DECIDABILITY OF HIGHER-ORDER MATCHING

COLIN STIRLING

School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK
e-mail address: cps@staffmail.ed.ac.uk

ABSTRACT. We show that the higher-order matching problem is decidable using a game-theoretic argument.

CONTENTS

1. INTRODUCTION

Higher-order unification is the problem given an equation $t = u$ containing free variables is there a solution substitution θ such that $t\theta$ and $u\theta$ have the same normal form? The terms t and u are from the simply typed lambda calculus and the same normal form is with respect to $\beta\eta$-equivalence. Higher-order matching is the particular instance when the term u is closed; can t be pattern matched to u? Although higher-order unification is undecidable (even if free variables are only second-order [6]), higher-order matching was conjectured to be decidable by Huet [7]. If matching is decidable then it is known to have non-elementary complexity [17, 23]. Decidability has been proved for the general problem

up to order 4 (by showing decidability of observational equivalence of lambda terms) and for various special cases [12, 13, 14, 15, 3]. Comon and Jurski define tree automata that characterise all solutions to a 4th-order problem, thereby, showing that they form a regular set [2]. Loader showed that matching is undecidable for the variant definition of the same normal form that only uses β-equivalence by encoding lambda definability as matching [10]: see [8] for a proof that uses the halting problem. An excellent source of information about unification and matching is [5].

In this paper, we confirm Huet's conjecture that higher-order matching is decidable. The proof first appeals to Padovani's and Schubert's reduction of matching to the (dual) interpolation problem [14, 13] and is then in the tradition described by Dowek [5]: "these [decidability] proofs are rather technical ... because they all proceed by transforming potential solutions into smaller ones cutting and pasting term pieces". The proof method is partly inspired by model-checking games (such as in [19]) where a model, a transition graph, is traversed relative to a property and players make choices at appropriate positions. Given a (dual) interpolation problem P, we define a game where the model is a closed lambda term t in η-long normal form that is a potential solution to P; game-playing moves around the term t (dependent on P). The game captures the dynamics of β-reduction on t without changing it (using substitution). Unlike model-checking games, play may arbitrarily jump around a term because of binding.

The principal virtue of the game is that small pieces of a solution term can be understood in terms of sequences of positions in the game and how they, thereby, contribute to solving the problem P. We identify regions of a term, "tiles", and classify them according to these intervals of play. Two transformations that preserve solution terms are introduced. With these, we show that 3rd-order matching is decidable using the small model property: if there is a solution to a problem then there is a small solution to it. In [20], we introduced the game and two more transformations that uniformly solved known subcases of matching (including 4th-order). The key observation for decidability at 3rd-order is the tree-model property: each play descends a branch of a solution term because of the paucity of binding. For higher-orders, the idea is to induce as far as possible the tree-model property, to tame play jumping within a term due to binding. The mechanism for doing this involves unfolding a lambda term with respect to game playing which is analogous to unravelling a transition system in modal logic. Unfolding involves "tile lowering", copying regions of a term down its branches. The proof of decidability at higher-order uses unfolding and from its combinatorial properties the small model property follows.

In Section 2 we introduce higher-order matching and (dual) interpolation and in Section 3 we define some basic ingredients of the problem. In Section 4 we describe the tree-checking game that characterises (dual) interpolation and in Section 5 some properties of the resulting game are highlighted. In Section 6 we identify tiles as regions of terms and define their plays and in Section 7 we show that 3rd-order matching is decidable using a tree model property of game playing; the first step in this argument is to define a partition of each play in a game. The definition of partition is extended to all orders in Section 8. This then forms the basis for the notion of tile unfolding that is described in Section 9 and how it leads to decidability of matching via the small model property. The complexity analysis is the size of a smallest term, if there is one, that solves a problem. However, the bounds are extremely coarse. Finally, we conclude with general remarks and ideas for future work.

2. Matching and dual interpolation

Simple types are generated from a single base type $\mathbf{0}$ using the binary $\to$ operator. For simplicity, we assume only one base type: everything that is to follow can be extended to arbitrary many base types. A type is $\mathbf{0}$ or $A \to B$ where A and B are types. If $A \neq \mathbf{0}$ then it has the form $A_1 \to \ldots \to A_n \to \mathbf{0}$, assuming $\to$ associates to the right, which we abbreviate to $(A_1, \ldots, A_n, \mathbf{0})$ following Ong [11]. A standard definition of *order* is: the order of $\mathbf{0}$ is 1 and the order of $(A_1, \ldots, A_n, \mathbf{0})$ is $k + 1$ where k is the maximum of the orders of the A_is.

Terms of the simply typed lambda calculus are built from a countable set of typed variables $x, y, \ldots$ and constants $a, f, \ldots$ (so, each variable and constant has a unique type).

Definition 2.1. The set of simply typed lambda terms is the smallest set T such that
(1) if x (f) has type A then $x : A \in T$ ($f : A \in T$),
(2) if $t : B \in T$ and $x : A \in T$ then $\lambda x.t : A \to B \in T$,
(3) if $t : A \to B \in T$ and $u : A \in T$ then $(tu) : B \in T$.

The *order* of a typed term $t : A$ is the order of its type A.

In a sequence of unparenthesised applications, we adopt the usual convention that application associates to the left; so $tu_1 \ldots u_k$ is $((\ldots (tu_1) \ldots)u_k)$. The usual definitions of free and bound variable occurrences and when a typed term is closed are assumed. Moreover, we assume the standard definitions and properties of α-equivalence, β-reduction, η-reduction and $\beta\eta$-equivalence, $=_{\beta\eta}$, such as strong normalisation of β-reduction: see for instance, Barendregt [1].

Definition 2.2. A *matching problem* is an equation $v = u$ where $v, u : A$ and u is closed. The *order* of the problem is the maximum of the orders of the free variables $x_1, \ldots, x_n$ in v. A *solution* is a sequence of terms $t_1, \ldots, t_n$ such that $v\{t_1/x_1, \ldots, t_n/x_n\} =_{\beta\eta} u$ where $v\{t_1/x_1, \ldots, t_n/x_n\}$ is the simultaneous substitution of t_i for each free occurrence of x_i in v for each $i : 1 \leq i \leq n$.

The decision question is: given a matching problem, does it have a solution? It suffices to consider the case when $A = \mathbf{0}$: a problem $v = u$ of type $(A_1, \ldots, A_n, \mathbf{0})$ reduces to the equivalent problem $vf_1 \ldots f_n = uf_1 \ldots f_n$ of type $\mathbf{0}$ where each $f_i : A_i$ is a fresh constant which is not allowed in solution terms. Equivalent variants of matching include the "range question" and the "pattern matching problem" [18, 8].

As described by Huet [7], every simply typed lambda calculus term is $\beta\eta$-equivalent to a unique term in η-*long normal form*,
(1) if $t : \mathbf{0}$ then it is $u : \mathbf{0}$ where u is a constant or a variable, or $u\,t_1 \ldots t_k$ where $u : (B_1, \ldots, B_k, \mathbf{0})$ is a constant or a variable and each $t_i : B_i$ is in η-long normal form,
(2) if $t : (A_1, \ldots, A_n, \mathbf{0})$ then t is $\lambda y_1 \ldots y_n.t'$ where each $y_i : A_i$ and $t' : \mathbf{0}$ is in η-long normal form.

Throughout, we write $\lambda z_1 \ldots z_n$ for $\lambda z_1 \ldots \lambda z_n$. A term is *well-named* if each occurrence of a variable y within a lambda abstraction is unique. In the following, we assume that a term in *normal form* is always in η-long normal form; consequently, β-equality and $\beta\eta$-equality coincide (for instance, see [22]).

Definition 2.3. Assume $u : \mathbf{0}$ and each $v_i : A_i$, $1 \leq i \leq n$, is a closed term in normal form and $x : (A_1, \ldots, A_n, \mathbf{0})$.
(1) $x\,v_1 \ldots v_n = u$ is an *interpolation equation*.
(2) $x\,v_1 \ldots v_n \neq u$ is an *interpolation disequation*.

(3) A finite family of interpolation equations $x\, v_1^i \ldots v_n^i = u_i$, where $i : 1 \leq i \leq m$, with the same free variable x, is an *interpolation problem P*.

(4) A finite family of interpolation equations and disequations $x\, v_1^i \ldots v_n^i \approx_i u_i$, when $i : 1 \leq i \leq m$, with the same free variable x and where each $\approx_i \in \{=, \neq\}$, is a *dual interpolation problem P*.

(5) The type of problem P is that of x and the order of P is the order of x.

(6) A *solution* of P of type A is a closed term $t : A$ in normal form such that for each equation $t\, v_1^i \ldots v_n^i =_\beta u_i$ and, in the case of dual interpolation, for each disequation $t\, v_1^i \ldots v_n^i \neq_\beta u_i$. We write $t \models P$ if t is a solution of P.

Conceptually, (dual) interpolation is simpler than matching because there is a single variable x that appears at the head of each (dis)equation. However, Schubert shows that a matching problem of order n reduces to an interpolation problem of order at most $n+2$ and Padovani shows it reduces to a dual interpolation problem of order n, [14, 13]. Consequently, the higher-order matching problem reduces to the following decision question.

Decision Question Given a (dual) interpolation problem P, is there a term $t \models P$?

It is this question that is solved positively in the rest of the paper. Throughout, we assume a fixed dual interpolation problem P of type A whose order is greater than 1 (as an order 1 problem is easily decided). A problem P has the form $x\, v_1^i \ldots v_n^i \approx_i u_i$, $1 \leq i \leq m$, where the normal form terms v_j^i and u_i are well-named and no pair share bound variables.

3. Preliminaries

We start with some examples of interpolation problems.

Example 3.1. The following is a 4th-order problem

$$
\begin{array}{rcl}
x\, \lambda y_1 y_2.y_1 y_2 & = & f a \\
x\, \lambda y_3 y_4.y_3 (y_3 y_4) & = & f(f a)
\end{array}
$$

with $x : (((\mathbf{0}, \mathbf{0}), \mathbf{0}, \mathbf{0}), \mathbf{0})$ and $f : (\mathbf{0}, \mathbf{0})$. $\qquad\square$

Example 3.2. The problem $x(\lambda z.z) = f(\lambda x_1 x_2 x_3.x_1 x_3)a$ has order 3 where x has type $((\mathbf{0}, \mathbf{0}), \mathbf{0})$ and $f : (((\mathbf{0}, \mathbf{0}), \mathbf{0}, \mathbf{0}, \mathbf{0}), \mathbf{0}, \mathbf{0})$ assuming $x_2 : \mathbf{0}$. $\qquad\square$

Example 3.3. The next equation, due to Luke Ong, is 5th-order

$$
x(\lambda y_1 y_2.y_1(\lambda y_3.y_2(y_1(\lambda y_4.y_3)))) = h(g(h(ha)))
$$

with $x : ((((\mathbf{0}, \mathbf{0}), \mathbf{0}), (\mathbf{0}, \mathbf{0}), \mathbf{0}), \mathbf{0})$ and $g, h : (\mathbf{0}, \mathbf{0})$. $\qquad\square$

A right term u of an interpolation (dis)equation $xv_1 \ldots v_n \approx u$ may contain bound variables, such as $f(\lambda x_1 x_2 x_3.x_1 x_3)a$ of Example 3.2. Let $X = \{x_1, \ldots, x_k\}$ be the set of bound variables in u and let $C = \{c_1, \ldots, c_k\}$ be a fresh set of constants where each c_i has the same type as x_i.

Definition 3.4. The *ground closure* of a closed term w, whose bound variables belong to X, with respect to C, written $\mathrm{Cl}(w, X, C)$, is defined inductively:

(1) if $w = a : \mathbf{0}$ then $\mathrm{Cl}(w, X, C) = \{a\}$,

(2) if $w = f\, w_1 \ldots w_n$ then $\mathrm{Cl}(w, X, C) = \{w\} \cup \bigcup \{\mathrm{Cl}(w_i, X, C) \mid 1 \leq i \leq n\}$,

(3) if $w = \lambda x_{j_1} \ldots x_{j_n}.u$ then $\mathrm{Cl}(w, X, C) = \mathrm{Cl}(u\{c_{j_1}/x_{j_1}, \ldots, c_{j_n}/x_{j_n}\}, X, C)$.

If $u = f(\lambda x_1 x_2 x_3 . x_1 x_3)a$ then its ground closure with respect to $\{c_1, c_2, c_3\}$ is the set of terms $\{u, c_1 c_3, c_3, a\}$. The ground closure of $h(g(h(ha)))$ of Example 3.3 with respect to the empty set is its subterms $\{h(g(h(ha))), g(h(ha)), h(ha), ha, a\}$. An element of a ground closure always has base type; in the case of a right term u of an interpolation (dis)equation, its ground closure with respect to a set of constants consists of all subterms of type $\mathbf{0}$ when free variables x_i are replaced with constants c_i of the same type.

We also identify subterms of left terms v_j of a (dis)equation $x\, v_1 \ldots v_n \approx u$ relative to the finite set C of constants for u. Such a subterm may contain free variables and may have a higher-order type.

Definition 3.5. The *subterms* of w relative to C, written $\mathrm{Sub}(w, C)$, is defined inductively using an auxiliary set $\mathrm{Sub}'(w, C)$:

(1) if w is a variable or a constant then $\mathrm{Sub}(w, C) = \mathrm{Sub}'(w, C) = \{w\}$,
(2) if w is $x\, w_1 \ldots w_n$ then $\mathrm{Sub}(w, C) = \mathrm{Sub}'(w, C) = \{w\} \cup \bigcup \{\mathrm{Sub}(w_i, C) \,|\, 1 \le i \le n\}$,
(3) if w is $f\, w_1 \ldots w_n$ then $\mathrm{Sub}(w, C) = \mathrm{Sub}'(w, C) = \{w\} \cup \bigcup \{\mathrm{Sub}'(w_i, C) \,|\, 1 \le i \le n\}$,
(4) if w is $\lambda y_1 \ldots y_n . v$ then $\mathrm{Sub}(w, C) = \{w\} \cup \mathrm{Sub}(v, C)$ and $\mathrm{Sub}'(w, C) = \bigcup \{\mathrm{Sub}(v\{c_{i_1}/y_1, \ldots, c_{i_n}/y_n\}, C) \,|\, c_{i_j} \in C$ has the same type as $y_j\}$.

The subterms of $\lambda z.z$ relative to $\{c_1, c_2, c_3\}$) of Example 3.2 is $\{\lambda z.z, z\}$. If instead of $\lambda z.z$ the left term of this example is $v = \lambda z.f(\lambda z_1 z_2 z_3 . z_1 z_2)z$, then $\mathrm{Sub}(v, \{c_1, c_2, c_3\})$ is $\{v, f(\lambda z_1 z_2 z_3 . z_1 z_2)z, c_1 c_2, c_1 c_3, c_2, c_3, z\}$: bound variables directly beneath a constant are replaced in their body by constants in C with the same type.

Given the problem P with (dis)equations $x\, v_1^i \ldots v_n^i \approx_i u_i$, $i : 1 \le i \le m$, for each i let X_i be the (possibly empty) set of bound variables in u_i and C_i be a corresponding set of new constants (that do not occur in P), the *forbidden* constants. We are interested in closed terms t in normal form where $t \models P$ and t does not contain forbidden constants.

Definition 3.6. Assume P is the fixed dual interpolation problem of type A.

(1) T is the set of *subtypes* of A including A and the subtypes of subterms of u_i.
(2) For each i, the *right subterms* are $\mathsf{R}_i = \mathrm{Cl}(u_i, X_i, C_i)$ and $\mathsf{R} = \bigcup \{\mathsf{R}_i \,|\, 1 \le i \le m\}$.
(3) For each i, the *left subterms* are $\mathsf{L}_i = C_i \cup \bigcup \{\mathrm{Sub}(v_j^i, C_i) \,|\, 1 \le j \le n\}$ and $\mathsf{L} = \bigcup \{\mathsf{L}_i \,|\, 1 \le i \le m\}$.
(4) The *arity* of P, α, is the largest width k of any type $(A_1, \ldots, A_k, \mathbf{0}) \in \mathsf{T}$.

Clearly, the sets T, R and L are each finite and α is bounded with respect to a given problem P. In Example 3.1, the set of forbidden constants $C_1 \cup C_2$ is empty and R is $\{f(fa), fa, a\}$, L_1 is $\{\lambda y_1 y_2 . y_1 y_2, y_1 y_2, y_2\}$ and its arity is 2.

Definition 3.7. The *right size*, $\delta(u)$, of a right term u relative to its constants C is defined inductively:

(1) if $u = a : \mathbf{0}$ then $\delta(u) = 0$,
(2) if $u = f\, w_1 \ldots w_k$ then $\delta(u) = 1 + \sum \{\delta(w_i) \,|\, 1 \le i \le k\}$,
(3) if $u = \lambda x_{i_1} \ldots x_{i_k} . w$, then $\delta(u) = \delta(w\{c_{i_1}/x_{i_1}, \ldots, c_{i_k}/x_{i_k}\})$.

Definition 3.8. The *right size* for P, δ, is $\sum \{\delta(u_i) \,|\, 1 \le i \le m\}$ of its right terms.

For instance, $\delta(h(g(h(ha)))) = 4$ and $\delta(f(\lambda x_1 x_2 x_3 . x_1 x_3)a) = 2$. If the right size δ for P is 0, then each (dis)equation in P has restricted form $x\, v_1 \ldots v_n \approx a$ where $a : \mathbf{0}$. Padovani proved that dual interpolation is decidable for this special case, the *atoms* case, by showing decidability of observational equivalence within minimal models [12].

4. Tree-checking games

We introduce a game-theoretic characterisation of dual interpolation inspired by model-checking games (such as in [19]) where a model, a transition graph, is traversed relative to a property and players make choices at appropriate positions. Similarly, in the following game the model is a putative solution term t that is traversed relative to the dual interpolation problem. The central motivation is to model the dynamics, β-reduction, without changing t by substituting into it. Because of binding play may jump around t.

A potential solution term t for P has the right type, is in normal form, is well-named (with variables that are disjoint from variables in P) and does not contain forbidden constants. Term t is represented as a tree, tree(t). If t is $y : \mathbf{0}$ or $a : \mathbf{0}$ then tree(t) is the single node labelled with t. In the case of $u\,v_1 \ldots v_k$ when u is a variable or a constant, we assume that a dummy lambda with the empty sequence of variables is placed directly above any subterm $v_i : \mathbf{0}$ in its tree representation. With this understanding, if t is $u\,v_1 \ldots v_k$ then tree(t) consists of the root node labelled u and k-successors, tree(v_1),...,tree(v_k). We use the notation $t \downarrow_i t'$ to represent that tree t' is the ith successor of the root node of t. We also use the standard abbreviation $\lambda \overline{y}$ for $\lambda y_1 \ldots y_n$ for some $n \geq 0$, so $\overline{y}$ is possibly the empty sequence of variables. If t is $\lambda \overline{y}.v$ then tree(t) consists of the root node labelled $\lambda \overline{y}$ and a single successor tree(v), so $t \downarrow_1$ tree(v).

For ease of exposition, we allow t to range over lambda terms, their trees and their root nodes: the context will make it clear which is meant. The introduction of dummy lambdas is a slight extension to η-long normal form; they make term trees more homogeneous, allow for an easier analysis of game playing and, as we shall see in later sections, they are useful for individuating regions of a term and for defining region transformations.

