (2) of Lemma 140. Thus it suffices to show that the interpretation i can be changed at predicates in $body_S(p)$ and $body_T(p)$ to achieve this outcome.
Choose any unitary vx-segment σ' in $body_S(p)$ such that for any $q \in whilePreds(body_S(p))$ such that $\neg(q \searrow_S x)$, σ' does not pass through $<q = \mathsf{T}>$, and that if σ' passes through $<q = Z>$ for any $q \in ifPreds(body_S(p))$, then $\neg(q \searrow_T x\,(\neg Z))$ holds (possible by Lemma 143).
We may make the following assumptions about i; for any $q \in whilePreds(body_S(p))$ such that $p \searrow_S q \searrow_S x$, then $q^i(\mathbf{t}) = \mathsf{T}$ if $q(\mathbf{t})$ is S-external and $q^i(\mathbf{t}) = \mathsf{F}$ otherwise; and for any $q \in whilePreds(body_S(p))$ such that $p \searrow_S q \wedge \neg(q \searrow_S x)$, or the same condition holds in T, then set $q^i(\mathbf{t}) = \mathsf{F}$ for all $\mathbf{t}$. By Theorem 109, this ensures that in both S and T, i defines a path which reaches the second occurrence of p after following a unitary path through its body. We also assume that if σ' passes through $<q = Z>$ for any $q \in ifPreds(body_S(p))$, or $q \searrow_T x\,(Z)$, then $q^i(\mathbf{t}) = Z$ for all $\mathbf{t}$. By the choice of σ', these are not contradictory assumptions.
By Part (1) of Lemma 133, $\sigma <p = \mathsf{T}> \sigma' \in pre(\pi_S(i, e))$ holds and there exists $\tau' \in pre(\Pi(body_T(p)))$ such that σ' and τ' satisfy the conditions required, proving the Lemma. $\square$

For Proposition 145 and Lemma 146, recall the definition of a preupdated symbol, Definition 91.

Proposition 145 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be u-equivalent LFL schemas. Let $q \in whilePreds(S)$ and assume $x \in Symbols(body_S(q)) \cap \mathcal{N}_S(u)$ and $above_{body_S(q)}(x) = x$. Then x is preupdated in S if and only if x is preupdated in T.*

Proof. This follows from Part (1) of Lemma 130. $\square$

Lemma 146 states that under certain strict conditions, given $\{u, \omega\}$-equivalent LFL schemas S and T and an interpretation i, if the path $\pi_S(i, e)$ passes through some $x \in \mathcal{N}_S(u)$ and $\pi_T(i, e)$ also passes through x, but not through the 'same occurrence' of x, then the predicate terms thus defined by the two paths are distinct. We have been unable to prove the Lemma under the weaker assumption that S and T are just u-equivalent for $u \in \mathcal{V}$, but we conjecture that it also holds in this case.

Lemma 146 *Let $u \in \mathcal{V} \cup \{\omega\}$ and let S, T be $\{u, \omega\}$-equivalent LFL schemas containing a while predicate p. Let i be an interpretation and let*

$$\sigma = \sigma' <p = \mathsf{T}> \sigma'' \in pre(\pi_S(i, e))$$

and

$$\tau = \tau' <p = \mathsf{T}> \tau'' \in pre(\pi_T(i, e))$$

with $\sigma'' \in \Pi(body_S(p))$ *and* $\tau'' \in \Pi(body_T(p))$. *Assume that*

$$(\sigma' \lessdot p = \mathsf{T} \gtrdot, \tau' \lessdot p = \mathsf{T} \gtrdot)$$

is a u-correspondence, and τ'' *is unitary. Assume that* σ'' *passes through a symbol* $x \in \mathcal{N}_S(u)$ *and* τ'' *does not pass through* x. *Let* γ *be the prefix of* σ'' *preceding the first occurrence of* x *(there may be two if* x *is a while predicate). Suppose that* $\pi_T(i,e)$ *passes through* x *and let* ν *be the prefix of* $\pi_T(i,e)$ *preceding one of the occurrences of* x *in* $\pi_T(i,e)$. *Then*

$$\mathcal{M}[\![schema(\sigma'\gamma)]\!]_e(\mathbf{refvec}_S(x)) \neq \mathcal{M}[\![schema(\nu)]\!]_e(\mathbf{refvec}_T(x))$$

holds.