Example 4.1. A solution t from [2] for the problem of Example 3.1 is the following term $\lambda z.z(\lambda x.f(z(\lambda u.x)b))(z(\lambda y.z(\lambda s.s)y)a)$. For instance, if $v = \lambda y_1 y_2.y_1 y_2$ then the normal form of tv is fa.

$$
\begin{aligned}
t\,v \;\; &=_\beta \;\; v(\lambda x.f(v(\lambda u.x)b))(v(\lambda y.v(\lambda s.s)y)a) \\
&=_\beta \;\; \lambda x.f(v(\lambda u.x)b)(v(\lambda y.v(\lambda s.s)y)a) \\
&=_\beta \;\; f(v(\lambda u.v(\lambda y.v(\lambda s.s)ya))b) \\
&=_\beta \;\; f(\lambda u.(v(\lambda y.v(\lambda s.s)y)a)b) \\
&=_\beta \;\; f(v(\lambda y.v(\lambda s.s)y)a) \\
&=_\beta \;\; f(\lambda y.(v(\lambda s.s)y)a) \\
&=_\beta \;\; f(v(\lambda s.s)a) \\
&=_\beta \;\; f((\lambda s.s)a) \\
&=_\beta \;\; fa
\end{aligned}
$$

The tree for t (without indices on edges) is depicted in Figure 1. For instance, in this tree (6) $\downarrow_1$ (7) and (6) $\downarrow_2$ (9). Each node (which we have uniquely identified with a natural number) is labelled with a $\lambda \overline{z}$, a variable or a constant. A node labelled with a constant or variable of type $\mathbf{0}$ is a leaf of the tree (such as node (10)). A node labelled with a higher-order constant or variable of type $(B_1, \ldots, B_k, \mathbf{0})$ has precisely k-successor nodes each labelled with a $\lambda \overline{y}$ (which may be dummy). A node labelled with a $\lambda \overline{z}$ has a single successor which is labelled with a constant or a variable. Therefore, every even level of the tree (when the root is at level 0) is a lambda node. $\square$

Innocent game semantics following Ong in [11] provides a possible game-theoretic foundation for (dual) interpolation. Given a potential solution term t and a (dis)equation

$(1)\lambda z$

$(2)z$

$(3)\lambda x$

$(11)\lambda$

$(4)f$

$(12)z$

$(5)\lambda$

$(13)\lambda y$

$(19)\lambda$

$(6)z$

$(14)z$

$(20)a$

$(7)\lambda u$

$(9)\lambda$

$(15)\lambda s$

$(17)\lambda$
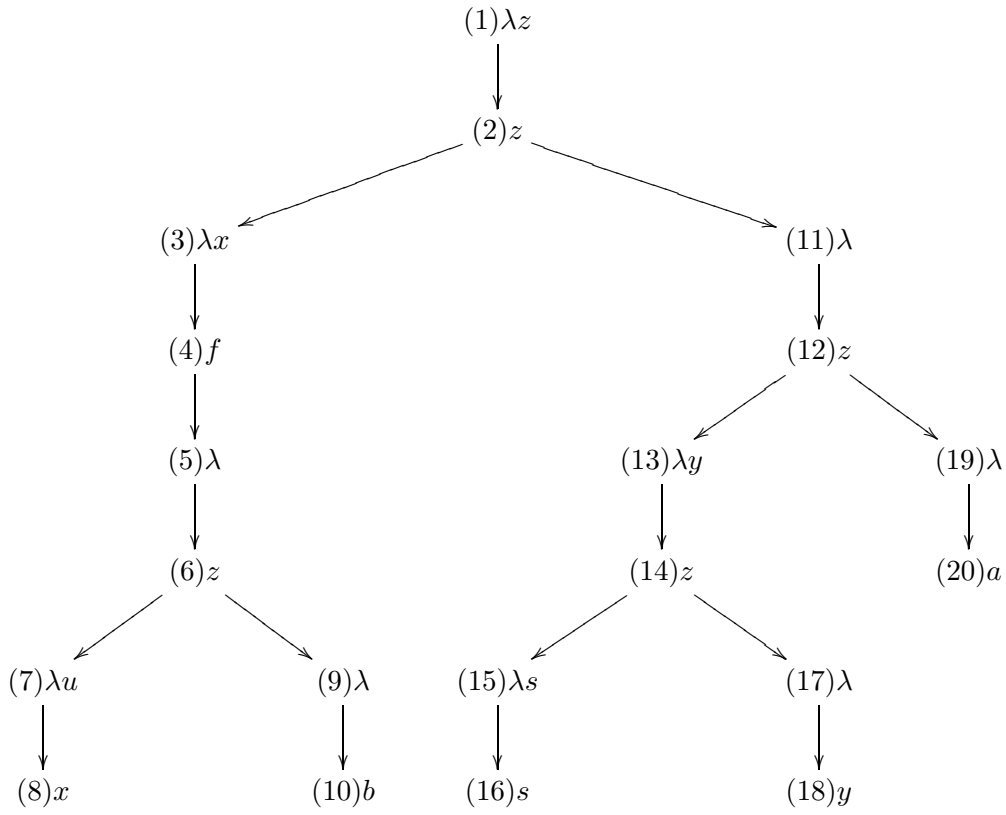
$(8)x$

$(10)b$

$(16)s$

$(18)y$

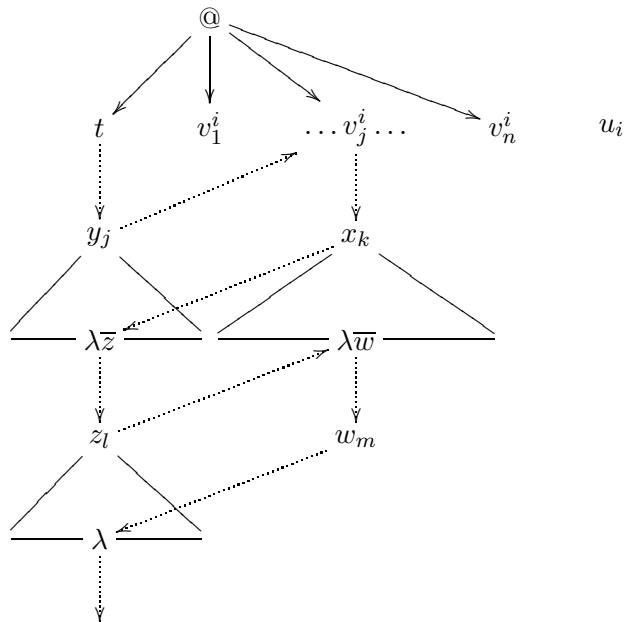Figure 1: A term tree that solves Example 3.1

Figure 2: Illustrating game semantics

$xv_1^i \ldots v_n^i \approx_i u_i$ there is the game board in Figure 2. Player Opponent chooses a branch of u_i. Then, there is a finite play that starts at the root of t and may repeatedly jump in and out of t and in and out of the v_j^i's. At a constant $a : \mathbf{0}$ play ends. At other constants f, player Proponent tries to match Opponent's choice of branch. Proponent wins, when the play finishes, if the sequence of constants encountered matches the branch chosen by Opponent (assuming a mechanism for forbidden constants). Play, for example, may reach y_j in t and then jump to v_j^i, as it is this subtree that is applied to $\lambda\overline{y}$ at the root of t, and then when at x_k in v_j^i play may return to t to an immediate successor of y_j labelled $\lambda\overline{z}$; play then may proceed to z_l and return to a successor of x_k in v_j^i, and so on. Game semantics models β-reduction on the fixed structure of Figure 2 without changing it using substitution. This is the rationale for the tree-checking game[1]. However, the game that we now define starts from the assumption that only t is the common structure for the problem P. Moreover, in later sections, we shall define transformations on t justified by game playing which introduces an asymmetry between it and the argument terms v_j^i of P which are fixed. Consequently, we insist that play is always in the term t. Jumping in and out of the v_j^i's is coded using states, as play traverses t. The game also avoids the justification pointers of game semantics by appealing to iteratively defined look-up tables.

The tree-checking game $\mathsf{G}(t, P)$ is played by one participant, player $\forall$, the *refuter* who attempts to show that t is not a solution of P. It appeals to a finite set of states involving elements of L and R from Definition 3.6: L are the subterms of the v_j^i's and R are those of the u_i's, both modulo the forbidden constants. There are four kinds of state, as follows.

- An *argument* state has the form $q[(l_1, \ldots, l_k), r]$ where each $l_j \in \mathsf{L}$ (and k can be 0) and $r \in \mathsf{R}$. Such a state will only occur at a node in t labelled $\lambda z_1 \ldots z_k$ where each l_j has the same type as z_j: $l_1 \ldots l_k$ are the subterms that are applied to the subterm rooted at $\lambda z_1 \ldots z_k$. A state $q[(\,\,), r]$ occurs at a node of t labelled with a dummy lambda.
- A *value* state has the form $q[l, r]$ where $l \in \mathsf{L}$ and $r \in \mathsf{R}$. This state can only occur at a node of t labelled with a variable y which has the same type as l: l is the subterm of some v_j^i that play at y would jump to in game semantics.
- An *empty* state has the form $q[-, r]$ where $r \in \mathsf{R}$ and can only occur at a node of t labelled with a higher-order constant $f : (B_1, \ldots, B_k, \mathbf{0})$ when r has the form $f u_1 \ldots u_k$.
- A *final* state is either $q[\forall]$, winning for the refuter, or $q[\exists]$, losing for the refuter.

As play traverses t, there are two kinds of free variables: those in the current subtree of t (such as y_j in Figure 2) and those in the left terms of a current argument or value state $q[(l_1, \ldots, l_k), r]$ or $q[l, r]$ (such as x_k in Figure 2). A free variable in a subtree of t is associated with a single left subterm (an element of L) and a free variable in a left subterm l of an argument or value state is associated with a unique subtree of t. So, the game appeals to look-up tables or nested environments $\theta \in \Theta_k$ and $\xi \in \Xi_k$ at a position $k \geq 1$ that are defined iteratively.

Definition 4.2. The sets of partial mappings Θ_k and Ξ_k are defined iteratively as follows.
(1) $\Theta_1 = \{\theta_1\}$ and $\Xi_1 = \{\xi_1\}$ where both θ_1 and ξ_1 are empty (that is, have no entries).
(2) For $k > 1$, $\theta \in \Theta_k$ iff θ is a partial map from variables that are labels of nodes in t to triples $l\xi j$ where $l \in \mathsf{L}$, $j < k$ and $\xi \in \Xi_j$.

---

[1] I am indebted to Luke Ong for pointing out there is a close formal relationship between the tree-checking game and game semantics.

(3) For $k > 1$, $\xi \in \Xi_k$ iff ξ is a partial map from variables that can occur in terms of L to triples $t'\theta j$ where t' is a subtree of t, $j < k$ and $\theta \in \Theta_j$.

A variable y in t may be associated with a left subterm $l \in \mathsf{L}$ which contains free variables: hence, the need for $\theta(y)$ to be a triple $l\xi j$ as ξ records the values of the free variables in l as determined at the earlier position j: we include the position j which is crucial to the understanding of game playing in later sections, when relationships between positions are analysed. Similarly, a variable z in a left subterm may be associated with a subtree of t which contains free variables; so, $\xi(z) = t'\theta j$ where θ has entries for the free variables in t' at the earlier position j. Initially, at the beginning of play when there are no free variables and no previous moves, $\theta \in \Theta_1$ and $\xi \in \Xi_1$ are both empty. The look-up tables play the same role as environments for abstract machines of the lambda calculus (such as a Krivine machine). So the game will simulate the evaluation of a branch of the normal form in the same way abstract machines compute closures. However, unlike these abstract machines, the game here essentially depends on η-long normal forms.

Definition 4.3. A *play* of $\mathsf{G}(t, P)$ is a sequence of positions $t_1 q_1 \theta_1 \xi_1, \ldots, t_n q_n \theta_n \xi_n$ where
(1) each t_i is a node of t and t_1 is the root node of t,
(2) each q_i is a state, q_n is a final state and q_1 is decided as follows: $\forall$ chooses a (dis)equation $x v_1^i \ldots v_n^i \approx_i u_i$ in P and $q_1 = q[(v_1^i, \ldots, v_n^i), u_i]$,
(3) for each i, $\theta_i \in \Theta_i$ and $\xi_i \in \Xi_i$,
(4) position $t_{m+1} q_{m+1} \theta_{m+1} \xi_{m+1}$, $m < n$, is determined by a single move in Figure 3 from position $t_m q_m \theta_m \xi_m$ according to the label at t_m.

The initial position of a play of $\mathsf{G}(t, P)$ is $t_1 q[(l_1, \ldots, l_n), r]\theta_1 \xi_1$ where t_1 is the initial node of t labelled $\lambda y_1 \ldots y_n$ for some $y_1, \ldots, y_n$ and $x l_1 \ldots l_n \approx r$ is a (dis)equation in P. At this position we are interested whether $(t_1 \theta_1)(l_1 \xi_1) \ldots (l_n \xi_1) =_\beta r$ where the look-up tables are viewed as substitutions: initially, they are empty because there are no free variables in the terms $t, l_1, \ldots, l_n$. The initial position is the same as an initial position in game semantics, except the terms l_j and r in the game here are part of the state (and the choice of branch in r will take place as play proceeds).

The possible moves from position m to $m + 1$ in a play are listed in Figure 3 and are divided into three groups that depend on the label at t_m of position m. Group A covers the case when it is a $\lambda \overline{y}$, group B a constant f (whose type is not $\mathbf{0}$) and group C a variable y. For look-up tables θ_{m+1} and ξ_{m+1}, we assume that the substitution notation also stands for function updating: $\mu' = \mu\{\nu_1/y_1, \ldots, \nu_k/y_k\}$ means that the entries in μ' are the same as in μ except for the y_i's, as $\mu'(y_i) = \nu_i$ for each y_i, $1 \le i \le k$. This notation is permitted even if $k = 0$.

Consider group A moves when position m is at the node t_m labelled $\lambda y_1 \ldots y_j$, $j \ge 0$, so q_m has the form $q[(l_1, \ldots, l_j), r]$ where each l_i has the same type as y_i. (If $j = 0$ then the position is at a dummy lambda.) The "interpretation" of position m in terms of β-reduction (see the proof of Theorem 5.7) is whether $(t_m \theta_m)(l_1 \xi_m) \ldots (l_j \xi_m) =_\beta r$. Node t_m has a single successor t_{m+1} and so play descends to it. However, the subtree at t_{m+1} may contain free occurrences of the y_i's (when $j > 0$): the interpretation of each such occurrence is $l_i \xi_m$ because $t_m = \lambda y_1 \ldots y_j.t_{m+1}$ and so θ_{m+1} is an updated version of θ_m reflecting this association of the y_i's with the l_i's and the look-up table ξ_m at position m. Move A1 is when t_{m+1} is labelled with $a : \mathbf{0}$. We can now immediately decide whether $(t_m \theta_m)(l_1 \xi_m) \ldots (l_j \xi_m) =_\beta r$ just by comparing r and a; so position $m + 1$ is final (and won by the refuter if r is different from a). For move A2, if t_{m+1} is labelled with $f : (B_1, \ldots, B_k, \mathbf{0})$ and r does not have the

A. t_m is labelled $\lambda y_1 \ldots y_j$ and $j \geq 0$. Assume $q_m = q[(l_1, \ldots, l_j), r]$.
   Then, $t_{m+1} = t'$ such that $t_m \downarrow_1 t'$, $\theta_{m+1} = \theta_m\{l_1\xi_m m/y_1, \ldots, l_j\xi_m m/y_j\}$ and q_{m+1},
   ξ_{m+1} are defined by cases on the label at t_{m+1}.
   1. $a : \mathbf{0}$. Then, $\xi_{m+1} = \xi_m$. If $r = a$ then $q_{m+1} = q[\exists]$ else $q_{m+1} = q[\forall]$.
   2. $f : (B_1, \ldots, B_k, \mathbf{0})$. Then, $\xi_{m+1} = \xi_m$. If $r = fs_1 \ldots s_k$ then $q_{m+1} = q[-, r]$ else
      $q_{m+1} = q[\forall]$.
   3. $y : B$. If $\theta_{m+1}(y) = l\xi i$, then $\xi_{m+1} = \xi$ and $q_{m+1} = q[l, r]$.
B. t_m is labelled $f : (B_1, \ldots, B_k, \mathbf{0})$. Assume $q_m = q[-, fs_1 \ldots s_k]$.
   1. Then, $\theta_{m+1} = \theta_m$, $\xi_{m+1} = \xi_m$ and $\forall$ chooses $d : 1 \leq d \leq k$ and $t_{m+1} = t'$ such that
      $t_m \downarrow_d t'$ and q_{m+1} is by cases on s_d.
      $s_d : \mathbf{0}$. Then $q_{m+1} = q[(\ ), s_d]$.
      $s_d = \lambda x_{i_1} \ldots x_{i_n}.s$. Then $q_{m+1} = q[(c_{i_1}, \ldots, c_{i_n}), s\{c_{i_1}/x_{i_1}, \ldots, c_{i_n}/x_{i_n}\}]$.
C. t_m is labelled y. Assume $q_m = q[l, r]$.
   If $l : \mathbf{0}$ then $\xi_{m+1} = \xi_m$ else $l = \lambda z_1 \ldots z_j.w$ and $t_m \downarrow_i t'_i$, $1 \leq i \leq j$, and $\xi_{m+1} = $
   $\xi_m\{t'_1\theta_m m/z_1, \ldots, t'_j\theta_m m/z_j\}$. Elements t_{m+1}, q_{m+1} and θ_{m+1} are by cases on l.
   1. a or $\lambda z_1 \ldots z_j.a$ where $a : \mathbf{0}$. Then, $t_{m+1} = t_m$ and $\theta_{m+1} = \theta_m$.
      If $r = a$ then $q_{m+1} = q[\exists]$ else $q_{m+1} = q[\forall]$.
   2. $c : (B_1, \ldots, B_k, \mathbf{0})$. Then $\theta_{m+1} = \theta_m$.
      If $r \neq cs_1 \ldots s_k$ then $t_{m+1} = t_m$ and $q_{m+1} = q[\forall]$ else $r = cs_1 \ldots s_k$ and $\forall$ chooses
      $d : 1 \leq d \leq k$, and $t_{m+1} = t'$ such that $t_m \downarrow_d t'$ and q_{m+1} is by cases on s_d.
      $s_d : \mathbf{0}$. Then $q_{m+1} = q[(\ ), s_d]$.
      $s_d = \lambda x_{i_1} \ldots x_{i_n}.s$. Then $q_{m+1} = q[(c_{i_1}, \ldots, c_{i_n}), s\{c_{i_1}/x_{i_1}, \ldots, c_{i_n}/x_{i_n}\}]$.
   3. $fw_1 \ldots w_k$ or $\lambda z_1 \ldots z_j.fw_1 \ldots w_k$. Then $t_{m+1} = t_m$ and $\theta_{m+1} = \theta_m$.
      If $r \neq fs_1 \ldots s_k$, then $q_{m+1} = q[\forall]$ else $r = fs_1 \ldots s_k$ and $\forall$ chooses $d : 1 \leq d \leq k$
      and q_{m+1} is by cases on s_d.
      $s_d : \mathbf{0}$. Then $q_{m+1} = q[w_d, s_d]$.
      $s_d = \lambda x_{i_1} \ldots x_{i_n}.s$ and $w_d = \lambda y_1 \ldots y_n.w'$.   Then $q_{m+1} = $
      $q[w'\{c_{i_1}/y_1, \ldots, c_{i_n}/y_n\}, s\{c_{i_1}/x_{i_1}, \ldots, c_{i_n}/x_{i_n}\}]$.
   4. $x\, l_1 \ldots l_k$ or $\lambda z_1 \ldots z_j.x\, l_1 \ldots l_k$ where $k \geq 0$.
      If $\xi_{m+1}(x) = t'\theta i$ then $t_{m+1} = t'$, $\theta_{m+1} = \theta$ and $q_{m+1} = q[(l_1, \ldots, l_k), r]$.

Figure 3: Game moves

form $fs_1 \ldots s_k$ then $q_{m+1} = q[\forall]$ as we now know that $(t_m\theta_m)(l_1\xi_m) \ldots (l_j\xi_m) \neq_\beta r$. If
$r = fs_1 \ldots s_k$ then position $m + 1$ is $t_{m+1}q[-, r]\theta_{m+1}\xi_{m+1}$ where $\xi_{m+1} = \xi_m$. Move A3 is
when t_{m+1} is labelled with a variable y. Thus, y is a free variable occurrence in the tree
t_{m+1} whose interpretation is $\theta_{m+1}(y) = l\xi i$ decided at the earlier position $i \leq m$. So, ξ_{m+1}
is set to ξ (as it interprets the free variables in l in $q_{m+1} = q[l, r]$).
   The B move covers the case when t_m is labelled with a constant $f : (B_1, \ldots, B_k, \mathbf{0})$;
because of move A2, we only need to consider it when the state is $q_m = q[-, fs_1 \ldots s_k]$ for
some $s_1, \ldots, s_k$. The "interpretation" of such a position m is whether $(t_m\theta_m) =_\beta fs_1 \ldots s_k$.
In which case $t_m = ft'_1 \ldots t'_k$. So, $(t_m\theta_m) =_\beta fs_1 \ldots s_k$ if, and only if, for each $d : 1 \leq d \leq k$,
$t'_d\theta_m =_\beta s_d$. The refuter $\forall$ chooses such a d. The delicacy is that s_d may be of higher type, of
the form $\lambda x_{i_1} \ldots x_{i_n}.s$ and so, therefore, t'_d is labelled $\lambda y'_1 \ldots y'_n$ for some $y'_1, \ldots y'_n$ because it
has the same type: position $m+1$ is then $t'_d q[(c_{i_1}, \ldots, c_{i_n}), s\{c_{i_1}/x_{i_1}, \ldots, c_{i_n}/x_{i_n}\}]\theta_{m+1}\xi_{m+1}$
where $\theta_{m+1} = \theta_m$, and $\xi_{m+1} = \xi_m$: here we are making use of the forbidden constants c_{i_j}.
If s_d has ground type then $q_{m+1} = q[(\ ), s_d]$ (as the label at t'_d is a dummy lambda).

Group C moves cover the case where t_m is labelled with a variable y. The "interpretation" of position m, $t_m q[l, r] \theta_m \xi_m$, is whether $(l\xi_m)(t'_1\theta_m) \ldots (t'_j\theta_m) =_\beta r$ where for $j \geq 0$, $t_m \downarrow_i t'_i$ when $1 \leq i \leq j$. If $j > 0$ then $l = \lambda z_1 \ldots z_j.w$ for some $z_1, \ldots, z_j$ and the free occurrences of z_i in w are associated with $t'_i\theta_m$ at position m; so, ξ_{m+1} is an updated version of ξ_m reflecting this association. When $j = 0$, $l = w$ and the interpretation is whether $l\xi_m =_\beta r$. For both cases $l = \lambda z_1 \ldots z_j.w$ and $l = w$ play proceeds by examining the "head" of w. Move C1 covers the case where it is $a : \mathbf{0}$ (which is possibly a forbidden constant). We can now immediately decide whether $(l\xi_m)(t'_1\theta_m) \ldots (t'_j\theta_m) =_\beta r$ just by comparing l and r. Move C2 covers the case where l is a forbidden constant c at higher type $(B_1, \ldots, B_j, \mathbf{0})$. If $r \neq cs_1 \ldots s_j$ then we know that $c(t'_1\theta_m) \ldots (t'_j\theta_m) \neq_\beta r$ and, so $q_{m+1} = q[\forall]$. If $r = cs_1 \ldots s_j$ then $c(t'_1\theta_m) \ldots (t'_j\theta_m) =_\beta r$ if, and only if, for each d, $(t'_d\theta_m) =_\beta s_d$. As with B1, the refuter $\forall$ chooses such a d. Again we need to specially deal with the case that s_d has higher type. Move C3 deals with the case that w is $fw_1 \ldots w_k$. Again, if r does not have the form $fs_1 \ldots s_k$ then $q_{m+1} = q[\forall]$ as we now know that $(l\xi_m)(t'_1\theta_m) \ldots (t'_j\theta_m) \neq_\beta r$. If $r = fs_1 \ldots s_k$ then $(l\xi_m)(t'_1\theta_m) \ldots (t'_j\theta_m) =_\beta r$ if, and only if, for each d, $(w_d\xi_{m+1}) =_\beta s_d$. Therefore, $\forall$ chooses such a d. For the next position, again we need to deal with the possibility that s_d is of higher type: if $s_d = \lambda x_{i_1} \ldots x_{i_n}.s$ then w_d which has the same type must be of the form $\lambda y_1 \ldots y_n.w'$ for some $y_1, \ldots, y_n$; so we must substitute the same forbidden constant for each x_{i_j} and y_j; so q_{m+1} is $q[w'\{c_{i_1}/y_1, \ldots, c_{i_n}/y_n\}, s\{c_{i_1}/x_{i_1}, \ldots, c_{i_n}/x_{i_n}\}]$. If $s_d : \mathbf{0}$ then the next position $m + 1$ is $t_{m+1}q[w_d, s_d]\theta_{m+1}\xi_{m+1}$ where $t_{m+1} = t_m$ and $\theta_{m+1} = \theta_m$. Move C4 covers the case when $w = xl_1 \ldots l_k$, $k \geq 0$. Therefore, x is a free variable in l whose interpretation $\xi_{m+1}(x) = t'\theta i$ which was determined at the earlier position $i \leq m$; so, x is associated with the subtree t' of t and play therefore jumps to it.

$\forall$ can exercise choice, by carving out a branch of a right term of a (dis)equation modulo forbidden constants, with moves B1, C2 and C3. The look-up tables are used in earnest with moves A3 and C4 to interpret the two kinds of free variable. Move C4 allows play to jump elsewhere in the term tree (always to a node labelled with a lambda): it also opens the possibility that play can repeatedly be at the same node of t. With moves A1–A3, B1 and C2 (unless play finishes) control passes down the term tree while it remains stationary in the case of C1 and C3.

Definition 4.4. If $t_1q_1\theta_1\xi_1, \ldots, t_nq_n\theta_n\xi_n$ is a play of $\mathsf{G}(t, P)$ then player $\forall$ *wins the play* if the final state is $q[\forall]$ and she *loses* it otherwise (if the final state is $q[\exists]$).

Example 4.5. Let P be the problem $x(\lambda z.z) = u$ where $u = f(\lambda x_1 x_2 x_3.x_1 x_3)a$ of Example 3.2. The bound variables in the right term u are $\{x_1, x_2, x_3\}$; so, assume corresponding forbidden constants $C = \{c_1, c_2, c_3\}$. Let $t = \lambda y.y(y(f(\lambda x z_1 z_2.x(yz_2))(ya)))$: tree$(t)$ is depicted in Figure 4. $\mathsf{G}(t, p)$ consists of two plays that descend t. (For 3rd-order problems P play cannot jump around a term tree, as we observe in Section 7.) Both plays start as follows where we have supplied which move is applied to produce the next position: the initial component of each position is a node of t.

$$(1)\, q[(\lambda z.z), u]\, \theta_1\, \xi_1$$
$$(2)\, q[\lambda z.z, u]\, \theta_2\xi_2 \qquad \theta_2 = \theta_1\{(\lambda z.z)\xi_1 1/y\} \qquad \xi_2 = \xi_1 \qquad\qquad A3$$
$$(3)\, q[(\ ), u]\, \theta_3\xi_3 \qquad \theta_3 = \theta_2 \qquad\qquad\qquad\qquad \xi_3 = \xi_1\{(3)\theta_2 2/z\} \quad C4$$
$$(4)\, q[\lambda z.z, u]\, \theta_4\xi_4 \qquad \theta_4 = \theta_2 \qquad\qquad\qquad\qquad \xi_4 = \xi_1 \qquad\qquad A3$$
$$(5)\, q[(\ ), u]\, \theta_5\, \xi_5 \qquad \theta_5 = \theta_2 \qquad\qquad\qquad\qquad \xi_5 = \xi_1\{(5)\theta_2 4/z\} \quad C4$$
$$(6)\, q[-, u]\, \theta_6\, \xi_6 \qquad \theta_6 = \theta_2 \qquad\qquad\qquad\qquad \xi_6 = \xi_5 \qquad\qquad A2$$

$$(1)\lambda y$$

$$\downarrow$$

$$(2)y$$

$$\downarrow$$

$$(3)\lambda$$

$$\downarrow$$

$$(4)y$$

$$\downarrow$$

$$(5)\lambda$$

$$\downarrow$$

$$(6)f$$

$(7)\lambda x z_1 z_2$ $\qquad\qquad$ $(13)\lambda$

$(8)x$ $\qquad\qquad\qquad$ $(14)y$

$(9)\lambda$ $\qquad\qquad\qquad$ $(15)\lambda$

$(10)y$ $\qquad\qquad\qquad$ $(16)a$
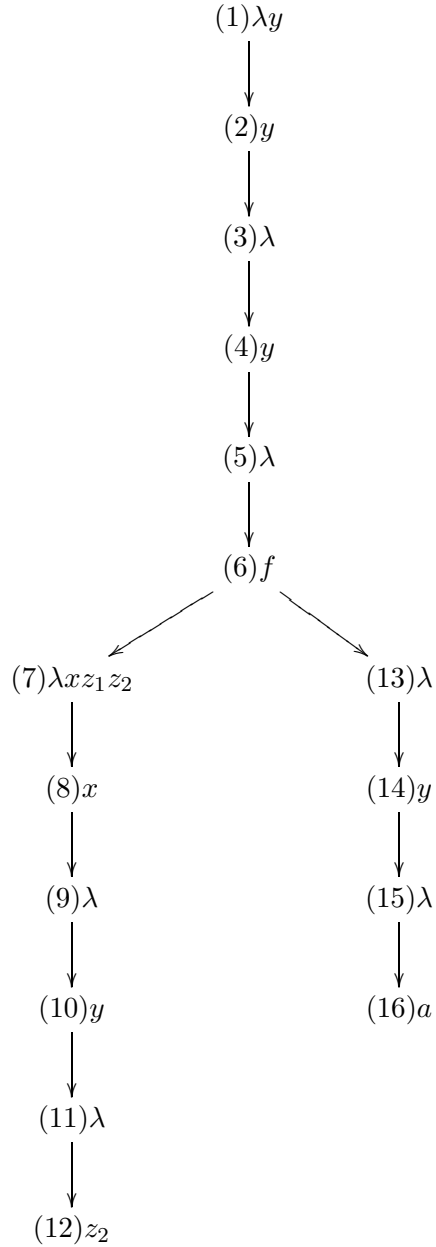
$(11)\lambda$

$(12)z_2$

Figure 4: A solution term with order 3 for Example 4.5

The initial state is an argument state $q[(\lambda z.z), u]$ and control is at node (1). Play descends from node (1) to (2) calling the value $\lambda z.z$ by A3. Next, by C4, because z is the head variable in the body of $\lambda z.z$, has no arguments and is associated with node (3), the next state is the argument state $q[(\ ), u]$ and control is at (3). Play descends from (3) to (4) calling the value $\lambda z.z$ by A3. Again by C4, z is the head variable in the body of $\lambda z.z$, has no arguments and is now associated with (5), the next state is the argument state $q[(\ ), u]$ and control is at (5). A2 is then applied (because the right term u in the state has f as

$(1)\, q[(v), fa]\, \theta_1\, \xi_1$			
$(2)\, q[v, fa]\, \theta_2 \xi_2$	$\theta_2 = \theta_1\{v\xi_1 1/z\}$	$\xi_2 = \xi_1$	$A3$
$(3)\, q[(y_2), fa]\, \theta_3 \xi_3$	$\theta_3 = \theta_2$	$\xi_3 = \xi_1\{(3)\theta_2 2/y_1, (11)\theta_2 2/y_2\}$	$C4$
$(4)\, q[-, fa]\, \theta_4 \xi_4$	$\theta_4 = \theta_2\{y_2\xi_3 3/x\}$	$\xi_4 = \xi_3$	$A2$
$(5)\, q[(\ ), a]\, \theta_5\, \xi_5$	$\theta_5 = \theta_4$	$\xi_5 = \xi_4$	$B1$
$(6)\, q[v, a]\, \theta_6\, \xi_6$	$\theta_6 = \theta_4$	$\xi_6 = \xi_1$	$A3$
$(7)\, q[(y_2), a]\, \theta_7\, \xi_7$	$\theta_7 = \theta_4$	$\xi_7 = \xi_1\{(7)\theta_4 6/y_1, (9)\theta_4 6/y_2\}$	$C4$
$(8)\, q[y_2, a]\, \theta_8\, \xi_8$	$\theta_8 = \theta_4\{y_2\xi_7 7/u\}$	$\xi_8 = \xi_3$	$A3$
$(11)\, q[(\ ), a]\, \theta_9\, \xi_9$	$\theta_9 = \theta_2$	$\xi_9 = \xi_3$	$C4$
$(12)\, q[v, a]\, \theta_{10}\xi_{10}$	$\theta_{10} = \theta_2$	$\xi_{10} = \xi_1$	$A3$
$(13)\, q[(y_2), a]\, \theta_{11}\xi_{11}$	$\theta_{11} = \theta_2$	$\xi_{11} = \xi_1\{(13)\theta_2 10/y_1, (19)\theta_2 10/y_2\}$	$C4$
$(14)\, q[v, a]\, \theta_{12}\xi_{12}$	$\theta_{12} = \theta_2\{y_2\xi_{11} 11/y\}$	$\xi_{12} = \xi_1$	$A3$
$(15)\, q[(y_2), a]\, \theta_{13}\, \xi_{13}$	$\theta_{13} = \theta_{12}$	$\xi_{13} = \xi_1\{(15)\theta_{12} 12/y_1, (17)\theta_{12} 12/y_2\}$	$C4$
$(16)\, q[y_2, a]\, \theta_{14}\, \xi_{14}$	$\theta_{14} = \theta_{12}\{y_2\xi_{13} 13/s\}$	$\xi_{14} = \xi_{13}$	$A3$
$(17)\, q[(\ ), a]\, \theta_{15}\, \xi_{15}$	$\theta_{15} = \theta_{12}$	$\xi_{15} = \xi_{13}$	$C4$
$(18)\, q[y_2, a]\, \theta_{16}\, \xi_{16}$	$\theta_{16} = \theta_{12}$	$\xi_{16} = \xi_{11}$	$A3$
$(19)\, q[(\ ), a]\, \theta_{17}\, \xi_{17}$	$\theta_{17} = \theta_2$	$\xi_{17} = \xi_{11}$	$C4$
$(20)\, q[\ \exists\ ]\, \theta_{18}\, \xi_{18}$	$\theta_{18} = \theta_2$	$\xi_{18} = \xi_{11}$	$A1$

Figure 5: The play of Example 4.6 on the tree in Figure 1

head constant) and control passes from (5) to (6). Move B1 is now applied and there is a $\forall$ choice as to which branch of u to take. If direction 1 is chosen then play continues as follows.

$(7)\, q[(c_1, c_2, c_3), c_1 c_3]\, \theta_7\, \xi_7$	$\theta_7 = \theta_2$	$\xi_7 = \xi_5$	$B1$
$(8)\, q[c_1, c_1 c_3]\, \theta_8\, \xi_8$	$\theta_8 = \theta_2\{c_1\xi_5 7/x, c_2\xi_5 7/z_1, c_3\xi_5 7/z_2\}$	$\xi_8 = \xi_5$	$A3$
$(9)\, q[(\ ), c_3]\, \theta_9\, \xi_9$	$\theta_9 = \theta_8$	$\xi_9 = \xi_5$	$C2$
$(10)\, q[\lambda z.z, c_3]\, \theta_{10}\xi_{10}$	$\theta_{10} = \theta_8$	$\xi_{10} = \xi_5$	$A3$
$(11)\, q[(\ ), c_3]\, \theta_{11}\, \xi_{11}$	$\theta_{11} = \theta_8$	$\xi_{11} = \xi_5\{(11)\theta_8 10/z\}$	$C4$
$(12)\, q[c_3, c_3]\, \theta_{12}\, \xi_{12}$	$\theta_{12} = \theta_8$	$\xi_{12} = \xi_5$	$A3$
$(12)\, q[\ \exists\ ]\, \theta_{13}\, \xi_{13}$	$\theta_{13} = \theta_8$	$\xi_{13} = \xi_5$	$C1$

Forbidden constants are introduced for replacing x_1, x_2, x_3 in the body of u to give the right term $c_1 c_3$ and as arguments (c_1, c_2, c_3) for the variables x, z_1, z_2 bound at (7); see the updated look-up table θ_8. At (8) the value c_1 is called using move A3 and then by C2, control proceeds to (9) and the right term becomes the argument c_3 of c_1. At (10) the value $\lambda z.z$ is called again by A3 and by C4 as z is the head variable and is associated with node (11) control passes to it (with the empty sequence of arguments). Finally, at (12), the value c_3 is called by A3, and then by C1, $\forall$ loses the play. She also loses if direction 2 is chosen at position 7 as the reader can verify. $\qquad\square$

Example 4.6. Let P be the first equation of Example 3.1 $x\, v = fa$ where $v = \lambda y_1 y_2.y_1 y_2$ (which is a 4th-order problem). The tree t of Figure 1 solves P as was demonstrated in Example 4.1 through β-reduction. The single play in $\mathsf{G}(t, P)$ is presented in Figure 5. The initial state is $q[(v), fa]$ at the root of t. By move A3, play descends to node (2) calling the value v. Next by C4 because y_1 is the head variable in the body of v, is associated with the subtree at (3) and has argument y_2 play moves to (3) with argument state $q[(y_2), fa]$. By A2 play descends to node (4) labelled f and then move B1 is applied without a choice for

$\forall$ because the type of $f : (\mathbf{0}, \mathbf{0})$ has arity one, and so the play descends to node (5) with a change in the right term of the state from fa to a. By move A3 play descends to node (6) calling the value v again. By C4 because the head variable y_1 is associated with the subtree at (7) and has argument y_2 play moves to (7). By move A3, play descends to (8). The entry $\theta_8(x)$ is $y_2\xi_3 3$, so the state is $q[y_2, a]$. By move C4 because $\xi_3(y_2) = (11)\theta_2 2$, that is, y_2 is associated with the subtree rooted at (11), play jumps from node (8) to node (11). If node (8) were labelled u then because the entry $\theta_8(u) = y_2\xi_7 7$ the state would again be $q[y_2, a]$; play would then jump to node (9) because $\xi_7(y_2) = (9)\theta_4 6$. Play descends from node (11) to (12), (13), (14), (15) and (16) and jumps to node (17) and descends to (18) and then jumps to node (19) before descending to node (20) by move A1, where the refuter loses the play.   □

Example 4.7. We now examine the equation of Example 3.3 which illustrates play jumping in more detail and how the game moves, especially A3 and C4, essentially depend on η-long normal forms Let P be the equation

$$x(\lambda y_1 y_2.y_1(\lambda y_3.y_2(y_1(\lambda y_4.y_3)))) = h(g(h(ha)))$$

and let t be $\lambda z.z(\lambda z_1.z(\lambda x_1.z_1(x_1(z_1a)))\lambda x_2.gx_2)\lambda z_2.hz_2$ which is a solution; t as a tree is depicted in Figure 6. The single play for $\mathsf{G}(t, P)$ is presented in Figure 7 where the following abbreviations for left and right subterms are employed.

$$
\begin{aligned}
v &= \lambda y_1 y_2.y_1(\lambda y_3.y_2(y_1(\lambda y_4.y_3))) \\
v_1 &= \lambda y_3.y_2(y_1(\lambda y_4.y_3)) \\
v_2 &= y_1(\lambda y_4.y_3) \\
v_3 &= \lambda y_4.y_3 \\
u &= h(g(h(ha))) \\
u_1 &= g(h(ha)) \\
u_2 &= h(ha)
\end{aligned}
$$

Play starts at node (1) with state $q[(v), u]$ and by A3 descends to (2). The head variable of the body of v is y_1 which has argument v_1 and so play moves to (3) by C4 with state $q[(v_1), u]$. Similarly, it descends to (4) by A3 and then to (5) by C4 with state $q[(v_1), u]$ at which point it moves to (6) which is labelled z_1; $\theta_6(z_1) = v_1\eta_3 3$ and the head variable of the body of v_1 is y_2 and its argument is v_2; so, play jumps to (17) with state $q[(v_2), u]$. It then descends to (18), (19) and (20) with state $q[v_2, u_1]$; the head variable of v_2 is y_1 and it has argument v_3; so play returns to (3) with state $q[(v_3), u_1]$. It then descends to (4) and (5) and moves to (6); the head variable of the body of v_3 is y_3 which is associated with node (7), as $\xi_{14}(y_3) = (7)\theta_6 6$. Play proceeds to (8), jumps to (13), reaches (16) and then returns to (5) and so on. Player $\forall$ eventually loses when play reaches (12) as the reader can check.   □

Definition 4.8. If P is a (dual) interpolation problem then $\forall$ *loses the game* $\mathsf{G}(t, P)$ if and only if

(1) for every equation in P, $\forall$ loses every play whose initial state is given from it,
(2) for each disequation in P, $\forall$ wins some play whose initial state is given from it.

The game *characterises* dual interpolation.

Theorem 4.9. $\forall$ *loses* $\mathsf{G}(t, P)$ *if, and only if,* $t \models P$.