Proof. Observe that σ'' is unitary by Lemma 141, Part (2), and x is not, in fact, a while predicate by Part (3) of this Lemma. We define a finite sequence $x_1 = x, \ldots, x_n$ with each $x_r \in \mathcal{F}$ if $r \geq 2$, as follows. The path τ'' does not pass through any x_r. Given x_r, we choose x_{r+1} to satisfy $x_{r+1} \rightsquigarrow_{schema(\gamma)} x_r$ (or $x_{r+1} \rightsquigarrow_{schema(\gamma)} v \in Refset_S(x_r)$ if $r = 1$) such that the path τ'' does not pass through any x_r. If no such x_{r+1} exists then put $r = n$. Let $q \in whilePreds(T)$ lie immediately above x_n.
Let γ' be the prefix of γ preceding x_n. Thus the vector term

$$\mathcal{M}[\![schema(\sigma'\gamma)]\!]_e(\mathbf{refvec}_S(x))$$

contains $\mathcal{M}[\![schema(\sigma'\gamma')]\!]_e(\mathbf{refvec}_S(x_n))$ as a vector subterm, and so if the Lemma is false, the vector term $\mathcal{M}[\![schema(\nu)]\!]_e(\mathbf{refvec}_T(x))$ has a vector subterm $\mathbf{t} = \mathcal{M}[\![schema(\sigma'\gamma')]\!]_e(\mathbf{refvec}_S(x_n))$.
Let G be the set of function symbols g satisfying $g \rightsquigarrow_{\gamma'} v$ for some $v \in Refset_S(x_n)$. We now show that for any component $w \in Refset_T(x_n)$, there are no assignments to w occurring on ν after all g-assignments for $g \in G$ have occurred on τ''. Let $w \in Refset_T(x_n)$ and assume w is the rth component of $\mathbf{refvec}_T(x_n)$. Suppose first that there exists $g \in G$ such that $g \rightsquigarrow_S x_n\,(r)$. By the choice of x_n, both σ'' and τ'' pass through g and define the same term $t = \mathcal{M}[\![schema(\sigma'\gamma')]\!]_e(assign_S(g))$ in each case, by Part (2) of Corollary 140. Clearly t is a component of $\mathbf{t}$ and so by the liberality of T, if the Lemma is false there are no assignments to w in ν after the occurrence of g in τ''. On the other hand, if no such $g \in G$ exists, then by Lemma 144, it follows similarly that there are no assignments to w in ν after the first occurrence of p.
Assume that q is the while predicate lying immediately above x_n in S and T. Clearly $p \searrow_S q \vee p = q$ holds. Suppose first that x_n is preupdated in T. Then ν has a suffix $\lessdot q = \mathsf{T} \gtrdot \bar{\nu}$ beginning after τ'' has ended such that $\bar{\nu} \in pre(\Pi(body_T(q)))$ passes through some $g \in \mathcal{F}$ with $g \rightsquigarrow_{body_T(q)} x_n$. This immediately contradicts what we have just shown. Thus by Proposition 145, x_n is not preupdated in T, and hence not preupdated in S. Thus x_n is postupdated in S and so there exists $h \in \mathcal{F}$ satisfying $back_S(q, h, x_n)$ such that the segment of σ'' lying in $body_S(q)$ passes through h. Since by Lemma 141, Part (2), and Part (1) of Corollary 140, τ'' also enters $body_T(q)$, by Lemma 141, Part (1), τ'' also passes through h. Since τ is unitary, by Lemma 134 h occurs on τ'' after all occurrences of G, again giving a contradiction. $\square$

The following Theorem shows that $\{u,\omega\}$-equivalence implies conditions (12) and (13) of $\{u,\omega\}$-similarity.

Theorem 147 (main result of this section) *Let $u \in \mathcal{V}\cup\{\omega\}$ and let S, T be u,ω-equivalent LFL schemas. If $p \in \mathit{whilePreds}(S)$, $f \in \mathit{Funcs}(S)$, $x \in \mathcal{N}_S(u)$, $f \rightsquigarrow_S x$ and $x \in \mathcal{N}_S(u)$ with $v = \mathit{assign}_S(f)$ and $w = \mathit{assign}_T(f)$, then*

$$f \rightsquigarrow_{\mathit{body}_S(p)} v \wedge v \rightsquigarrow_{\mathit{body}_S(p)} x$$

$$\Longleftrightarrow$$

$$f \rightsquigarrow_{\mathit{body}_T(p)} w \wedge w \rightsquigarrow_{\mathit{body}_T(p)} x$$

holds, and if also $q \in \mathit{ifPreds}(S)$ and $p \searrow_S q$ and $Z \in \{\mathsf{T},\mathsf{F}\}$ and $v \rightsquigarrow_{\mathit{body}_S(p)} x$ holds, then

$$(\mathit{outif}_{\mathit{body}_S(p)}(q,Z,f,v) \vee \mathit{thru}_{\mathit{body}_S(p)}(q,Z,f,v))$$
$$\Longleftrightarrow$$
$$(\mathit{outif}_{\mathit{body}_T(p)}(q,Z,f,w) \vee \mathit{thru}_{\mathit{body}_T(p)}(q,Z,f,w))$$

holds.

Proof. Suppose first that $f \rightsquigarrow_{\mathit{body}_S(p)} v \wedge v \rightsquigarrow_{\mathit{body}_S(p)} x$ holds. We will show that $f \rightsquigarrow_{\mathit{body}_T(p)} w \wedge w \rightsquigarrow_{\mathit{body}_T(p)} x$ holds; the converse follows similarly.
By Part (1) of Lemma 130, $w \rightsquigarrow_{\mathit{body}_T(p)} x$ holds, so we need only prove $f \rightsquigarrow_{\mathit{body}_T(p)} w$. Since $x \in \mathcal{N}_S(u)$, by Lemma 27 we may assume that there is an xF-segment in S for $xF \in \mathcal{F}^*r$ and $r \in \mathcal{P} \cap \mathcal{N}_S(u)$. By the hypotheses given, there is an fxF-segment ν in S which can be written $\nu = \nu' \lessdot p = \mathsf{T} \gtrdot \nu''\nu'''$, where ν' is an fv-segment and ν'' is a vx-segment. Let $\mu \in \mathit{pre}(\Pi(S))$ be unitary and end just before f. Thus there is an fxF-couple $I = \{i,j\}$ for S such that $\mathit{head}_S(I)$ is the prefix of $\mu\nu$ preceding the final occurrence of r. Let Ω be the set of all pairs (p',T) such that $p' \in \mathit{whilePreds}(S)$ and $p' \searrow_S p$. By Part (2) of Lemma 74 if $u = \omega$, we may assume that I is Ω-finite; and since μ is unitary, by Lemma 69 if $\Omega \neq \emptyset$, we may assume that I is Ω-minimal.
Thus I is an $r(\mathbf{t})$-couple for some vector term $\mathbf{t}$ of which $\mathcal{M}[\![\mathit{schema}(\mu)]\!]_e(\mathbf{refvec}_S(f))$ and $\mathcal{M}[\![\mathit{schema}(\mu\nu'\nu'')]\!]_e(\mathbf{refvec}_S(x))$ are vector subterms. We now look at T. Let $i \in I$ be terminating; clearly one element of I is. Clearly $\pi_T(i,e)$ passes through f and hence through $\lessdot p = \mathsf{T}\gtrdot$. Thus $\pi_T(i,e)$ has a prefix $\gamma' \lessdot p = \mathsf{T}\gtrdot \gamma''$, where $\gamma'' \in \Pi(\mathit{body}_T(p))$ and by the Ω-minimality assumption, p is superunitary in this prefix. By Lemma 141 , Part (2), γ'' is unitary. By Lemma 146 applied to f and x, γ'' passes through f, but if it passes through x, then it does not define the term