$(1)\lambda z$

$(2)z$

$(3)\lambda z_1$ $(17)\lambda z_2$

$(4)z$ $(18)h$

$(5)\lambda x_1$ $(13)\lambda x_2$ $(19)\lambda$

$(6)z_1$ $(14)g$ $(20)z_2$

$(7)\lambda$ $(15)\lambda$

$(8)x_1$ $(16)x_2$
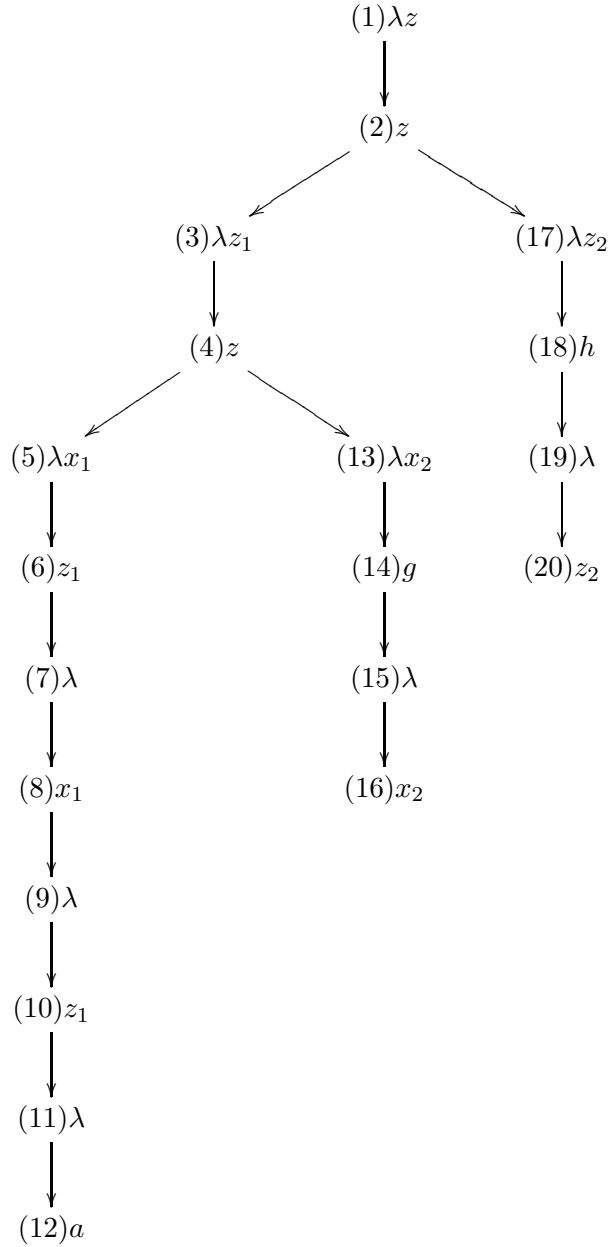
$(9)\lambda$

$(10)z_1$

$(11)\lambda$

$(12)a$

Figure 6: A 5th order term tree for Example 4.7

Proof. The simply typed λ-calculus is strongly normalising and is Church-Rosser modulo α-equivalence. For every term t there is an m such that t reduces to normal form using at most m β-reductions (whatever the reduction strategy). Therefore, for any position $t_i q_i \theta_i \xi_i$ of a play of $\mathsf{G}(t, P)$ we say that it m-holds (m-fails) if $q_i = q[\exists]$ ($q_i = q[\forall]$) and when q_i is not final, by cases on t_i and q_i (and look-up tables become delayed substitutions)

- if $t_i = \lambda\overline{y}$, $q_i = q[(l_1, \ldots, l_k), r]$ and t' is $(t_i\theta_i)l_1\xi_i \ldots l_k\xi_i$ then $t' =_\beta r$ ($t' \neq_\beta r$) and t' reduces to normal form with at most m β-reductions,

$(1)\ q[(v),u]\,\theta_1\,\xi_1$

$(2)\ q[v,u]\,\theta_2\xi_2$	$\theta_2 = \theta_1\{v\xi_1 1/z\}$	$\xi_2 = \xi_1$	$A3$
$(3)\ q[(v_1),u]\,\theta_3\xi_3$	$\theta_3 = \theta_2$	$\xi_3 = \xi_1\{(3)\theta_2 2/y_1,(17)\theta_2 2/y_2\}$	$C4$
$(4)\ q[v,u]\,\theta_4\xi_4$	$\theta_4 = \theta_2\{v_1\xi_3 3/z_1\}$	$\xi_4 = \xi_1$	$A3$
$(5)\ q[(v_1),u]\,\theta_5\,\xi_5$	$\theta_5 = \theta_4$	$\xi_5 = \xi_1\{(5)\theta_4 4/y_1,(13)\theta_4 4/y_2\}$	$C4$
$(6)\ q[v_1,u]\,\theta_6\,\xi_6$	$\theta_6 = \theta_4\{v_1\xi_5 5/x_1\}$	$\xi_6 = \xi_3$	$A3$
$(17)\ q[(v_2),u]\,\theta_7\,\xi_7$	$\theta_7 = \theta_2$	$\xi_7 = \xi_3\{(7)\theta_6 6/y_3\}$	$C4$
$(18)\ q[-,u]\,\theta_8\,\xi_8$	$\theta_8 = \theta_2\{v_2\xi_7 7/z_2\}$	$\xi_8 = \xi_7$	$A2$
$(19)\ q[(\ ),u_1]\,\theta_9\,\xi_9$	$\theta_9 = \theta_8$	$\xi_9 = \xi_7$	$B1$
$(20)\ q[v_2,u_1]\,\theta_{10}\xi_{10}$	$\theta_{10} = \theta_8$	$\xi_{10} = \xi_7$	$A3$
$(3)\ q[(v_3),u_1]\,\theta_{11}\xi_{11}$	$\theta_{11} = \theta_2$	$\xi_{11} = \xi_7$	$C4$
$(4)\ q[v,u_1]\,\theta_{12}\xi_{12}$	$\theta_{12} = \theta_2\{v_3\xi_7 11/z_1\}$	$\xi_{12} = \xi_1$	$A3$
$(5)\ q[(v_1),u_1]\,\theta_{13}\,\xi_{13}$	$\theta_{13} = \theta_{12}$	$\xi_{13} = \xi_1\{(5)\theta_{12}12/y_1,(13)\theta_{12}12/y_2\}$	$C4$
$(6)\ q[v_3,u_1]\,\theta_{14}\,\xi_{14}$	$\theta_{14} = \theta_{12}\{v_1\xi_{13}13/x_1\}$	$\xi_{14} = \xi_7$	$A3$
$(7)\ q[(\ ),u_1]\,\theta_{15}\,\xi_{15}$	$\theta_{15} = \theta_6$	$\xi_{15} = \xi_{14}\{(7)\theta_{14}14/y_4\}$	$C4$
$(8)\ q[v_1,u_1]\,\theta_{16}\,\xi_{16}$	$\theta_{16} = \theta_6$	$\xi_{16} = \xi_5$	$A3$
$(13)\ q[(v_2),u_1]\,\theta_{17}\,\xi_{17}$	$\theta_{17} = \theta_4$	$\xi_{17} = \xi_5\{(9)\theta_6 16/y_3\}$	$C4$
$(14)\ q[-,u_1]\,\theta_{18}\,\xi_{18}$	$\theta_{18} = \theta_4\{v_2\xi_{17}17/x_2\}$	$\xi_{18} = \xi_{17}$	$A2$
$(15)\ q[(\ ),u_2]\,\theta_{19}\,\xi_{19}$	$\theta_{19} = \theta_{18}$	$\xi_{19} = \xi_{17}$	$B1$
$(16)\ q[v_2,u_2]\,\theta_{20}\,\xi_{20}$	$\theta_{20} = \theta_{18}$	$\xi_{20} = \xi_{17}$	$A3$
$(5)\ q[(v_3),u_2]\,\theta_{21}\,\xi_{21}$	$\theta_{21} = \theta_4$	$\xi_{21} = \xi_{17}$	$C4$
$(6)\ q[v_1,u_2]\,\theta_{22}\,\xi_{22}$	$\theta_{22} = \theta_4\{v_3\xi_{17}21/x_1\}$	$\xi_{22} = \xi_3$	$A3$
$(17)\ q[(v_2),u_2]\,\theta_{23}\,\xi_{23}$	$\theta_{23} = \theta_2$	$\xi_{23} = \xi_3\{(7)\theta_{22}22/y_3\}$	$C4$
$(18)\ q[-,u_2]\,\theta_{24}\,\xi_{24}$	$\theta_{24} = \theta_2\{v_2\xi_{23}23/z_2\}$	$\xi_{24} = \xi_{23}$	$A2$
$(19)\ q[(\ ),ha]\,\theta_{25}\,\xi_{25}$	$\theta_{25} = \theta_{24}$	$\xi_{25} = \xi_{23}$	$B1$
$(20)\ q[v_2,ha]\,\theta_{26}\xi_{26}$	$\theta_{26} = \theta_{24}$	$\xi_{26} = \xi_{23}$	$A3$
$(3)\ q[(v_3),ha]\,\theta_{27}\xi_{27}$	$\theta_{27} = \theta_2$	$\xi_{27} = \xi_{23}$	$C4$
$(4)\ q[v,ha]\,\theta_{28}\xi_{28}$	$\theta_{28} = \theta_2\{v_3\xi_{23}27/z_1\}$	$\xi_{28} = \xi_1$	$A3$
$(5)\ q[(v_1),ha]\,\theta_{29}\,\xi_{29}$	$\theta_{29} = \theta_{28}$	$\xi_{29} = \xi_1\{(5)\theta_{28}28/y_1,(13)\theta_{28}28/y_2\}$	$C4$
$(6)\ q[v_3,ha]\,\theta_{30}\,\xi_{30}$	$\theta_{30} = \theta_{28}\{v_1\xi_{29}29/x_1\}$	$\xi_{30} = \xi_{23}$	$A3$
$(7)\ q[(\ ),ha]\,\theta_{31}\,\xi_{31}$	$\theta_{31} = \theta_{22}$	$\xi_{31} = \xi_{23}\{(7)\theta_{30}30/y_4\}$	$C4$
$(8)\ q[v_3,ha]\,\theta_{32}\,\xi_{32}$	$\theta_{32} = \theta_{22}$	$\xi_{32} = \xi_{17}$	$A3$
$(9)\ q[(\ ),ha]\,\theta_{33}\,\xi_{33}$	$\theta_{33} = \theta_6$	$\xi_{33} = \xi_{32}\{(9)\theta_{22}32/y_4\}$	$C4$
$(10)\ q[v_1,ha]\,\theta_{34}\,\xi_{34}$	$\theta_{34} = \theta_6$	$\xi_{34} = \xi_3$	$A3$
$(17)\ q[(v_2),ha]\,\theta_{35}\,\xi_{35}$	$\theta_{35} = \theta_2$	$\xi_{35} = \xi_3\{(11)\theta_6 34/y_3\}$	$C4$
$(18)\ q[-,ha]\,\theta_{36}\,\xi_{36}$	$\theta_{36} = \theta_2\{v_2\xi_{35}35/z_2\}$	$\xi_{36} = \xi_{35}$	$A2$
$(19)\ q[(\ ),a]\,\theta_{37}\,\xi_{37}$	$\theta_{37} = \theta_{36}$	$\xi_{37} = \xi_{35}$	$B1$
$(20)\ q[v_2,a]\,\theta_{38}\xi_{38}$	$\theta_{38} = \theta_{36}$	$\xi_{38} = \xi_{35}$	$A3$
$(3)\ q[(v_3),a]\,\theta_{39}\xi_{39}$	$\theta_{39} = \theta_2$	$\xi_{39} = \xi_{35}$	$C4$
$(4)\ q[v,a]\,\theta_{40}\xi_{40}$	$\theta_{40} = \theta_2\{v_3\xi_{35}39/z_1\}$	$\xi_{40} = \xi_1$	$A3$
$(5)\ q[(v_1),a]\,\theta_{41}\,\xi_{41}$	$\theta_{41} = \theta_{40}$	$\xi_{41} = \xi_1\{(5)\theta_{40}40/y_1,(13)\theta_{40}40/y_2\}$	$C4$
$(6)\ q[v_3,a]\,\theta_{42}\,\xi_{42}$	$\theta_{42} = \theta_{40}\{v_1\xi_{41}41/x_1\}$	$\xi_{42} = \xi_{35}$	$A3$
$(11)\ q[(\ ),a]\,\theta_{43}\,\xi_{43}$	$\theta_{43} = \theta_6$	$\xi_{43} = \xi_{35}\{(7)\theta_{42}42/y_4\}$	$C4$
$(12)\ q[\ \exists\ ]\,\theta_{44}\,\xi_{44}$	$\theta_{44} = \theta_6$	$\xi_{44} = \xi_{43}$	$A1$

Figure 7: A 5th-order play

- if $t_i = f$, $q_i = q[-, r]$ and t' is $t_i\theta_i$ then $t' =_\beta r$ ($t' \neq_\beta r$) and t' reduces to normal form with at most m β-reductions,
- if $t_i = z$, $q_i = q[l, r]$ and $t_i \downarrow_j t'_j$, $1 \le j \le k$ for $k \ge 0$, and t' is $(l\xi_i)t'_1\theta_i \dots t'_k\theta_i$ then $t' =_\beta r$ ($t' \neq_\beta r$) and t' reduces to normal form with at most m β-reductions.

The proof is done by invoking as a measure a pair of integers, first the size of right term in q_i and second the maximal number of β-reductions needed for t' to reduce to normal form; the pair is ordered with the lexicographic ordering. The following properties are easy to show by case analysis.

(1) If $t_iq_i\theta_i\xi_i$ m-holds then $q_i = q[\exists]$ or for any next position $t_{i+1}q_{i+1}\theta_{i+1}\xi_{i+1}$ there is an m' such that it m' holds and either $m' \le m$ or the size of the right term in q_{i+1} is strictly smaller than in q_i.

(2) If $t_iq_i\theta_i\xi_i$ m-fails then $q_i = q[\forall]$ or there is an m' and a next position $t_{i+1}q_{i+1}\theta_{i+1}\xi_{i+1}$ that m'-fails and either $m' \le m$ or the size of the right term in q_{i+1} is strictly smaller than in q_i.

For instance, assume $t_iq_i\theta_i\xi_i$ m-holds, $t_i = \lambda y_1 \dots y_k$, $t_i \downarrow_1 t_{i+1} = y$, $t_{i+1} \downarrow_j t'_j$ for $1 \le j \le p$ and $q_i = q[(l_1, \dots, l_k), r]$. Therefore, $\theta_{i+1} = \theta_i\{l_1\xi_i i/y_1, \dots, l_k\xi_i i/y_k\}$, $q_{i+1} = q[l, r]$ and $\xi_{i+1} = \xi'$ when $\theta_{i+1}(y) = l\xi'n$. So, $t_i = \lambda y_1 \dots y_k.y\, t'_1 \dots t'_p$ and by assumption $(t_i\theta_i)l_1\xi_i \dots l_k\xi_i =_\beta r$. With k β-reductions we obtain $(l\xi_{i+1})t'_1\theta_{i+1} \dots t'_p\theta_{i+1}$ and position $t_{i+1}q_{i+1}\theta_{i+1}\xi_{i+1}$, therefore, $(m - k)$-holds. Next, assume $t_iq_i\theta_i\xi_i$ m-holds, $t_i = f$, $q_i = q[-, fs_1 \dots s_k]$ and $t_i \downarrow_j t'_j$ for $1 \le j \le k$. By assumption, $(ft'_1 \dots t'_k)\theta_i =_\beta fs_1 \dots s_k$. So, $t'_j\theta_i =_\beta s_j$. Consider any choice of next position. If $s_j : \mathbf{0}$ then $q_{i+1} = q[(\ ), s_j]$, $t_{i+1} = t'_j$ and $\theta_{i+1} = \theta_i$. Therefore, $t'_j\theta_{i+1} =_\beta s_j$ and so this next position m'-holds for some m' and s_j is strictly smaller than $fs_1 \dots s_k$. Alternatively, $s_j = \lambda x_{i_1} \dots x_{i_n}.s$. Therefore, $t'_j = \lambda z_1 \dots z_n.t'$ and $t'\theta_i\{c_{i_1}/z_1, \dots, c_{i_n}/z_n\} =_\beta s\{c_{i_1}/x_{i_1}, \dots, c_{i_n}/x_{i_n}\}$ m'-holds for some m' provided that the c_{i_j}'s are new (which is guaranteed as they are forbidden constants). So the next position m'-holds and the right term of $q_{i+1} = q[(c_{i_1}, \dots, c_{i_n}), s\{c_{i_1}/x_{i_1}, \dots, c_{i_n}/x_{i_n}\}]$ is strictly smaller than $fs_1 \dots s_k$. Assume $t_iq_i\theta_i\xi_i$ m-holds and $t_i = y$, $q_i = q[l, r]$, $l = \lambda z_1 \dots z_k.w$, $w = z\, l_1 \dots l_p$, $t_i \downarrow_j t'_j$ for $1 \le j \le k$ and $\xi_{i+1}(z) = t''\theta'n$; so, $t_{i+1} = t''$ and $\theta_{i+1} = \theta'$. By assumption, $((\lambda z_1 \dots z_k.w)\xi_i)(t'_1\theta_i) \dots (t'_k\theta_i) =_\beta r$. With k β-reductions the left term in this equation becomes $(t_{i+1}\theta_{i+1})l_1\xi_{i+1} \dots l_p\xi_{i+1}$ and so the next position $(m-k)$-holds. All other cases of (1) are close to one of these three, and the proof of (2) is also very similar. The only cases where the measure, size of right term in state and the number of β-reductions to normal form, does not decrease are applications of A3 when t_m is labelled with a dummy lambda and C4 when $l = w$ of Figure 3. As a supplementary argument we show that there cannot be an indefinite sequence of such applications of A3 followed by C4 by examining the index j that is called at these positions; namely, $\theta_{i+1}(y) = l\xi j$ in the case of A3 and $\xi_{i+1}(x) = t'\theta'j$ in the case of C4; this index must be strictly decreasing in such repeated sequence of applications of A3 followed by C4.

The result now follows: if $t \models P$ then for each initial position that starts from an equation there is an m such that it m-holds and for each disequation there is an m such that it m-fails. Conversely, if $t \not\models P$ then there is an initial position for some equation that m-fails or for some disequation there is an m such that it m-holds. $\square$

The game is analogous to a model-checking game in the sense that deciding a possibly complex temporal property of a transition graph can be formulated as a game whose arena is the graph and where the moves are locally small steps that traverse it; similarly, the

complex property whether t solves P is here formalised as a game whose arena is t involving locally small steps and local moves. In both cases, play proceeds until one definitely knows an outcome.

5. PROPERTIES OF GAME PLAYING

In the following we let $\pi, \sigma, \dots$ range over plays in a game $\mathsf{G}(t, P)$. The total number of different plays is at most the sum of the number of branches in the right terms u of P. For instance, in the case of Example 4.7 whose right term is $h(g(h(ha)))$ there is a single play. We now examine some properties of plays and introduce relationships between play positions that uses the play indices in the look-up tables.

Remark 5.1. Theorem 4.9 allows one to restrict the set of constants that can appear in a potential solution term t for P. Let $d : \mathbf{0}$ be a new constant that does not occur in any right term u of a (dis)equation in P (and which is also not a forbidden constant). Without loss of generality, we can assume that any potential solution term t to P only contains the constant d and constants that occur in the right terms of P: a similar observation is made in [13]. The justification appeals to moves A1 and A2 of Figure 3. Assume $t \models P$. Control in a play associated with an interpolation equation in P can never be at a node in t labelled with a constant that does not occur in a right term (as $\forall$ would win the play). If control in a play associated with an interpolation disequation in P is at a node t' labelled with a constant that does not occur in a right term then replacing t' in t with a single node labelled d preserves $\forall$'s win. $\square$

Definition 5.2.
(1) The *length* of π, $|\pi|$, is the number of positions $t_i q_i \theta_i \eta_i$ in π.
(2) The *ith position* of π is $\pi(i)$ where $1 \le i \le |\pi|$.
(3) The sequence of positions $\pi(i), \dots, \pi(j)$, $i \le j$, is written $\pi(i, j)$.

We write $t \in \pi(i)$, $q \in \pi(i)$, $\theta \in \pi(i)$ and $\xi \in \pi(i)$ when $\pi(i) = tq\theta\xi$ and $t \notin \pi(i)$ means that $\pi(i) = t'q\theta\xi$ and $t \ne t'$. We shall describe a sequence of positions $\pi(i, j)$ as an *interval*.

Definition 5.3. The *right term* of state $q[(l_1, \dots, l_k), r]$, $q[-, r]$ or $q[l, r]$ is the term r. Each l_i is a *left term* of $q[(l_1, \dots, l_k), r]$ and l is the *left term* of $q[l, r]$.

Definition 5.4. The interval $\pi(i, j)$ is *ri, right term invariant*, if $q \in \pi(i)$ and $q' \in \pi(j)$ share the same right term r. It is *nri* if it is not ri and $q' \in \pi(j)$ is not a final state.

Clearly, if $\pi(i, j)$ is ri then *every* state at every position in this interval shares the same right term. For instance, when π is the play of Figure 5, the interval $\pi(5, 17)$ is ri as all its states share the right term a; also, each position in $\pi(6, 17)$ is the result of moves A3 or C4. The outcome of the other moves in Figure 3, A1, A2, B1, C1, C2 and C3, depend on the right term of the state.

Fact 5.5. If $\pi(i, j)$ is ri and $t_j \in \pi(j)$ is labelled $\lambda\overline{y}$ for some $\overline{y}$ then each position in $\pi(i + 1, j)$ is the result of move A3 or C4 of Figure 3.

Intervals that are ri do not directly contribute to the solution of P.

Fact 5.6. If $t_i q_i \theta_i \xi_i, \dots, t_n q_n \theta_n \xi_n$ is a sequence of positions that is ri, t_n is labelled $\lambda\overline{y}$ and $q\{r'/r\}$ is state q with right term r' instead of r, then $t_i q_i \{r'/r\} \theta_i \xi_i, \dots, t_n q_n \{r'/r\} \theta_n \xi_n$ is also a sequence of positions that is ri.