$$x(\mathcal{M}[\![\mathit{schema}(\mu\nu'\nu'')]\!]_e(\mathbf{refvec}_S(x)))$$

at that point. Thus γ'' ends in an fw-segment.
We have shown that $f \rightsquigarrow_{\mathit{body}_S(p)} v \iff f \rightsquigarrow_{\mathit{body}_T(p)} w$ holds. Now assume $f \rightsquigarrow_{\mathit{body}_S(p)} v$ and

$$\neg(\mathit{outif}_{\mathit{body}_S(p)}(q,Z,f,v) \vee \mathit{thru}_{\mathit{body}_S(p)}(q,Z,f,v))$$

holds. We will show that $\neg(\mathit{outif}_{\mathit{body}_T(p)}(q,Z,f,w) \vee \mathit{thru}_{\mathit{body}_T(p)}(q,Z,f,w))$ holds; again the converse follows similarly. In the construction above, choose a segment ν'

which does not go through $\lessdot q = Z \gtrdot$. If $\mathit{outif}_{\mathit{body}_T(p)}(q, Z, f, w) \vee \mathit{thru}_{\mathit{body}_T(p)}(q, Z, f, w)$ holds, then γ'' does pass through $\lessdot q = Z \gtrdot$. Let $q(\mathbf{t}')$ be the term that $\mathit{head}_T(I)$ defines at this point. By Lemma 146 applied to q, neither $\pi_S(i, e)$ nor $\pi_S(j, e)$ passes through $q(\mathbf{t}')$. Thus we may replace i and j with $i(q(\mathbf{t}') = \neg(Z))$ and $j(q(\mathbf{t}') = \neg(Z))$ without changing the paths through S. The paths through T defined by the new couple pass through $\lessdot q = \neg Z \gtrdot$ instead of $\lessdot q = Z \gtrdot$ within γ''; since γ'' is unitary, we have shown $\neg(\mathit{outif}_{\mathit{body}_T(p)}(q, Z, f, w) \vee \mathit{thru}_{\mathit{body}_T(p)}(q, Z, f, w))$. Hence

$$\mathit{outif}_{\mathit{body}_T(p)}(q, Z, f, w) \vee \mathit{thru}_{\mathit{body}_T(p)}(q, Z, f, w)$$

$$\Rightarrow$$

$$\mathit{outif}_{\mathit{body}_S(p)}(q, Z, f, v) \vee \mathit{thru}_{\mathit{body}_S(p)}(q, Z, f, v)$$

holds, and the converse follows similarly. $\square$

15 The Main Theorem and Further Directions

The main result of this paper is the following.

Theorem 148 *Let S, T be LFL schemas. Then*

$$S \cong T \iff S \mathit{\,simil\,} T$$

holds. If $V \subseteq \mathcal{V} \cup \{\omega\}$ and $\omega \in V$ then

$$S \cong_V T \iff S \mathit{\,simil}_V\, T$$

holds. In particular, it is decidable in polynomial time whether S and T are equivalent.

Proof. The first assertion is a special case of the second (where V is the set containing all variables assigned in either S or T, plus ω). To prove the second assertion, assume $S \mathit{\,simil}_V\, T$ for LFL schemas S, T with V as given. Then $S \cong_V T$ follows from Theorem 55. Conversely, if $S \cong_V T$, then Conditions (1)–(4) of u-similarity for each $u \in V$ follow from Theorem 75, Conditions (2) and (3) follow from Theorem 96, Condition (7) from Theorem 118, Condition (8) from Theorem 126, Condition (9) from Theorem 127, Condition (10) from Theorem 105, Condition (8) from Theorem 126, Condition (11) from Theorem 137 and Conditions (12) and (13) from Theorem 147 (since $\omega \in V$ in this last case). The polynomial time bound follows from Theorem 39. □

Since we have been unable to prove that $S \cong_v T$ implies Conditions (12) and (13) of $S \mathit{\,simil}_u\, T$ for LFL schemas S, T and $v \in \mathcal{V}$, the hypothesis $\omega \in V$ in Theorem 148 is necessary.