Consider a position $tq\theta\xi$ of a play. If there is a free occurrence of y in the subtree t then $\theta(y)$ is defined; similarly, if there is a free occurrence of a variable z in a left term of q then $\xi(z)$ is defined. In contrast, if there is a bound occurrence of y, a node labelled $\lambda y_1 \ldots y_k$ with $y = y_j$, in the subtree t then $\theta(y)$ is not defined and similarly, if there is a bound occurrence of z in a left term of q then $\xi(z)$ is not defined.

Proposition 5.7. *Assume $\pi(i) = tq\theta\xi$.*

(1) *If $\lambda y_1 \ldots y_k$ labels a node in the subtree rooted at t then for each $j : 1 \le j \le k$, $\theta(y_j)$ is undefined.*

(2) *If y occurs free in the subtree rooted at t then $\theta(y)$ is defined.*

(3) *If $q = q[l, r]$ or $q[(l_1, \ldots, l_k), r]$ and $\lambda z_1 \ldots z_m$ occurs in l or in some l_j, $1 \le j \le k$, then for any $n : 1 \le n \le m$, $\xi(z_n)$ is not defined.*

(4) *If $q = q[l, r]$ or $q[(l_1, \ldots, l_k), r]$ and z occurs free in l or in some l_j, $j : 1 \le j \le k$, then $\xi(z)$ is defined.*

Proof. We prove this by induction on the position i in a play π. We also show by induction on i that if $\pi(i + 1) = t' \, q' \, \theta' \, \xi'$ then the following additional four properties hold.

- For any y, if $\theta'(y) = l\xi''j$ and $\lambda z_1 \ldots z_m$ occurs in l, then for any $n : 1 \le n \le m$, $\xi''(z_n)$ is not defined.
- For any y, if $\theta'(y) = l\xi''j$ and z occurs free in l, then $\xi''(z)$ is defined.
- If $\xi'(z) = t''\theta''j$ and $\lambda y_1 \ldots y_k$ labels a node in the subtree rooted at t'' then for $i : 1 \le i \le k$, $\theta''(y_i)$ is undefined.
- If $\xi'(z) = t''\theta''j$ and y occurs free in the subtree rooted at t'' then $\theta''(y)$ is defined.

For the base case, consider an initial position $\pi(1) = t_1 \, q[(v_1^i, \ldots, v_n^i), u_i] \, \theta_1\xi_1$ where t_1 is the root node of t labelled $\lambda z_1 \ldots z_n$. There are no free variables in t or in the v_j^i's. Moreover, both θ_1 and ξ_1 are empty; therefore (1) to (4) hold. Now we need to show the additional properties for $\pi(2) = t'q'\theta'\xi'$: by definition $\theta' = \theta_1\{v_1^i\xi_11/z_1, \ldots, v_n^i\xi_11/z_n\}$ and $\xi' = \xi_1$. Therefore, these properties hold. Consider next the general case for position $\pi(i)$. If $\pi(i)$ is the result of moves A1-A3 of Figure 3 applied to $\pi(i-1)$ then (1) and (2) follow because they are true at $\pi(i-1)$ and $\theta \in \pi(i)$ is an update of $\theta' \in \pi(i-1)$ with respect to the (potentially) free variables $y_1, \ldots, y_j$ such that $t_{i-1} \in \pi(i - 1)$ is labelled $\lambda y_1 \ldots y_j$. Parts (3) and (4) trivially hold for A1 and A2. In the case of A3, if $t_i \in \pi(i)$ is labelled $y \in \{y_1, \ldots, y_j\}$ when $t_{i-1} \in \pi(i - 1)$ is labelled $\lambda y_1 \ldots y_j$ then (3) and (4) follow from the induction hypothesis that they hold at $\pi(i-1)$; otherwise they follow from the induction hypothesis for the first two additional properties at $\pi(i-1)$. If $\pi(i)$ is the result of move B1 of Figure 3 to $\pi(i-1)$ then as θ, ξ are unchanged $(1) - (4)$ remain true. Finally, we examine the case when $\pi(i)$ is the result of moves C1-C4 to $\pi(i - 1)$. Cases (3) and (4) hold because $\xi \in \pi(i)$ is a simple updating of $\xi' \in \pi(i-1)$ where they hold. Cases (1) and (2) hold for C1-C3 because $\theta \in \pi(i)$ and $\theta' \in \pi(i - 1)$ are the same. In the case of move C4, (1) and (2) either follow from the induction hypothesis that they hold at $\pi(i-1)$ or from the induction hypothesis for the final two additional properties at $\pi(i - 1)$. Using that $(1) - (4)$ are true at $\pi(i)$, the argument is similar for showing that the four additional properties hold at $\pi(i + 1) = t'q'\theta'\xi'$. $\square$

Now we examine some simple relationships between look-up tables. We allow μ, ν to range over both kinds of look-up tables.

Definition 5.8. Two look-up tables μ, μ' are equal, $\mu = \mu'$, if, and only if, $\text{dom}(\mu) = \text{dom}(\mu')$ and for all $x \in \text{dom}(\mu)$, $\mu(x) = \mu'(x)$; that is, if $\mu(x) = s\nu i$ and $\mu'(x) = s'\nu'i'$ then $s = s'$, $\nu = \nu'$ and $i = i'$.

This is well defined because the definitions of look-up tables $\theta \in \Theta_k$ and $\xi \in \Xi_k$ in Definition 4.2 are well-founded with respect to the embedding of look-up tables.

Definition 5.9. A look-up table μ *extends* μ' if for all $x \in \mathrm{dom}(\mu')$, $\mu(x) = \mu'(x)$.

Example 5.10. Assume π is the play in Figure 7 that operates on the term tree in Figure 6. The look-up table $\theta \in \pi(18)$ consists of three entries $\{v\xi_1 1/z, v_1\xi_3 3/z_1, v_2\xi_{17}17/x_2\}$; it therefore extends $\theta' \in \pi(11)$ which is just the single entry $\{v\xi_1 1/z\}$. On the other hand, although $\theta'' \in \pi(36)$ extends θ' it does not extend θ; it consists of the entries $\{v\xi_1 1/z, v_2\xi_{35}35/z_2\}$. In similar fashion, $\xi \in \pi(35)$ which is $\{(3)\theta_2 2/y_1, (17)\theta_2 2/y_2, (11)\theta_6 34/y_3\}$ extends $\xi_6 \in \pi(6)$ which only contains the first two of these entries. $\qquad\square$

Let $\pi(i) = tq\theta\xi$ and let $\pi(j) = t'q'\theta'\xi'$ be a later position. If t' is a subtree of t and θ' extends θ then the free variable occurrences that are common to both t and t' have the same interpretation; their meaning is preserved at position j. Similarly, if η' extends η then the free variable occurrences that are common to the left terms of q and q' have the same interpretation.

Both look-up tables of positions that are the result of moves A1, A2, B1, C1-C3 of Figure 3 extend (or are equal to) those of the previous position. In the case of A3 the θ look-up table extends the one from the previous position but this is not true, in general, for the ξ table. Dually, in the case of C4 the ξ table extends the one of the previous position but this may not hold for the θ look-up table. We want to restore when *both* look-up tables are extensions of an earlier position. For this we introduce a similar notion to that in game semantics that later positions are justified by earlier positions [11]. We define when a later position is a child of an earlier position. It is at this point that we appeal to the third component of an entry in a look-up table.

Definition 5.11. Position $\pi(j) = tq\theta\xi$ is a *child* of position $\pi(i)$ if $i < j < |\pi|$ and the following by cases of which move $\pi(j)$ is the result of

(1) A2, B1, C2 or C3: then $i = j - 1$,
(2) A3: then t is labelled y and $\theta(y) = l\xi'i$,
(3) C4: then $q[l, r] \in \pi(j - 1)$, the head variable in l is x and $\xi(x) = t\theta i$.

Assume $\pi(j) = tq\theta\xi$ is the child of $\pi(i) = t'q'\theta'\xi'$; if $\pi(j)$ is the result of A3 then t' is the binder of t and if it is the result of C4 then t is a successor of t'.

Fact 5.12. Assume $\pi(j)$ is a child of $\pi(i)$.
(1) If $\pi(j) = t'q[l', r']\theta'\xi'$ is the result of A3 and $\pi(i) = t\,q[(l_1, \ldots, l_m), r]\theta\,\xi$, then t binds t', $\xi' = \xi$ and for some $k : 1 \le k \le m$, $l' = l_k$.
(2) If $\pi(j) = t'q[(l_1, \ldots, l_n), r']\theta'\xi'$ is the result of C4, $q[l', r'] \in \pi(j - 1)$ and $\pi(i) = tq[l, r]\theta\xi$ then $\theta' = \theta$, $l = \lambda z_1 \ldots z_m.w$ and for some $k : 1 \le k \le m$, $t \downarrow_k t'$ and the head variable of l' is z_k.

Fact 5.13. If $1 < j < |\pi|$ then there is a unique $i < j$ such that $\pi(j)$ is a child of $\pi(i)$.

For this reason, we also say that $\pi(i)$ is *the parent* of $\pi(j)$ instead of $\pi(j)$ is a child of $\pi(i)$.

Example 5.14. Let π be the play in Figure 7 which is on the tree in Figure 6. Every position that occurs at nodes (2) and (4) labelled with z is a child of $\pi(1)$; examples include $\pi(2)$, $\pi(12)$ and $\pi(40)$. Not every position that occurs at the nodes (6) and (10) labelled z_1 is a child of $\pi(3)$; positions $\pi(6)$, $\pi(22)$ and $\pi(34)$ are whereas $\pi(14)$, $\pi(30)$ and $\pi(35)$ are

not. Nodes (5) and (6) are the successors of (4); children of position $\pi(4)$ at (4) are $\pi(5)$, $\pi(17)$ and $\pi(21)$ while $\pi(13)$, $\pi(29)$ and $\pi(41)$ are not. Position $\pi(24)$ is a child of $\pi(23)$ through A2 and $\pi(9)$ is a child of $\pi(8)$ through B1. $\square$

Fact 5.15. If $\pi(j)$ is a child of $\pi(i)$, $t \in \pi(i)$ and $t' \in \pi(j)$ then $t = t'$ or there is a path of successors from t to t'.

The following is a critical consequence of the definition of a child position that *both* its look-up tables extend those of its parent.

Proposition 5.16. *Assume $\pi(j)$ is a child of $\pi(i)$.*
(1) $\theta' \in \pi(j)$ *extends* $\theta \in \pi(i)$,
(2) $\xi' \in \pi(j)$ *extends* $\xi \in \pi$.

Proof. Assume that $\pi(j)$ is a child of $\pi(i)$ and $\theta', \xi' \in \pi(j)$ and $\theta, \xi \in \pi(i)$. If $\pi(j)$ is a child of $\pi(i)$ as a result of B1 or C2 of Figure 3 then the result is true because the look-up tables of $\pi(i)$ and $\pi(j)$ are the same. In the case of C3, $\theta' = \theta$ and $\xi' = \xi$ or $\xi' = \xi\{t'_1\theta i/z_1, \ldots, t'_k\theta i/z_k\}$ for some t'_i, z_i, $1 \le i \le k$. Therefore, ξ' extends ξ as the z_i's are not defined in ξ by Proposition 5.7. A similar argument applies when $\pi(j)$ is the result of A2; now $\xi' = \xi$ and $\theta' = \theta$ or $\theta' = \theta\{l'_1\xi i/y_1, \ldots, l'_k\xi i/y_k\}$ for some l'_i and y_i, $1 \le i \le k$, and θ' extends θ as the y_i's are not defined in θ using Proposition 5.7. Next we examine the cases for A3 and C4. By definition of A3, the look-up table $\xi' = \xi$ and by definition of C4, $\theta' = \theta$. Therefore, we just need to prove the result for the other look-up tables. In both cases the proof proceeds by case analysis of $j - i$. The initial case is when $j = i + 1$. Therefore, for A3 this means that $t' \in \pi(j)$ is labelled with a variable y which is bound by $\lambda\overline{y}$ which labels $t \in \pi(j-1)$; so $\theta' = \theta\{l_1\xi i/y_1, \ldots, l_m\xi i/y_m\}$ for some m and l_k, $1 \le k \le m$. Consequently, θ' extends θ using Proposition 5.7 (since θ does not have an entry for y). Similarly, in the case of C4 when $j = i + 1$, $q[\lambda\overline{z}.z_p l_1 \ldots l_k, r] \in \pi(i)$ for some $k \ge 0$ and $\xi' = \xi\{t'_1\theta i/z_1, \ldots, t'_m\theta i/z_m\}$ and ξ does not have entries for the z_i's using Proposition 5.7. For the general case for A3, we examine the branch between $t \in \pi(i)$ and $t' \in \pi(j)$ where t is labelled $\lambda\overline{y}$ and t' is labelled y; which is the sequence of nodes with labels $u_1, \lambda\overline{y}_1, \ldots, u_n, \lambda\overline{y}_n$ where each u_i is a constant or variable and $n \ge 0$. Consider position $\lambda\overline{y}_n \in \pi(j-1)$. Clearly, θ' extends $\theta_{j-1} \in \pi(j-1)$ by a similar argument to the base case. Position $\pi(j-1)$ is a child of $\pi(j_n)$ for some j_n with $t_n \in \pi(j_n)$ which is labelled u_n by C4, B1 or C2: in all cases θ_{j-1} extends $\theta_{j_n} \in \pi(j_n)$ by definition of these moves. The argument continues for position $\pi(j_n-1)$. So, we reach a position $\pi(i_1)$ with $t_1 \in \pi(i_1)$ labelled u_1. By assumption, $\theta' \in \pi(j)$ extends $\theta_{i_1} \in \pi(i_1)$: this means they have the same entry for the y_i's in $\theta(j)$ when they are bound by $\lambda\overline{y}$ which is the label of $t \in \pi(i)$. Consider the relationship between $\pi(i_1)$ and $\pi(i)$. Clearly, it cannot be the case that $i > i_1$ because this would contradict the entries in θ_{i_1} for the variables in $\overline{y}$. Moreover, by definition of the moves in A, $\theta_{i+1} \in \pi(i+1)$ has the same entries for the y's in $\overline{y}$ as $\pi(i_1)$ and $t_1 \in \pi(i+1)$. A small argument shows that $\pi(i_1)$ must, therefore, be $\pi(i+1)$: otherwise, $\pi(j_1)$ would not be a child of $\pi(i_1)$. Consequently, θ' extends θ. We now examine the general case for C4. One possibility is that $\pi(j)$ is the result of a sequence of C3 moves followed by C4: clearly, in this case ξ' extends ξ. Otherwise, $q[\lambda\overline{z}.w, r] \in \pi(i)$ and $\xi_{i+1} \in \pi(i+1)$ is $\xi\{t'_1\theta i/z_1, \ldots, t'_m\theta i/z_m\}$ for some m; so ξ_{i+1} extends ξ. Therefore, $t_{j-1} \in \pi(j-1)$ is labelled with some y' and $q \in \pi(j-1)$ has the form $q[l, r']$ where l is $z_k l'_1 \ldots l'_{m'}$ or $\lambda\overline{x}.z_k l'_1 \ldots l'_{m'}$ for some k: we know that ξ' extends $\xi_{j-1} \in \pi(j-1)$. There may be a sequence of positions $\pi(j_n, j-1)$ where each $\pi(j')$, $j_n < j' \le j - 1$ is the result of C3 (and, therefore, ξ' extends $\xi_{j_n} \in \pi(j_n)$). Otherwise $j_n = j - 1$. Position $\pi(j_n)$

is a child of a unique position $t_{j_{n-1}} \in \pi(j_{n-1})$ labelled $\lambda \overline{y}'$ by A3 and so $\xi_{j_n} \in \pi(j_n)$ extends $\xi_{j_{n-1}} \in \pi(j_{n-1})$. This argument is now repeated: consider position $t'' \in \pi(j_{n-1}-1)$ labelled with y''. Again, there may be a sequence $\pi(j_{n-2}, j_{n-1}-1)$ where the moves are the result of C3 and either $j_{n-2} = i$ or $\pi(j_{n-2})$ is a child of $\pi(j_{n-3})$. Eventually, for some k, $j_{n-k} = i$ as each $\xi_{j_n} \in \pi(j_n)$ has the entries $t'_l \theta i$ for the z_l's. $\square$

Definition 5.17. The binary relation *is a descendent of* on positions is the reflexive and transitive closure of "is a child of".

6. Tiles and their plays

In this section, we connect the static structure, regions of a potential solution term of a (dual) interpolation problem, with the dynamics of game playing. To this end, partial subtrees of a term tree are introduced.

Definition 6.1. Assume $B = (B_1, \ldots, B_k, \mathbf{0})$.
(1) Node t' labelled λ is an *atomic leaf* of type $\mathbf{0}$.
(2) Node t' labelled $\lambda x_1 \ldots x_k$ is an *atomic leaf* of type B when each $x_j : B_j$
(3) If t' is labelled $u : \mathbf{0}$ then t' is a *simple tile*.
(4) If node t' is labelled $u : B$ and each node t_j is an atomic leaf of type B_j and $t' \downarrow_j t_j$, $1 \le j \le k$, then $t'(t_1, \ldots, t_k)$ is a *simple tile*.

A potential solution tree without its initial lambda is a tree of simple tiles. For instance, the region $(2)((3), (11))$ of Figure 1 is a simple tile labelled $z(\lambda x, \lambda)$ with atomic leaves (3) and (11) labelled λx and λ; the region $(12)((13), (19))$ is labelled $z(\lambda y, \lambda)$; the region $(4)((5))$ is a simple tile labelled $f(\lambda)$. Single nodes such as (16) and (20) are also simple tiles but without atomic leaves. The definition precludes node (2) by itself or $(2)((3))$ as simple tiles.

In the following, as it makes the presentation cleaner, we describe tiles directly through their labelling. For example, $z(\lambda x, \lambda)$ identifies $(2)((3), (11))$ of Figure 1. When there is ambiguity, such as with $z_1(\lambda)$ of Figure 6, we disambiguate by describing the root node; $z_1(\lambda)$ at (6) versus $z_1(\lambda)$ at (10).

Tiles can be composed to form composite tiles. A (possibly composite) tile is a partial tree which can be extended at any of its atomic leaves. If $\tau(\lambda \overline{x})$ is a tile with leaf $\lambda \overline{x}$ and τ' is a simple tile, then $\tau(\lambda \overline{x}.\tau')$ is the composite tile that is the result of placing τ' directly beneath $\lambda \overline{x}$ in τ. For instance, we can compose the tile $z(\lambda y, \lambda)$ of Figure 1 with the tile $z(\lambda s, \lambda)$ beneath λy and produce the composite tile $z(\lambda y.z(\lambda s, \lambda), \lambda)$ which has three atomic leaves: in Figure 1 this tile is the region $(12)((13)(14)((15), (17)), (19))$. We write $\tau(\lambda \overline{x}_1, \ldots, \lambda \overline{x}_k)$ if τ is a (composite) tile with atomic leaves $\lambda \overline{x}_1, \ldots, \lambda \overline{x}_k$.

A tile $\tau(\lambda \overline{x}_1, \ldots, \lambda \overline{x}_k)$ is a multi-holed context. It is also a subregion of a term and we assume that the usual definitions of free and bound variable occurrences apply: for instance, the free variables in $z(\lambda y.z(\lambda s.s, \lambda), \lambda)$ are the two occurrences of z. Later we shall manipulate tiles and, therefore, we have given them an independent existence.

Definition 6.2. A (composite) tile τ is said to be *basic* if τ contains
(1) exactly one occurrence of a free variable and no occurrences of constants, or
(2) exactly one occurrence of a constant and no occurrences of free variables.

By definition, simple tiles are basic. The single occurrence of the free variable or constant in a basic tile is its "head" element. Particular contiguous regions of a term tree are basic tiles. In Figure 1 the region $z(\lambda s.s, \lambda)$ is a basic tile rooted at (14) with the single atomic leaf λ. However, if we also included node (18) then it would be a composite tile $z(\lambda s.s, \lambda.y)$ without atomic leaves, but not a basic tile.

Definition 6.3. Assume τ and γ are basic tiles in a tree.
(1) γ is *j-below* $\tau(\lambda\overline{x}_1, \ldots, \lambda\overline{x}_k)$ if there is a path of successors from $\lambda\overline{x}_j$ to γ.
(2) γ is *below* τ (or τ is *above* γ) if γ is j-below τ for some j.

In Figure 6, the tile $x_1(\lambda)$, rooted at (8), is 1-below $z(\lambda z_1, \lambda z_2)$ and z_2 at (20) is 2-below the same tile.