Of the various related problems which seem worth studying, two strike us as being particularly promising.

15.1 Computing minimal slices of schemas

For the purpose of program slicing, given a schema S and variable v, it is of interest to be able to compute those minimal slices of S (with minimality defined by symbol sets) which are v-equivalent to S and which preserve termination. It follows from Theorem 76 and Part (1) of Theorem 42 that for any $u \in \mathcal{V}$, the minimal slice T of an LFL schema S such that $S \cong_u T$ and $\mathcal{M}[\![S]\!]^j_d \neq \perp \Rightarrow \mathcal{M}[\![T]\!]^j_d \neq \perp$ always holds is precisely the slice of S such that $\mathit{Symbols}(T) = \mathcal{N}_S(v)$ holds. The first author has proved in [40] that this also holds if the linearity hypothesis is replaced by function-linearity (a schema is function-linear if it does not contain more than one occurrence of the same function symbol), provided that the definition of $\mathcal{N}_S(u)$ is generalised to allow for multiple occurrences of predicate symbols.

If S is merely free and linear then $S \cong_u T$ need not imply $\mathit{Symbols}(T) \supseteq \mathcal{N}_S(u)$, as the example of Figure 6 shows. Owing to the constant g-assignment, S is not liberal, though it is free. Clearly $f \in \mathcal{N}_S(u)$ holds, but the slice of S obtained by deleting the f-assignment, which is also free, is u-equivalent to S. It is also ω-equivalent to S, and hence satisfies the termination requirement for slices.

It would be of interest to find a method of computing the minimal slice of S satisfying these conditions under weaker hypotheses than the assumption that S is liberal, free and function-linear.

15.2 Using Schema Transformations to Construct Equivalent Schemas

Given a linear schema S and $u \in \mathcal{V} \cup \{\omega\}$, it can be shown using Theorem 55 that the following transformations of S preserve u-equivalence.

- Changing the variables mentioned in S in any way that preserves u-similarity.
- Replacing S by a slice T of S, such that T contains every element of $\mathcal{N}_S(u)$.
- Pulling out a subschema from an if subschema of S; that is, replacing a subschema

$$\begin{array}{ll} \mathit{if}\ p(\mathbf{v}) & \mathit{then}\ \ S_1 S_2 \\ & \mathit{else}\ \ S_3 \end{array}$$

  of S by the schema

$$\begin{array}{l} S_1 \\ \begin{array}{ll} \mathit{if}\ p(\mathbf{v}) & \mathit{then}\ \ S_2 \\ & \mathit{else}\ \ S_3 \end{array} \end{array}$$

  provided that this does not create a new fx-segment μ for $f \in \mathit{Funcs}(S_1)$ and $x \in \mathcal{N}_S(u) \cup \{u\}$ such that either $x = p$ or μ passes through $\lessdot p = \mathsf{F} \gtrdot$. Also, if $u = \omega$ then S_1 must not contain a while predicate, otherwise Condition (7) of simil_u is violated. Clearly the true and false parts of p may be interchanged.
- Changing the order of 'towers' of if predicates; that is, interchanging $p(\mathbf{u})$ and $q(\mathbf{v})$ in a subschema

$$
\begin{array}{l}
\textit{if } p(\mathbf{u}) \ \textit{ then} \\
\qquad\qquad \{ \\
\qquad\qquad \textit{if } q(\mathbf{v}) \ \textit{ then } \ T \\
\qquad\qquad\qquad\quad \textit{else} \ \ \Lambda \\
\qquad\qquad \} \\
\qquad \textit{else} \ \ \Lambda
\end{array}
$$

of S. Again, the true and false parts of p or q may be interchanged.

- Replacing a subschema S_1S_2 of S by S_2S_1 to give a schema T, provided that no variable is assigned in both S_1 and S_2, and S_1S_2 contains no fx-segment with $f \in \mathit{Funcs}(S_1)$ and $x \in \mathit{Symbols}(S_2)$, and the same statement holds with (S, 1, 2) replaced by (T, 2, 1).

We conjecture that given any LFL schema S, all u-similar LFL schemas can be obtained from S by a sequence of these transformations and their inverses.
It may also be possible to prove that given an LFL schema S, any u-equivalent LFL schema may be reached from S by a finite sequence of such transformations without using Theorem 148, thus giving an alternative (and possibly shorter) way of proving this theorem than the one we have given in this report.

References

[1] Z. Manna, Mathematical Theory of Computation, McGraw–Hill, 1974.

[2] M. R. Laurence, S. Danicic, M. Harman, R. M. Hierons, J. D. Howroyd, Equivalence of conservative, free, linear schemas is decidable, Theoretical Computer Science 290 (1) (2002) 831–862.

[3] M. S. Paterson, Equivalence problems in a model of computation, PhD thesis, University of Cambridge, UK (1967).

[4] E. Ashcroft, Z. Manna, Translating program schemas to while-schemas, SIAM Journal on Computing 4 (2) (1975) 125–146.

[5] D. C. Luckham, D. M. R. Park, M. S. Paterson, On formalised computer programs, Journal of Computer and System Sciences 4 (1970) 220–249.

[6] Y. I. Ianov, The logical schemes of algorithms, in: Problems of Cybernetics, Vol. 1, Pergamon Press, New York, 1960, pp. 82–140.

[7] V. K. Sabelfeld, An algorithm for deciding functional equivalence in a new class of program schemes, Journal of Theoretical Computer Science 71 (1990) 265–279.

[8] E. J. Weyuker, Modifications of the program scheme model, Journal of Computer and System Sciences 18 (3) (1979) 281–293.

[9] E. J. Weyuker, Translatability and decidability questions for restricted classes of program schemas, SIAM Journal on Computing 8 (4) (1979) 587–598.

[10] C. A. Albayrak, T. Noll, The WHILE hierarchy of program schemes is infinite, Lecture Notes in Computer Science 1378 (1998) 35–47.

[11] A. K. Chandra, On the decision problems of program schemas with commutative and invertable functions, in: Conference Record of the ACM Symposium on Principles of Programming Languages, ACM SIGACT-SIGPLAN, Boston, Massachusetts, 1973, pp. 235–242.

[12] A. K. Chandra, Z. Manna, Program schemas with equality, in: Conference Record, Fourth Annual ACM Symposium on Theory of Computing, Denver, Colorado, 1972, pp. 52–64.

[13] A. K. Chandra, Z. Manna, On the power of programming features, Computer Languages 1 (3) (1975) 219–232.

[14] E. Ashcroft, Z. Manna, A. Pnueli, Decidable properties of monadic functional schemas, Journal of the ACM 20 (3) (1973) 489–499.

[15] M. S. Paterson, A simple monadic schema which is not equivalent to any program schema, unpublished memo, MIT, Project Mac, dec 1970 .

[16] A. De Lucia, A. R. Fasolino, M. Munro, Understanding function behaviours through program slicing, in: 4^{th} IEEE Workshop on Program Comprehension, IEEE Computer Society Press, Los Alamitos, California, USA, Berlin, Germany, 1996, pp. 9–18.

[17] G. Canfora, A. Cimitile, A. De Lucia, G. A. D. Lucca, Software salvaging based on conditions, in: International Conference on Software Maintenance (ICSM'96), IEEE Computer Society Press, Los Alamitos, California, USA, Victoria, Canada, 1994, pp. 424–433.

[18] A. Cimitile, A. De Lucia, M. Munro, A specification driven slicing process for identifying reusable functions, Software maintenance: Research and Practice 8 (1996) 145–178.

[19] K. B. Gallagher, J. R. Lyle, Using program slicing in software maintenance, IEEE Transactions on Software Engineering 17 (8) (1991) 751–761.

[20] K. B. Gallagher, Evaluating the surgeon's assistant: Results of a pilot study, in: Proceedings of the International Conference on Software Maintenance, IEEE Computer Society Press, Los Alamitos, California, USA, 1992, pp. 236–244.

[21] M. Weiser, J. R. Lyle, Experiments on slicing–based debugging aids, Empirical studies of programmers, Soloway and Iyengar (eds.), Molex, 1985, Ch. 12, pp. 187–197.