Definition 6.4. Assume τ and γ are basic tiles in a tree.
(1) γ is an *immediate j-dependent* of tile τ if γ is j-below τ and γ contains a free variable that is bound in τ.
(2) γ is a *j-dependent* of τ if it is an immediate j-dependent of τ or there is a τ' that is an immediate j-dependent of τ and γ is a k-dependent of τ' for some k.
(3) γ is a (*immediate*) *dependent* of τ if γ is a (immediate) j-dependent of τ for some j.

The tile $z_1(\lambda)$ rooted at (6) of Figure 6 is an immediate 1-dependent of $z(\lambda z_1, \lambda z_2)$ rooted at (2) and z_2 rooted at (20) is a 2-dependent. Given a tile τ, its set of dependents are all the tiles below it whose free variables are either bound within τ or are bound within a dependent of τ. For instance, the dependents of $z(\lambda x_1, \lambda x_2)$ of Figure 6 are $x_1(\lambda)$ rooted at (8) and x_2 at (16).

Definition 6.5. Tiles τ and γ belong to the *same family* in a tree if one is a dependent of the other, or there is a τ' such that both are dependents of τ'. The *family* of tiles associated with τ in a tree consists of τ and each tile γ that belongs to the same family as τ.

In Figure 6 the family associated with x_2 at (16) is the set of tiles containing $z(\lambda x_1, \lambda x_2)$ at (4), $x_1(\lambda)$ at (8) and x_2.

Remark 6.6. Assume basic tiles τ and γ. If position $\pi(j)$ is the result of move A3 of Figure 3 and is a child of $\pi(i)$ and $t' \in \pi(j)$ is in γ and $t \in \pi(i)$ is in τ then γ is a dependent of τ. If $\pi(j)$ is the result of move C4 and $t' \in \pi(j)$ is in γ and $t \in \pi(j-1)$ is in τ, then these two tiles belong to the same family: this important property is proved in Section 8. In the play of Figure 7, play at x_1 at position 16 jumps to λx_2 of $z(\lambda x_1, \lambda x_2)$ and play at x_2 at position 20 jumps to λx_1. ☐

Definition 6.7. Assume τ and γ are basic tiles that each contain an occurrence of a free variable.
(1) τ and γ are *equivalent*, written $\tau \equiv \gamma$, if they are α-equivalent; that is, they are the same basic tiles with the same number and type of atomic leaves and with the same single free variable occurrence y.
(2) If $\tau \equiv \gamma$ and t_1 is a node of τ and t_1' is the corresponding equivalent node of γ then we write $t_1 \equiv_\gamma^\tau t_1'$.
(3) If $\tau \equiv \gamma$ and γ is below τ then γ is said to be an *embedded* tile.

The reader can verify that "tile equivalence" is indeed an equivalence relation. In Figure 1, each pair of tiles $z(\lambda x, \lambda)$, $z(\lambda u, \lambda)$, $z(\lambda y, \lambda)$ and $z(\lambda s, \lambda)$ is equivalent: nodes such as those

with labels λu and λs correspond. Each of these tiles except the first is also an embedded tile. The tile $z_1(\lambda)$ rooted at (6) is equivalent to $z_1(\lambda)$ rooted at (10) of Figure 6: however, neither is equivalent to $x_1(\lambda)$ rooted at (8).

Definition 6.8. Assume $\tau = \tau(\lambda\overline{x}_1, \ldots, \lambda\overline{x}_k)$ is a basic tile.

(1) τ is a *top* tile if it contains a free variable occurrence that is is bound by the initial lambda of the term tree.

(2) τ is *j-end* if τ has no immediate j-dependents. It is an *end* tile if it is j-end for *all* $j : 1 \leq j \leq k$.

(3) τ is a *constant* tile if it contains an occurrence of a constant or it is a dependent of a constant tile.

The tile $z(\lambda x, \lambda)$ rooted at (2) in Figure 1 is a top tile as z is bound by the initial lambda at node (1). It is also 2-end because no tile beneath node (11) contains a free variable occurrence that is bound within it: however, it is not 1-end because of the tile occurrence x at node (8). Tile $z(\lambda u, \lambda)$ rooted at (6) is a top tile and also an end tile. Tiles $f(\lambda x z_1 z_2, \lambda)$ rooted at (6), $x(\lambda)$ at (8) and z_2 at (12) in Figure 4 are all constant tiles.

The previous definitions provide a classification of basic tiles within a tree that only appeals to the static structure of the tree. Tiles can also be categorised in terms of dynamic properties of game playing.

Definition 6.9. The interval $\pi(i, j)$ is a *play* on the simple tile $u(\lambda\overline{x}_1, \ldots, \lambda\overline{x}_k)$ if $u \in \pi(i)$, $\lambda\overline{x}_m \in \pi(j)$ for some $m : 1 \leq m \leq k$ and $\pi(j)$ is a child of $\pi(i)$. It is an *m-play* if $\lambda\overline{x}_m \in \pi(j)$.

A simple tile has the form $y(\lambda\overline{x}_1, \ldots, \lambda\overline{x}_k)$ or $f(\lambda\overline{x}_1, \ldots, \lambda\overline{x}_k)$. A play $\pi(i, j)$ on such a tile starts at the head of the tile and ends at one of its atomic leaves; importantly, $\pi(j)$ must be a child of $\pi(i)$. A play on a simple constant tile $u(\lambda\overline{x}_1, \ldots, \lambda\overline{x}_k)$ is a consecutive pair of positions $\pi(i, i + 1)$ with $u \in \pi(i)$ and $\lambda\overline{x}_m \in \pi(i + 1)$ for some m (by moves B1 or C2 of Figure 3).

Fact 6.10. If $\pi(i, j)$ is a play on a simple constant tile then $j = i + 1$.

For instance, if π be the play in Example 4.5 whose term tree is depicted in Figure 4 then $\pi(6, 7)$ is a play on the simple constant tile $f(\lambda x z_1 z_2, \lambda)$ and $\pi(8, 9)$ is a play on the simple constant tile $x(\lambda)$.

A play $\pi(i, j)$ on a simple non-constant tile $y(\lambda\overline{x}_1, \ldots, \lambda\overline{x}_k)$ can have arbitrary length. It starts at the node labelled with y and finishes at a node labelling an atomic leaf $\lambda\overline{x}_m$. In between, control can be almost anywhere in the tree (including the node labelling y). However, because $\pi(j)$ is a child of $\pi(i)$ the look-up tables of $\pi(j)$ extend those of $\pi(i)$ as shown in Proposition 5.16.

Example 6.11. Let π be the play in Figure 7 on the tree in Figure 6. The simple tile $\tau = z(\lambda z_1, \lambda z_2)$ is rooted at (2). There are various plays on τ: 1-plays include $\pi(2, 3)$ and $\pi(2, 39)$; $\pi(2, 23)$ and $\pi(2, 35)$ are 2-plays. If $\gamma = z(\lambda x_1, \lambda x_2)$ rooted at (4) then $\pi(4, 5)$ and $\pi(12, 13)$ are 1-plays on γ: however, the interval $\pi(4, 13)$ is not a play on γ because $\pi(13)$ is a child of $\pi(12)$. If τ is the end tile $x_1(\lambda)$ rooted at (8) then there is just one play $\pi(16, 33)$ on it; the interval $\pi(32, 33)$ is not a play on τ. □

The definition of play on a simple tile can be extended to arbitrary composite tiles by composing consecutive plays on the simple tiles from its root to one of its atomic leaves.

Definition 6.12. The interval $\pi(i, j)$ is a play on the composite tile $\tau = \tau(\lambda \overline{x}_1, \ldots, \lambda \overline{x}_k)$ if there is a path of successor nodes with labels $u_1, \lambda \overline{y}_1, \ldots u_n, \lambda \overline{y}_n$ from the root of τ to an atomic leaf $\lambda \overline{x}_m = \lambda \overline{y}_n$ such that $\pi(i, j) = \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ and $\pi(i_l, j_l)$ is a play on the simple tile $u_l(\ldots \lambda \overline{y}_l \ldots)$ with $\lambda \overline{y}_l \in \pi(j_l)$ for $1 \leq l \leq n$. It is an m-play if $\lambda \overline{x}_m \in \pi(j)$.

If $t \in \pi(j)$, $j > 1$, and t is labelled with a lambda then there is a unique partition of $\pi(2, j)$ into plays on the simple tiles that occur on the branch from the node directly beneath the initial λ of the tree to t. The partition also preserves children.

Proposition 6.13. *Assume $t \in \pi(j)$ and t is labelled with $\lambda \overline{y}_n$ and $\lambda \overline{y}_0, u_1, \lambda \overline{y}_1, \ldots u_n, \lambda \overline{y}_n$ is the labels of the sequence of nodes from the root of the tree to t. Then there is a unique partition of $\pi(1, j) = \pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ such that for $1 \leq m \leq n$*

(1) $j_0 = 1$ and $\pi(i_m, j_m)$ is a play on the simple tile $u_m(\ldots \lambda \overline{y}_m \ldots)$ with $\lambda \overline{y}_m \in \pi(j_m)$,
(2) if u_m is a variable bound by $\lambda \overline{y}_k$ then $\pi(i_m)$ is a child of $\pi(j_k)$.

Proof. Assume $t \in \pi(j)$ and t is labelled with $\lambda \overline{y}_n$ and $\lambda \overline{y}_0, u_1, \lambda \overline{y}_1, \ldots, u_n, \lambda \overline{y}_n$ are the labels of the sequence of nodes from the root of the term tree to $\lambda \overline{y}_n$. Assume $j_n = j$. Let $\pi(i_n, j_n)$ be the play on the simple tile $t_n = u_n(\ldots \lambda \overline{y}_n \ldots)$ such that $\pi(j_n)$ is a child of $\pi(i_n)$: move $\pi(j_n)$ is the result of B1, C2 or C4 of Figure 3 (and, therefore, $\pi(i_n)$ is uniquely defined from Definition 5.11). Let $j_{n-1} = i_n - 1$, the argument is now repeated for $\pi(i_{n-1}, j_{n-1})$ as a play on $u_{n-1}(\ldots \lambda \overline{y}_{n-1} \ldots)$ and so on for subsequent tiles in the branch from the node labelled $\lambda \overline{y}_n$ to the root. Clearly, this will define a partition of $\pi(2, j)$ into $\pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ with $i_1 = 2$: hence, we can add $\pi(j_0)$ with $j_0 = 1$ at the beginning. Next assume that u_m is a variable bound by $\lambda \overline{y}_k$. It is straightforward to show that the look-up tables in $\pi(i_m)$ extend those in $\pi(j_k + 1)$, which, therefore, implies that $\pi(i_m)$ is a child of $\pi(j_k)$; this follows from repeated application of Proposition 5.16. $\quad\square$

Definition 6.14. If $t \in \pi(j)$ and t is labelled with $\lambda \overline{y}_n$ and $\lambda \overline{y}_0, u_1, \lambda \overline{y}_1, \ldots u_n, \lambda \overline{y}_n$ is the labels of the sequence of nodes from the root of the tree to t then the unique partition $\pi(1, j) = \pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ such that for $1 \leq m \leq n$ where each $\pi(i_m, j_m)$ is the play on $u_m(\ldots \lambda \overline{y}_m \ldots)$ of Proposition 6.13 is called the *b-partition for position $\pi(j)$*.

Example 6.15. Let π be the play in Figure 7 on the tree in Figure 6. Consider the two positions $\lambda \in \pi(15)$ and $\lambda \in \pi(31)$ where λ is at node (7). Below we illustrate the b-partitions for $\pi(15)$ and $\pi(31)$. The branch from the root to λ is presented horizontally.

λz	z	λz_1	z	λx_1	z_1	λ
$\pi(1)$	$\pi(2,3)$		$\pi(4,5)$		$\pi(6,15)$	
$\pi(1)$	$\pi(2,3)$		$\pi(4,21)$		$\pi(22,31)$	

The b-partitions capture children along their branches: $\pi(2)$ and $\pi(4)$ are children of $\pi(1)$; $\pi(3)$ is a child of $\pi(2)$; $\pi(5)$ and $\pi(21)$ are children of $\pi(4)$; $\pi(6)$ and $\pi(22)$ are children of $\pi(3)$. Plays within these b-partitions may jump around the term tree: for instance, the play $\pi(22, 31)$ on $z_1(\lambda)$ rooted at (10) includes a sequence of moves that descends the branch below (17). $\quad\square$

We now exhibit some useful uniformities of plays on tiles. Given a play π and tile τ we examine three aspects. First, there can be multiple plays $\pi(i, n_1), \ldots, \pi(i, n_m)$ on τ from the same initial position. Second, there can be multiple plays from different starting positions $\pi(i_1, n_1), \ldots, \pi(i_m, n_m)$ on τ. A third consideration is the relationship between plays on τ' and τ when these tiles are equivalent.

We start with a pertinent property of a simple non-constant tile.

Proposition 6.16. *If $\pi(i,m)$ and $\pi(i,n)$ are plays on the simple tile $y(\lambda\overline{x}_1,\ldots,\lambda\overline{x}_k)$ and $i < m < n$ then there is a position $\pi(m')$, $m' < n$, that is a child of $\pi(m)$.*

Proof. Assume $\pi(i) = t\, q[\lambda z_1 \ldots z_k.w, r]\,\theta\,\xi$, t is labelled y and $\pi(i,m)$, $\pi(i,n)$, $n > m$, are plays on the simple tile $y(\lambda\overline{x}_1,\ldots,\lambda\overline{x}_k)$; so, $\lambda\overline{x}_j \in \pi(m)$ for some j. The look-up table $\xi_{i+1} \in \pi(i+1)$ is $\xi\{\lambda\overline{x}_1\theta i/z_1,\ldots,\lambda\overline{x}_k\theta i/z_k\}$ and the ξ look-up tables in $\pi(m-1)$ and $\pi(n-1)$ both extend ξ_{i+1} because $\pi(m)$, $\pi(n)$ are children of $\pi(i)$. Clearly, no look-up table $\xi_l \in \pi(l)$, $l < i+1$, has entries $\xi(z_{i'}) = \lambda\overline{x}_{i'}\theta i$ because for any z, if $\xi_l(z) = t'\theta'j$ then $j < i$. Consider the first position $\pi(m_1)$ after $\pi(m)$ that is at a node labelled with a variable y_1. Clearly, this node is below the node labelled $\lambda\overline{x}_j$ in the tree. If y_1 is bound by $\lambda\overline{x}_j$ then $\pi(m_1)$ is a child of $\pi(m)$ and the result is proved. Otherwise $\pi(m_1)$ is a child of an earlier position $\pi(l)$. Either $l < i$ or $i < l$. Assume the former; so, via move A3 the look-up table $\xi_l \in \pi(l)$ cannot have an entry for any z of the form $t'\theta''i$ because $l < i$. Play may then jump anywhere in the term tree using move C4. If there is not a play $\pi(m_1, n_1)$ on the simple tile whose head node is labelled y_1 then for all later positions $\pi(m_2)$, $m_2 > m_1$, $\xi_{m_2} \in \pi(m_2)$ cannot include an entry for any z of the form $t'\theta''i$; this is a contradiction because of the position $\pi(n)$. Therefore, play must reach a position $\pi(n_1)$ that is a child of $\pi(m_1)$; so $t'' \in \pi(n_1)$ is a successor of the node labelled with y_1 and is therefore below (the node labelled) $\lambda\overline{x}_j$. Next assume that $i < l$ so y_1 is bound by a $\lambda\overline{y}$ that is below (the node labelled) $\lambda\overline{x}_j$. But then y_1 is bound to a leaf of a constant tile that occurs between the node labelled $\lambda\overline{x}_j$ and the node labelled y_1 and so move C2 must apply and so play descends to a successor with position $\pi(n_1)$ that is a child of $\pi(m_1)$ where $n_1 = m_1 + 1$. This argument is now repeated for the next position after $\pi(n_1)$ that is at a node labelled with a variable $y_2 \in \pi(m_2)$: this node must be below the node labelled $\lambda\overline{x}_j$. The argument proceeds as above, except there is the new case that $\pi(m_2)$ is a child of $\pi(n_1)$. However, by move A3, $\xi_{m_2} \in \pi(m_2)$ cannot include an entry for some z of the form $t'\theta'i$. Therefore, play must eventually reach a child of $\pi(m)$. $\square$

By iteration, this property extends to basic tiles.

Corollary 6.17. *If $\pi(i,m)$ and $\pi(i,n)$ are plays on the basic tile $\tau(\lambda\overline{x}_1,\ldots,\lambda\overline{x}_k)$, $i < m < n$, and $\pi(i,m) = \pi(i_1,j_1),\ldots,\pi(i_n,j_n)$ where $u_1,\lambda\overline{y}_1,\ldots,u_n,\lambda\overline{y}_n$ is the path of (labels of) nodes from the root of τ to its atomic leaf $\lambda\overline{x}_j = \lambda\overline{y}_n \in \pi(m)$ and $\pi(i_l,j_l)$ is a play on the simple tile $u_l(\ldots\lambda\overline{y}_l\ldots)$ with $\lambda\overline{y}_l \in \pi(j_l)$ for $1 \leq l \leq n$ then there is a position m' such that $m < m' < n$ and $\pi(m')$ is a child of $\pi(j_{l'})$ for some l'.*

One consequence of Corollary 6.17 is that there cannot be more than one play on a basic end tile that starts from the same position. More precisely, we have the following.

Proposition 6.18. *If $\pi(i,m)$ is a j-play on the basic tile τ which is j-end, $\pi(i,n)$ is a play on τ and $n \geq m$ then $n = m$.*

Proof. Assume $\tau = \tau(\lambda\overline{x}_1,\ldots,\overline{x}_k)$ is j-end. So τ has no dependents below the node labelled $\lambda\overline{x}_j$; therefore, the binder of every free variable that occurs in the subtree below this node occurs above the tile τ in the tree. Assume $\pi(i,m)$ is a j-play on τ and $\pi(i,n)$, $n > m$, is also a play on τ. The play $\pi(i,m)$ can be partitioned into plays on the simple tiles of τ from its root to its atomic leaf $\lambda\overline{x}_j \in \pi(m)$. Therefore, by Corollary 6.17 there is a position $\pi(m')$, $m < m' < n$, that is a child of one of the final positions of a simple tile in the partition of $\pi(i,m)$. But then, by the definition of child, see Fact 5.12, there must be a free variable below the node labelled $\lambda\overline{x}_j$ that is bound within τ which contradicts the assumption that τ is j-end. $\square$

Special restricted plays of basic tiles are defined.

Definition 6.19. Assume $\pi(i, j)$ is a play on basic tile τ.
(1) It is a *shortest play* on τ if no proper prefix $\pi(i, k)$, $k < j$, is also a play on τ.
(2) It is a *shortest m-play* on τ if $\pi(i, j)$ is a m-play on τ and no proper prefix $\pi(i, k)$, $k < j$, is also an m-play on τ.
(3) It is also an *internal* play (m-play) on τ provided that for any $n : i \leq n \leq j$, if $t \in \pi(n)$ then t is a node of τ.
(4) It is also an *ri* play (m-play) on τ if $\pi(i, j)$ is ri (see, Definition 5.4).

Let π be the play in Figure 5. The interval $\pi(2, 3)$ is a shortest play on tile $z(\lambda x, \lambda)$ rooted at (2) of Figure 1: this play is also ri, internal and a shortest 1-play. Although $\pi(2, 9)$ is a shortest 2-play, it is neither a shortest play nor an internal play.

Next, we define a uniformity condition concerning multiple plays on a basic tile.

Definition 6.20. Assume τ is a basic tile whose root node is t'. Tile τ is *j-directed with respect to the interval* $\pi(i, |\pi|)$ if
(1) $t' \notin \pi(m)$ for all $m : i \leq m \leq |\pi|$, or
(2) $\pi(m)$ is the first position $m \geq i$ with $t' \in \pi(m)$ and there is a shortest j-play $\pi(m, n)$ on τ, $\pi(m, n)$ is ri and τ is j-directed with respect to $\pi(n + 1, |\pi|)$.

Definition 6.21. The basic tile τ is *j-directed* with respect to the game $G(t, P)$ if τ is j-directed with respect to $\pi(1, |\pi|)$ for every play $\pi \in G(t, P)$.

If τ is j-directed with respect to a game then every play π contains a (unique) sequence of ri intervals $\pi(i_k, m_k)$, for some $k \geq 0$, which are shortest j-plays on τ as follows (assuming t' is the root of τ and $\lambda \overline{x}_j$ is its jth atomic leaf).

$$
\begin{array}{ccccccccc}
 & t' & & \lambda\overline{x}_j & & t' & & \lambda\overline{x}_j & \\
\pi(1) & \ldots & \pi(i_1) & \ldots & \pi(m_1) & \ldots & \pi(i_n) & \ldots & \pi(m_n) & \ldots & \pi(|\pi|)
\end{array}
$$

By definition, t' cannot occur in a position that is outside of these intervals $\pi(i_k, m_k)$. A tile τ can be j-directed with respect to a game for multiple j. If π is the play of Figure 7 on the tree in Figure 6 then tile $z(\lambda x_1, \lambda x_2)$ is 1-directed with respect to $\pi(1, |\pi|)$ because of the following sequence:

$$
\begin{array}{cccccccccc}
 & z & \lambda x_1 & & z & \lambda x_1 & & z & \lambda x_1 & & z & \lambda x_1 & \\
\pi(1) & \ldots & \pi(4)\,\pi(5) & \ldots & \pi(12)\,\pi(13) & \ldots & \pi(28)\,\pi(29) & \ldots & \pi(40)\,\pi(41) & \ldots & \pi(44)
\end{array}
$$

however, it is not 2-directed with respect to $\pi(1, |\pi|)$ because $\pi(4, 17)$ is nri.

Recall that a basic top tile, see Definition 6.8, has a single variable occurrence that is bound by the initial lambda of the term tree. Equivalent basic top tiles are subject to the following local uniformity properties.

Proposition 6.22. *Assume τ, γ are basic top tiles, $\tau \equiv \gamma$ and $\pi(i, i+m)$ is a shortest play on τ.*
(1) $\pi(i, i + m)$ *is an internal play on τ.*
(2) *If $\pi(i, i + m)$ is ri and $\pi(i', m')$ is a shortest play on γ then $\pi(i', m')$ is ri, $m' = i' + m$ and for all $j : 1 \leq j \leq m$, $t_1 \in \pi(i + j)$ implies $t_1' \in \pi(i' + j)$ where $t_1 \equiv_\gamma^\tau t_1'$.*
(3) *If $\pi(i, i + m)$ is ri and a j-play on τ then τ and γ are j-directed with respect to $\pi(1, |\pi|)$.*
(4) *If $\pi(i, i + m)$ is nri and $t \in \pi(i')$ where t is the root node of γ then there is an interval $\pi(i', i' + m')$ which is internal to γ that is either a shortest play on γ and nri or $|\pi| = i' + m'$.*

Proof. Let t be the root node of top tile τ and assume $t \in \pi(i)$. The state $q \in \pi(i)$ has the form $q[v, r]$ where v is a closed left term, a v_j^l from a (dis)equation of P. Therefore, a shortest play $\pi(i, i+m)$ on τ is internal (as a jump outside τ requires there to be a free variable in v via move C4 of Figure 3) which shows (1). If $\tau \equiv \gamma$, t' is the root of γ and $t' \in \pi(i')$ then the state $q' \in \pi(i')$ has the same left term $q[v, r']$ as $q[v, r] \in \pi(i)$. If the play $\pi(i, i+m)$ is ri then there is a corresponding ri play $\pi(i', i'+m)$ on γ consisting of the same sequence of corresponding positions in τ and states (except for their right terms r and r') which shows (2). Furthermore, tile τ is j-directed with respect to π when $\pi(i+m)$ is a j-play on τ because each time play is at t there is the same repeated shortest play on it (and similarly for γ). For (4), if the shortest play $\pi(i, i+m)$ on τ is nri and $t' \in \pi(i')$ then either there is a shortest play on γ that is nri (but may involve different $\forall$-choices) or play remains within γ and the final state is reached. $\square$

Tiles are equivalent to themselves, $\tau \equiv \tau$; so Proposition 6.22 also applies to repeated shortest plays on a single top tile τ. Consider the tree in Figure 6 and the play in Figure 7. The equivalent tiles $z(\lambda x_1, \lambda x_2) \equiv z(\lambda z_1, \lambda z_2)$ are top tiles. Shortest π plays on the first of these tiles $\pi(4, 5)$, $\pi(12, 13)$, $\pi(28, 29)$ and $\pi(40, 41)$ are 1-plays that are ri and each corresponds to the single shortest play $\pi(2, 3)$ on $z(\lambda z_1, \lambda z_2)$.

Top basic tiles are distinguished because their only free variable is bound by the initial lambda. We now show that there are also play uniformities on other equivalent tiles, in the case of *embedded* tiles (as in Definition 6.7). If $\tau \equiv \gamma$ and γ is below τ then shortest plays on the embedded tile γ are constrained by earlier plays on τ.

Proposition 6.23. *Assume τ, γ are basic tiles, $\tau \equiv \gamma$ and γ is below τ.*

(1) *If $\pi(j, j+m)$ is a shortest play on γ that is ri and a k-play then there is a shortest play $\pi(i, i+m)$ on τ, $i < j$, that is ri and a k-play.*

(2) *If $\pi(j, j+m)$ is a shortest play on γ that is nri then there is a shortest play $\pi(i, i+n)$ on τ, $i < j$, that is nri.*

Proof. Assume τ and γ are basic equivalent tiles, $\tau \equiv \gamma$, γ is below τ and $\pi(j, j+m)$ is a shortest play on γ that is ri and a k-play. Consider the unique b-partition of position $\pi(j+m)$ from the root of the term tree to the atomic leaf of γ, see Definition 6.14. Because γ is below τ, this b-partition contains a play $\pi(i, i+m')$ on τ. Because τ and γ are equivalent and $\pi(j, j+m)$ is ri, these tiles contain the same single free variable y at their root; therefore, positions $\pi(i)$ and $\pi(j)$ share a common parent. Therefore, $\xi \in \pi(i)$ is the same look-up table as $\xi \in \pi(j)$, $\theta' \in \pi(j)$ extends $\theta \in \pi(i)$ and if $q[l, r'] \in \pi(j)$ then $q[l, r] \in \pi(i)$ for some r. Now, it is an easy argument that $\pi(i, i+m)$ is ri and a k-play on τ. If instead $\pi(j, j+m)$ is nri then some prefix of $\pi(i, i+m')$ is a shortest nri play on τ that may involve different $\forall$ choices than in $\pi(j, j+m)$. $\square$

In the case of the tree in Figure 6 and the play in Figure 7, tile $z_1(\lambda)$ rooted at (10) is an embedded (end) tile: its single play $\pi(34, 43)$ which is nri corresponds to the earlier play $\pi(22, 31)$ on $z_1(\lambda)$ rooted at (6).

There is an even stronger property of embedded end tiles: an embedded end tile is either j-directed with respect to a game for some j, has at least one nri play or play finishes within it.

Proposition 6.24. *Assume τ, γ are basic end tiles in a term tree t, γ is j-below τ, $\tau \equiv \gamma$ and $\pi \in \mathsf{G}(t, P)$. Then either*

(1) *γ is j-directed with respect to $\pi(1, |\pi|)$, or*

(2) *there exist m, n, m', n' such that $m < n < m' \leq n'$, $\pi(m, n)$ is a nri j-play on τ, $t' \in \pi(m')$ is the root node of γ and $\pi(m', n')$ is a nri play on γ or $n' = |\pi|$ and no prefix of $\pi(m', n')$ is a play on γ.*

Proof. Assume τ, γ are basic end tiles in term tree t, $\tau \equiv \gamma$ and and γ is j-below τ. We follow the argument in the proof of Proposition 6.23. Both τ and γ have the same head variable y that is bound to the same $\lambda \overline{y}$ above τ in t. Let $\pi \in \mathsf{G}(t, P)$. Consider the first position $t' \in \pi(m)$ where t' is the root of γ and the b-partition for $\pi(m-1)$. This partition must contain a j-play on τ, $\pi(i, i+k)$ (because γ is j-below τ); positions $\pi(i)$ and $\pi(m)$ share a common parent. If $\pi(m, m')$ is play on γ then the b-partition for $\pi(m')$ is a simple extension of that for $\pi(m-1)$; the play on τ is still $\pi(i, i+k)$. By Proposition 6.18, $\pi(i, i+k)$ is a j-play and so it is a corresponding play to $\pi(m, m')$ by Proposition 6.23; so, $\pi(m, m')$ is ri and is a j-play on γ. This argument is repeated for the next position, $m'' > m'$, such that $t' \in \pi(m'')$ or until the j-play on τ in the b-partition is nri; in which case, there is either an nri play on γ or a final state is reached. $\square$

7. Two transformations

Given an interpolation problem P, the aim is to prove decidability of higher-order matching by establishing a small model property: if $t \models P$ then there is a *small* term $t' \models P$ (whose size is determined from P). As we saw in Section 5, the number of plays in the game $\mathsf{G}(t, P)$ is bounded by the number of branches in the right terms u of the (dis)equations of P. However, there is no upper bound on the length of a play that is independent of the size and order of t. Nevertheless, a long play must contain significant ri intervals that do not directly contribute to solving P: the number of times the right term in a state changes within all plays is bounded by δ, the right size of P, Definition 3.8. Game-theoretically, what will underpin the conversion of an arbitrary solution term t to a small solution term t' is manipulation of tiles and their ri plays (using omission, repetition and permutation). The proof for the general case is quite intricate. So, we proceed in stages. In this section we provide two proofs of decidability of 3rd-order matching, one using a tree model property of game playing and the other using properties of embedded end tiles. Both proofs appeal to omission of tiles and their ri intervals.

As a first step, we introduce two transformations on term trees (for terms of any order). A transformation $\mathbf{T}$ converts a term tree t into a term tree t', written $t \, \mathbf{T} \, t'$.

Definition 7.1. Assume t' is a subtree of t whose root is labelled with a variable y or a constant $f : B \neq \mathbf{0}$. The game $\mathsf{G}(t, P)$ *avoids* t' if for every play $\pi \in \mathsf{G}(t, P)$, $t' \notin \pi(i)$ for all $i : 1 \leq i \leq |\pi|$.

Definition 7.2. Assume t' and t'' are trees whose roots are labelled with a constant or a variable. Let $t[t''/t']$ be the result of replacing the subtree t' of t with the tree t''.

The first transformation is straightforward: if no play enters a subtree of t then it can be replaced with the single node labelled with constant $d : \mathbf{0}$ (introduced in Remark 5.1).

T1 If $\mathsf{G}(t, P)$ avoids t' then transform t to $t[d/t']$

The second transformation removes inner tiles from t: if a basic tile is both j-end, Definition 6.8, and j-directed with respect to the game $\mathsf{G}(t, P)$, Definition 6.21, then it is redundant.

T2 If $\tau(\lambda\overline{x}_1, \ldots, \lambda\overline{x}_k)$ is a j-end basic tile and j-directed with respect to $\mathsf{G}(t, P)$, t' is the subtree of t rooted at τ and t_j is the subtree directly beneath $\lambda\overline{x}_j$ of τ then transform t to $t[t_j/t']$.

An application of **T2** not only removes the tile $\tau(\lambda\overline{x}_1, \ldots, \lambda\overline{x}_k)$ from t but also all subtrees that occur directly beneath any atomic leaf $\lambda\overline{x}_i$, $i \neq j$, of τ. Because τ has no j-dependents, all free variables that occur in the subtree t_j directly below $\lambda\overline{x}_j$ are bound above τ in t; therefore, the result of applying **T2** is still a closed term (in normal form with the right type). If τ is j-directed with respect to a game then each play involves a (unique) sequence of ri intervals which are shortest j-plays on τ, as described in the previous section. Game-theoretically underpinning the correctness of **T2** is *omission* of these inessential intervals from each play.

Proposition 7.3. *If* $\mathsf{i} \in \{\mathbf{1}, \mathbf{2}\}$, $t \, \mathsf{Ti} \, t'$ *and* $t \models P$ *then* $t' \models P$.

Proof. This is clear in the case of **T1**. Assume $t \models P$, $\tau(\lambda\overline{x}_1, \ldots, \lambda\overline{x}_k)$ is j-end and j-directed with respect to the game $\mathsf{G}(t, P)$, t'' is the subtree at the root of τ and t_j is the subtree directly beneath $\lambda\overline{x}_j$. Let $t' = t[t_j/t'']$. We show that $t' \models P$. We convert each $\pi \in \mathsf{G}(t, P)$ into a play $\pi' \in \mathsf{G}(t', P)$ that ends with the same final state. Because τ is j-directed with respect to each play, π can be split uniquely into the following regions

$$
\begin{array}{ccccccccc}
& t'' & & \lambda\overline{x}_j & & t'' & & \lambda\overline{x}_j & \\
\pi(1) & \ldots & \pi(i_1) & \ldots & \pi(m_1) & \ldots & \pi(i_n) & \ldots & \pi(m_n) & \ldots & \pi(|\pi|)
\end{array}
$$

where each $\pi(i_l, m_l)$ is a (shortest) j-play on τ and is ri; by definition of j-directed, node t'' cannot occur outside of these intervals. Therefore, $\pi(m_k)$ is a child of $\pi(i_k)$ for each k Consequently, the play $\pi' \in \mathsf{G}(t', P)$ is just the outer intervals $\pi(1, i_1 - 1), \pi(m_1 + 1, i_2 - 1), \ldots, \pi(m_n + 1, |\pi|)$ (modulo the changes to the entries in the look-up tables) because for each l, $\pi(m_l)$ is a child of $\pi(i_l)$. We show, that if s is a node in τ or is m-below an atomic leaf $\lambda\overline{x}_m$, $m \neq j$, of τ then s cannot occur in any outer interval of π. If s were to appear in such a position then move C4 must have applied: there is then a variable y and a position in an outer region $y \in \pi(n)$ and $\theta \in \pi(n)$ and $\theta(y) = l\xi i$ and there is a free variable z in l such that $\xi(z) = s\theta'i'$. However, this is impossible. Consider $\theta_1 \in \pi(i_1)$: clearly, there is no free variable in the subtree rooted at t'' with this property. When play reaches $\pi(m_1)$ because τ is a j-end tile and because the look-up tables in $\pi(m_1)$ extend those in $\pi(i_1)$ there cannot be a free variable in the subtree t_j with this property either. This argument is now repeated for subsequent positions $\pi(i_k)$ and $\pi(m_k)$. $\square$

The two transformations are also *reversible*: we can add gratuitous subtrees and intersperse redundant j-end tiles with arbitrary subtrees beneath their other atomic leaves in any solution term.

Example 7.4. Consider the 4th-order problem $xv = fa, xw = f(fa)$ where $v = \lambda y_1 y_2.y_1 y_2$ and $w = \lambda y_3 y_4.y_3(y_3 y_4)$ from Example 3.1. A solution term is in Figure 1. There are two plays, π in Figure 5 for the first equation and π' in Figure 8 for the second. We examine applications of **T2** to the term. The simple tile $z(\lambda u, \lambda)$ rooted at (6) is 1-end and 1-directed with respect to the game: there are three 1-plays on it which are all ri, $\pi(6, 7)$, $\pi'(6, 7)$ and $\pi'(12, 13)$. Transformation **T2** allows us to remove this tile, so the leaf node (8) is directly beneath node (5). The *basic* tile $z(\lambda s.s, \lambda)$ rooted at (14) is 1-end and 1-directed with respect to the game: the only plays $\pi(12, 15)$, $\pi'(18, 23)$ and $\pi'(26, 31)$ are ri. A second application of **T2** removes it; therefore, node (18) is directly beneath node (13). Consequently, the basic

(1) $q[(w), f(fa)]\, \theta_1\, \xi_1$

(2) $q[w, f(fa)]\, \theta_2\xi_2$ $\theta_2 = \theta_1\{w\xi_1 1/z\}$ $\xi_2 = \xi_1$ $A3$

(3) $q[(y_3y_4), f(fa)]\, \theta_3\xi_3$ $\theta_3 = \theta_2$ $\xi_3 = \xi_1\{(3)\theta_2 2/y_3, (11)\theta_2 2/y_4\}$ $C4$

(4) $q[-, f(fa)]\, \theta_4\xi_4$ $\theta_4 = \theta_2\{(y_3y_4)\xi_3 3/x\}$ $\xi_4 = \xi_3$ $A2$

(5) $q[(\ ), fa]\, \theta_5\, \xi_5$ $\theta_5 = \theta_4$ $\xi_5 = \xi_3$ $B1$

(6) $q[w, fa]\, \theta_6\, \xi_6$ $\theta_6 = \theta_4$ $\xi_6 = \xi_1$ $A3$

(7) $q[(y_3y_4), fa]\, \theta_7\, \xi_7$ $\theta_7 = \theta_4$ $\xi_7 = \xi_1\{(7)\theta_4 6/y_3, (9)\theta_4 6/y_4\}$ $C4$

(8) $q[y_3y_4, fa]\, \theta_8\, \xi_8$ $\theta_8 = \theta_4\{y_2\xi_7 7/u\}$ $\xi_8 = \xi_3$ $A3$

(3) $q[(y_4), fa]\, \theta_9\xi_9$ $\theta_9 = \theta_2$ $\xi_9 = \xi_3$ $C4$

(4) $q[-, fa]\, \theta_{10}\xi_{10}$ $\theta_{10} = \theta_2\{y_4\xi_3 9/x\}$ $\xi_{10} = \xi_3$ $A2$

(5) $q[(\ ), a]\, \theta_{11}\, \xi_{11}$ $\theta_{11} = \theta_{10}$ $\xi_{11} = \xi_3$ $B1$

(6) $q[w, a]\, \theta_{12}\, \xi_{12}$ $\theta_{12} = \theta_{10}$ $\xi_{12} = \xi_1$ $A3$

(7) $q[(y_3y_4), a]\, \theta_{13}\, \xi_{13}$ $\theta_{13} = \theta_{10}$ $\xi_{13} = \xi_1\{(7)\theta_{10}12/y_3, (9)\theta_{10}12/y_4\}$ $C4$

(8) $q[y_4, a]\, \theta_{14}\, \xi_{14}$ $\theta_{14} = \theta_{10}\{y_4\xi_{13}13/u\}$ $\xi_{14} = \xi_3$ $A3$

(11) $q[(\ ), a]\, \theta_{15}\, \xi_{15}$ $\theta_{15} = \theta_2$ $\xi_{15} = \xi_3$ $C4$

(12) $q[w, a]\, \theta_{16}\xi_{16}$ $\theta_{16} = \theta_2$ $\xi_{16} = \xi_1$ $A3$

(13) $q[(y_3y_4), a]\, \theta_{17}\xi_{17}$ $\theta_{17} = \theta_2$ $\xi_{17} = \xi_1\{(13)\theta_2 16/y_3, (19)\theta_2 16/y_4\}$ $C4$

(14) $q[w, a]\, \theta_{18}\xi_{18}$ $\theta_{18} = \theta_2\{(y_3y_4)\xi_{17}17/y\}$ $\xi_{18} = \xi_1$ $A3$

(15) $q[(y_3y_4), a]\, \theta_{19}\, \xi_{19}$ $\theta_{19} = \theta_{18}$ $\xi_{19} = \xi_1\{(15)\theta_{18}18/y_3, (17)\theta_{18}18/y_4\}$ $C4$

(16) $q[y_3y_4, a]\, \theta_{20}\, \xi_{20}$ $\theta_{20} = \theta_{18}\{(y_3y_4)\xi_{19}19/s\}$ $\xi_{20} = \xi_{19}$ $A3$

(15) $q[(y_3y_4), a]\, \theta_{21}\, \xi_{21}$ $\theta_{21} = \theta_{20}$ $\xi_{21} = \xi_{19}\{(15)\theta_{20}20/y_3, (17)\theta_{20}20/y_4\}$ $C4$

(16) $q[y_4, a]\, \theta_{22}\, \xi_{22}$ $\theta_{22} = \theta_{20}\{y_4\xi_{21}21/s\}$ $\xi_{22} = \xi_{21}$ $A3$

(17) $q[(\ ), a]\, \theta_{23}\, \xi_{23}$ $\theta_{23} = \theta_{22}$ $\xi_{23} = \xi_{21}$ $C4$

(18) $q[y_3y_4, a]\, \theta_{24}\, \xi_{24}$ $\theta_{24} = \theta_{22}$ $\xi_{24} = \xi_{17}$ $A3$

(13) $q[(y_4), a]\, \theta_{25}\xi_{25}$ $\theta_{25} = \theta_2$ $\xi_{25} = \xi_{17}$ $C4$

(14) $q[w, a]\, \theta_{26}\xi_{26}$ $\theta_{26} = \theta_2\{(y_3y_4)\xi_{17}25/y\}$ $\xi_{26} = \xi_1$ $A3$

(15) $q[(y_3y_4), a]\, \theta_{27}\, \xi_{27}$ $\theta_{27} = \theta_{26}$ $\xi_{27} = \xi_1\{(15)\theta_{26}26/y_3, (17)\theta_{26}26/y_4\}$ $C4$

(16) $q[y_3y_4, a]\, \theta_{28}\, \xi_{28}$ $\theta_{28} = \theta_{26}\{(y_3y_4)\xi_{27}27/s\}$ $\xi_{28} = \xi_{27}$ $A3$

(15) $q[(y_4), a]\, \theta_{29}\, \xi_{29}$ $\theta_{29} = \theta_{28}$ $\xi_{29} = \xi_{27}\{(15)\theta_{28}28/y_3, (17)\theta_{28}28/y_4\}$ $C4$

(16) $q[y_4, a]\, \theta_{30}\, \xi_{30}$ $\theta_{30} = \theta_{28}\{y_4\xi_{29}29/s\}$ $\xi_{30} = \xi_{29}$ $A3$

(17) $q[(\ ), a]\, \theta_{31}\, \xi_{31}$ $\theta_{31} = \theta_{30}$ $\xi_{31} = \xi_{29}$ $C4$

(18) $q[y_4, a]\, \theta_{32}\, \xi_{32}$ $\theta_{32} = \theta_{30}$ $\xi_{32} = \xi_{17}$ $A3$

(19) $q[(\ ), a]\, \theta_{33}\, \xi_{33}$ $\theta_{33} = \theta_2$ $\xi_{33} = \xi_{17}$ $C4$

(20) $q[\ \exists\ ]\, \theta_{34}\, \xi_{34}$ $\theta_{34} = \theta_2$ $\xi_{34} = \xi_{17}$ $A1$

Figure 8: Another play on the tree in Figure 1 from Example 7.4

tile $z(\lambda y.y, \lambda)$ rooted at (12) is also 1-end and 1-directed with respect to the "reduced" game; the plays $\pi(12, 19)$ and $\pi'(16, 33)$ become the sequences $\pi(12)\pi(13)\pi(16)\pi(19)$ and $\pi'(16)\pi'(17)\pi'(24)\pi'(25)\pi'(32)\pi'(33)$ (modulo changes to the look-up tables). The starting term in Figure 1 is, therefore, reduced to the smaller solution term $\lambda z.z(\lambda x.fx)a$. $\square$

Assume t is a 3rd-order term. If we inspect it top-down, starting beneath the initial lambda then it consists of simple tiles, each of which is either a constant tile or a top tile $y(\lambda_1, \ldots, \lambda_k)$, $k \geq 0$, where each atomic leaf is labelled with a dummy λ because y has order at most 2; therefore, it is also an end tile.