[22] H. Agrawal, R. A. DeMillo, E. H. Spafford, Debugging with dynamic slicing and backtracking, Software – Practice and Experience 23 (6) (1993) 589–616.

[23] M. Kamkar, Interprocedural dynamic slicing with applications to debugging and testing, PhD Thesis, Department of Computer Science and Information Science, Linköping University, Sweden, available as Linköping Studies in Science and Technology, Dissertations, Number 297 (1993).

[24] J. R. Lyle, M. Weiser, Automatic program bug location by program slicing, in: 2^{nd} International Conference on Computers and Applications, IEEE Computer Society Press, Los Alamitos, California, USA, Peking, 1987, pp. 877–882.

[25] D. W. Binkley, The application of program slicing to regression testing, in: M. Harman, K. Gallagher (Eds.), Information and Software Technology Special Issue on Program Slicing, Vol. 40, Elsevier, 1998, pp. 583–594.

[26] R. Gupta, M. J. Harrold, M. L. Soffa, An approach to regression testing using slicing, in: Proceedings of the IEEE Conference on Software Maintenance, IEEE Computer Society Press, Los Alamitos, California, USA, Orlando, Florida, USA, 1992, pp. 299–308.

[27] M. Harman, S. Danicic, Using program slicing to simplify testing, in: 2^{nd} EuroSTAR conference on Software Testing Analysis and Review, Brussels, 1994.

[28] G. Canfora, A. Cimitile, M. Munro, RE^2: Reverse engineering and reuse re-engineering, Journal of Software Maintenance : Research and Practice 6 (2) (1994) 53–72.

[29] D. Simpson, S. H. Valentine, R. Mitchell, L. Liu, R. Ellis, Recoup – Maintaining Fortran, ACM Fortran forum 12 (3) (1993) 26–32.

[30] J. Beck, D. Eichmann, Program and interface slicing for reverse engineering, in: IEEE/ACM 15^{th} Conference on Software Engineering (ICSE'93), IEEE Computer Society Press, Los Alamitos, California, USA, 1993, pp. 509–518.

[31] A. Cimitile, A. De Lucia, M. Munro, Identifying reusable functions using specification driven program slicing: a case study, in: Proceedings of the IEEE International Conference on Software Maintenance (ICSM'95), IEEE Computer Society Press, Los Alamitos, California, USA, Nice, France, 1995, pp. 124–133.

[32] S. Horwitz, J. Prins, T. Reps, Integrating non–interfering versions of programs, ACM Transactions on Programming Languages and Systems 11 (3) (1989) 345–387.

[33] J. M. Bieman, L. M. Ott, Measuring functional cohesion, IEEE Transactions on Software Engineering 20 (8) (1994) 644–657.

[34] J. J. Thuss, An investigation into slice–based cohesion metrics, Master's thesis, Michigan Technological University (1988).

[35] A. Lakhotia, Rule–based approach to computing module cohesion, in: Proceedings of the 15^{th} Conference on Software Engineering (ICSE-15), 1993, pp. 34–44.

[36] D. W. Binkley, K. B. Gallagher, Program slicing, in: M. Zelkowitz (Ed.), Advances of Computing, Volume 43, Academic Press, 1996, pp. 1–50.

[37] A. De Lucia, Program slicing: methods and applications, in: 1st IEEE International workshop on Source Code Analysis and Manipulation, IEEE Computer Society Press, Los Alamitos, California, USA, Florence, Italy, 2001, pp 142-149.

[38] F. Tip, A survey of program slicing techniques, Tech. Rep. CS-R9438, Centrum voor Wiskunde en Informatica, Amsterdam (1994).

[39] M. Weiser, Program slices: Formal, psychological, and practical investigations of an automatic program abstraction method, PhD thesis, University of Michigan, Ann Arbor, MI (1979).

[40] M. R. Laurence, Characterising minimal semantics-preserving slices of function-linear, free, liberal program schemas, submitted to Journal of Logic and Algebraic Programming in July 2005 .