Fact 7.5. If t has order 3 and τ is a simple tile in t then either τ is a constant tile or an end tile which is also a top tile.

For instance, the tree in Figure 4 consists of four simple top tiles $y(\lambda)$ that are also end tiles rooted at (2), (4), (10) and (14) and the simple constant tiles $f(\lambda x z_1 z_2, \lambda)$, $x(\lambda)$, z_2 and a. Therefore, by repeated application of Fact 6.10 and Proposition 6.18 each play $\pi \in \mathsf{G}(t, P)$ when t is $3rd$-order merely descends a branch of t until it reaches a final state. We now examine this *tree model property* of plays in more detail and show how it leads to a very straightforward proof of decidability of 3rd-order matching.

Assume P is $3rd$-order. We define a (unique) partition of any play $\pi \in \mathsf{G}(t, P)$ in stages; at each stage we identify a simple tile, a subpart of t, and the interval at that stage. We call this iteratively defined notion of partition a p-partition (a "play partition") to distinguish it from Definition 6.14 of the b-partition for a position. (At 3rd-order, these partitions are intimately related as we shall note; in the next section we extend p-partitions to all orders and its definition uses b-partitions.)

Definition 7.6. Assume P is 3rd-order and $\pi \in \mathsf{G}(t, P)$. The *p-partition* of π is defined in stages $1 \le k \le n$ for some n as $\pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ where $j_0 = 1$. At each stage k there is

(1) the p-partition up to stage $k - 1$, $\pi(1, j_{k-1}) = \pi(j_0), \ldots, \pi(i_{k-1}, j_{k-1})$;
(2) the simple tile τ_k which occurs in t directly beneath node $t' \in \pi(j_{k-1})$;
(3) the composite tile γ_k of t consisting of all the nodes in the tiles $\tau_1, \ldots, \tau_k$;
(4) the position $\pi(i_k)$ with $t_k \in \pi(i_k)$ which is the root node of τ_k;
(5) the interval $\pi(i_k, j_k)$ determined as follows: j_k is the least $j > i_k$ such that
  • $t'' \in \pi(j)$ is an atomic leaf of τ_k, or
  • $j = |\pi|$.

The idea of a p-partition of $\pi \in \mathsf{G}(t, P)$ is to structurally relate parts of π to parts of t. At stage 1, τ_1 is the simple tile directly beneath the initial lambda of t which is either a constant or a top tile. The subpart of t at this stage, γ_1 is just τ_1. Assume t_1 is the root of τ_1; therefore, $t_1 \in \pi(i_1)$ (because $i_1 = 2$). Consider the interval $\pi(i_1, j_1)$: $\pi(j_1)$ is the first position such that either it is at an atomic leaf of τ_1 or it is the final position of the play. In the first case, $\pi(i_1, j_1)$ is a shortest play on τ_1; the tile τ_2 is the simple tile directly beneath the atomic leaf $t' \in \pi(j_1)$ of τ_1 and γ_2 is τ_1 and τ_2. The p-partition thereby continues: at stage k, the interval $\pi(i_k, j_k)$ is either a shortest play on τ_k or $j_k = |\pi|$ and then there are no further stages. The following is an easy consequence of Fact 6.10, Proposition 6.22 and the definition of b-partition.

Fact 7.7. Assume the *p-partition* of $\pi \in \mathsf{G}(t, P)$ is $\pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ and τ_k is the simple tile at stage k.

(1) For $k : 1 \le k < n$, $\pi(i_k, j_k)$ is a shortest play on τ_k.
(2) For $k : 1 \le k \le n$, $\pi(i_k, j_k)$ is internal to τ_k.
(3) For $k : 1 \le k < n$ the b-partition of $\pi(j_k) = \pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_k, j_k)$.

Example 7.8. If π is the play of Example 4.5 on the tree of Figure 4 then its p-partition is depicted below linearly.

$$
\begin{array}{cccccc}
y \quad \lambda & y \quad \lambda & f \quad \lambda x z_1 z_2 & x \quad \lambda & y \quad \lambda & z_2 \\
\pi(2,3) & \pi(4,5) & \pi(6,7) & \pi(8,9) & \pi(10,11) & \pi(12,12)
\end{array}
$$

The simple tiles at each stage are; $\tau_1 = y(\lambda)$, $\tau_2 = y(\lambda)$ rooted at (4), $\tau_3 = f(\lambda x z_1 z_2, \lambda)$, $\tau_4 = x(\lambda)$, $\tau_5 = y(\lambda)$ rooted at (10) and $\tau_6 = z_2$. For the other play π' in this example, there is the following p-partition.

$$
\begin{array}{ccccc}
y \quad \lambda & y \quad \lambda & f \quad \lambda & y \quad \lambda & a \\
\pi'(2,3) & \pi'(4,5) & \pi'(6,7) & \pi'(8,9) & \pi'(10,10)
\end{array}
$$

The two plays share the first three simple tiles, but play is at different atomic leaves of τ_3 after stage 3. $\qquad\qquad\square$

Consider the p-partitions of all plays in $\mathsf{G}(t, P)$. We slightly abuse notation: we let $\pi(i_k, j_k)$, $\pi'(i_k, j_k)$ be their intervals at stage k even when they have different ranges. Instead of a branch of simple tiles there is a tree of simple tiles: as each play shares the same simple tile τ_1 at stage 1 of its p-partition. The simple tile τ in t is *special* if it obeys one of the following three conditions

- $\tau = \tau_k$ for π at stage k and $\pi(i_k, j_k)$ is nri (see Definition 5.4),
- $\tau = \tau_k$ for π at stage k and $j_k = |\pi|$,
- $\tau = \tau_k$ for π, π' at stage k and $t' \neq t''$ when $t' \in \pi(j_k)$, $t'' \in \pi'(j_k)$.

The first kind of special tile explicitly contributes to solving P. The second kind identifies where a play finishes. The third kind separates plays; each p-partition after stage 1 that ends at the same atomic leaf of τ_1 shares τ_2 at stage 2 and so on. Therefore, branching in the tree of simple tiles will occur at τ_k if there are plays π, π' that end at different atomic leaves of τ_k at stage k (and agree on atomic leaves at all earlier stages). In Example 7.8, $f(\lambda x z_1 z_2, \lambda)$ separates the plays π and π'. The other special tiles in these plays are $x(\lambda)$ because of the nri interval $\pi(8,9)$ and z_2 and a as plays finish within them.

Any simple tile in t with at least one atomic leaf which is not special is *superfluous*. Either every play avoids it (so, **T1** applies) or every play on it is ri and ends at the same atomic leaf $\lambda \overline{x}_j$ for some j (so, is both j-end and j-directed with respect to the game and **T2** applies): four instance, all four simple tiles $y(\lambda)$ in Example 7.8 are both 1-end and 1-directed with respect to the game. There is an upper bound (relative to the problem P) on the number of special tiles that can occur in a term t as follows

- at most δ $(=$ the right size for P, Definition 3.8$)$ tiles that involve nri intervals;
- at most p $(=$ the number of plays[2]$)$ tiles where a play ends;
- at most $p - 1$ tiles that are play separators.

Decidability of 3rd-order matching, via the small model property, is, therefore, a simple consequence of the p-partitions and the identification of special simple tiles. For Example 7.8, the term of Figure 4 can be reduced to the smaller solution term $\lambda y. f(\lambda x z_1 z_2. x z_2) a$.

Definition 7.9. $|t|$ is the number of simple tiles with atomic leaves in a longest branch of t and $\|t\|$ is the total number of simple tiles with atomic leaves in t.

Fact 7.10. If t is a smallest solution to 3rd-order P and p is the number of plays in $\mathsf{G}(t, P)$, then $\|t\| \leq \delta + (2p - 1)$.

An alternative, and equally simple, proof of decidability of 3rd-order matching that does not appeal directly to the tree model property uses Proposition 6.24 and transformations **T1** and **T2**. In a large solution term, there must be embedded end tiles that are redundant. Dowek observes that solutions with embeddings $\lambda \overline{y}.(\ldots (y \ldots (y \ldots (y \ldots) \ldots) \ldots) \ldots)$ can

---

[2] p is bounded by the number of branches in the right terms of P.

be reduced to smaller solutions $\lambda \overline{y}.(\ldots (y \ldots) \ldots)$ in his proof of decidability of 3rd-order matching [4]. Assume t is a *smallest* solution term (with respect to $|| \cdot ||$) for P. It contains at most $\delta + p$ constant tiles with atomic leaves (otherwise, **T1** would apply and produce a smaller solution). Let c be the number of constant tiles with atomic leaves in t_0. Therefore, by Proposition 6.24, t contains at most $\lceil (\delta + p - c)/2 \rceil$ embedded simple top tiles (otherwise, **T2** would apply and produce a smaller solution). If α is the arity of P, Definition 3.6, then there are at most α inequivalent top simple tiles: as soon as a branch of t contains $\alpha + 1$ simple top tiles, there must be at least one embedded end tile. Consequently, no branch of t can contain more than $\alpha + (\lceil (\delta + p - c)/2 \rceil)$ simple top tiles. Because $c \leq \delta + p$, $|t| \leq \alpha + \delta + p + 1$ and 3rd-order matching is decidable.

The question is how to extend these straightforward proofs to all higher-orders. With a 4th or 5th-order tree there are two levels of simple non-constant tiles: top tiles $y(\lambda \overline{x}_1, \ldots, \lambda \overline{x}_k)$ and end tiles $z(\lambda \overline{z}_1, \ldots, \lambda \overline{z}_l)$ where z is bound by a $\lambda \overline{x}_j$. The number of levels increases with order: at 8th or 9th-order there are four levels. As soon as there is more than one level, game playing may jump around the tree as the examples in Figures 5 and 7 illustrate. For any order, if the tree-model property holds, so each play can be p-partitioned into internal plays on the simple tiles from the root to a tile where the play ends, then decidability is assured by iteration: redundant simple end tiles are first removed which causes further tiles to be end and so on.

Remark 7.11. Schubert defines an *unsophisticated* lambda term in [14]: $\lambda x_1 \ldots x_m.t$ is unsophisticated if each occurrence of $x_i t'_1 \ldots t'_k$ within t has the property that no t'_j contains a free variable x_l, $1 \leq l \leq m$. A 5th-order (dual) interpolation problem is unsophisticated if in each (dis)equation $x v_1 \ldots v_m \approx u$ the left terms v_i are unsophisticated. Schubert proves decidability of 5th-order unsophisticated dual interpolation. This restriction implies the tree model property. When play is at an end tile with head variable y, the state has the form $q[\lambda \overline{z}.u, r]$ where $\lambda \overline{z}.u$ is closed: consequently, play cannot jump back to the tile which binds y. (Decidability can be extended to all orders by defining *hereditary* unsophisticated terms where each occurrence of $\lambda \overline{y}.t'$ within it is unsophisticated.) $\qquad \square$

For arbitrary order, if t is a smallest solution term for P and it contains c simple constant tiles with atomic leaves then $c \leq \delta + p$. If t is large then it must contain embedded tiles. With 4th-order there must be embedded top tiles and with 5th-order there can also be embedded end tiles. However, Proposition 6.24 implies, for any order, that there cannot be more than $\lceil (\delta + p - c)/2 \rceil$ embedded end tiles. Consequently, if P is a 5th-order problem and t contains a bounded number k of top simple tiles then the small model property holds.

Fact 7.12. If t is a smallest solution to 5th-order P that contains at most k top simple tiles, then $|t| \leq (k \times (\alpha + 1)) + \delta + p + 1$

Fact 7.12 generalises to all orders: if t is a smallest solution to P that contains at most k simple tiles that are neither end nor constant tiles then it has a bounded size. This result slightly extends some cases examined in [3, 15] where there are restrictions on the number of free variables that a $\lambda \overline{y}$ can bind within a solution term.

A family of tiles in a tree consists of a top tile together with all its dependents (see Definition 6.5). The problem case for 4th or 5th-order is a solution term with arbitrary many top tiles (and, therefore, arbitrary many families). If a tree is large, then it must contain families of tiles all of whose plays are ri and, therefore, do not contribute to solving P. For even higher-orders, for the same reason, whole subfamilies of tiles are redundant.
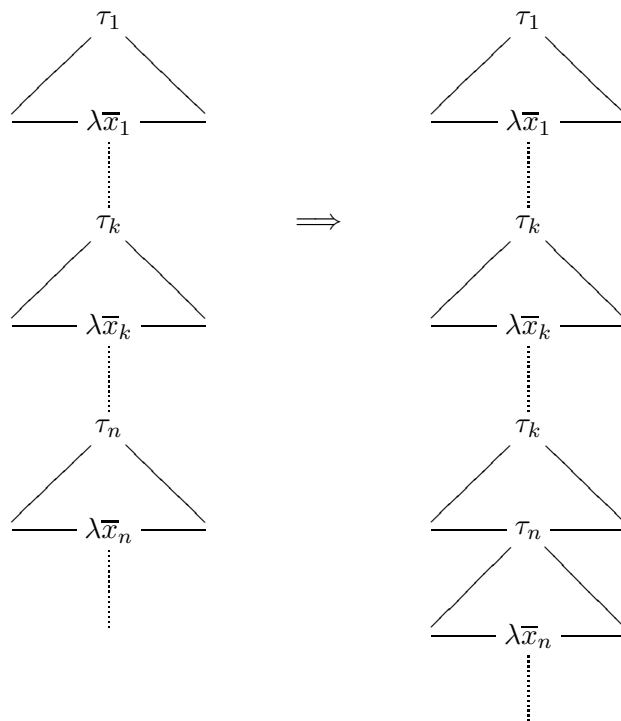
Figure 9: Tile Lowering

The difficulty is how to extract and remove redundant families and subfamilies of tiles. They may be entangled, occurring anywhere in a tree. What we would like to do is disentangle them thereby restoring, as far as possible, the tree model property. A key ingredient is *tile lowering* where we generalise to *basic* tiles. Consider the left tree in Figure 9 and assume τ_k is a basic tile and τ_n is a simple tile whose head variable is bound within τ_k (and no variable on the branch between $\lambda\overline{x}_k$ and τ_n is bound within τ_k). What we would like to do is to transform the left tree into the right tree where τ_k (and all the subtrees beneath its atomic leaves other than $\lambda\overline{x}_k$) is copied immediately above τ_n (thereby producing a larger basic tile) with the understanding that it is the lower occurrence of τ_k that binds τ_n and any free variables beneath τ_n that are bound within τ_k in the left tree. In [20], we introduced an explicit local transformation that has this effect for 4th-order matching and for the atoms case at all orders. The virtue of the transformation is that families of tiles are disentangled as the lower occurrence of τ_k is brought next to its dependent tile τ_n (and the upper occurrence of τ_k in the right tree is "closer" to being an end tile, as (some) binding is lost below). Game-theoretically, tile lowering will be justified in terms of permutations, omissions and repetitions of ri plays on tiles.

8. Partitioning of plays

The analysis now shifts from how single tiles to how families of tiles in a term contribute to solving a problem. We extend the definition of the p-partition of a play from the previous section to all orders. We start by examining simple properties of game playing that involve

families of tiles: the discussion uses the notion of when a position is a descendent of another position, Definition 5.17, when one tile is a dependent of another, Definition 6.4, and when two tiles belong to the same family, Definition 6.5.

Proposition 8.1. *Assume τ with root node t and $\gamma(\lambda\overline{x}_1, \ldots, \lambda\overline{x}_k)$ with root node t' are simple tiles.*

(1) If $t \in \pi(i)$, $t' \in \pi(j)$, $\pi(j)$ is the result of move A3 of Figure 3 and $\pi(j)$ is a descendent of $\pi(i)$ then γ is a dependent of τ.

(2) If $t \in \pi(i)$ and $\lambda\overline{x}_j \in \pi(i+1)$ and $\pi(i+1)$ is the result of move C4 then τ and γ belong to the same family of tiles.

Proof. (1) Assume that $t \in \pi(i)$, $t' \in \pi(j)$ and $\pi(j)$ is a descendent of $\pi(i)$. Therefore, because the descendent relation is the reflexive and transitive closure of the child relation, there is a subsequence of positions $\pi(i_1), \pi(i_2), \ldots, \pi(i_{2k+1})$ with $i = i_1$, $i_{2k+1} = j$, for all n, $\pi(i_{n+1})$ is a child of $\pi(i_n)$. Assume $t_m \in \pi(i_m)$ for each $m : 1 \leq m \leq 2k + 1$; so, $t = t_1$ and $t' = t_{2k+1}$. Because $\pi(j)$ is the result of A3, t' is labelled with a variable, that is bound by t_{2k}: therefore, γ is a dependent of the simple tile rooted at t_{2k-1}. This argument is now repeated because $\pi(i_{2k-1})$ must in turn be the result of A3 and so the simple tile rooted at t_{2k-1} is a dependent of the simple tile rooted at t_{2k-3}, and so on; consequently, as the dependency relation is transitive closed, γ is a dependent of τ. (2) Consider the following subsequences of positions $\pi(j_1), \pi(j_2), \pi(j_3), \ldots, \pi(j_{3k-2}), \pi(j_{3k-1}), \pi(j_{3k})$ where $i = j_{3k-1}$ and $i + 1 = j_{3k}$; each position $\pi(j_{3m})$ is the result of C4 on $\pi(j_{3m-1})$ which is the result of 0 or more applications of C3 on $\pi(j_{3m-2})$, $m \geq 1$; furthermore, for $m > 1$, $\pi(j_{3m-2})$ is the result of A3 and is a child of $\pi(j_{3m-3})$ and $\pi(j_1)$ is the result of A3 and a child of $\pi(1)$. Now by a routine induction on k, it follows that for every $m : 1 \leq m \leq k$, $t_m \in \pi(j_{3m})$ is an atomic leaf of some simple tile rooted at $t_n \in \pi(j_{3n} - 2)$ for some $n \leq m$ and that all such tiles belong to the same family. $\square$

In the previous section we alluded to the notion of tile level.

Definition 8.2. The *level* of a non-constant tile is defined inductively.

(1) If τ is a top tile then τ has level 1.
(2) If τ is an immediate dependent of γ and γ has level m then τ has level $m + 1$.

The number of possible levels increases with order. In a 4th or 5th-order term there are at most two levels of tile: top tiles and end tiles as illustrated in Figures 1 and 6. With a 8th or 9th-order term, there are at most four levels of tile and so on. The presence of dummy lambda in a term tree does not affect the notion of level (because a dummy lambda cannot be a binder).

If there is more than one level of tile in a term, game playing may pass repeatedly through the same sequences of nodes of a tree. The following definition captures when two such intervals are said to correspond.

Definition 8.3. The intervals $\pi(i,j)$, $\pi(i',j')$ *correspond*, written $\pi(i,j) \sim \pi(i',j')$, provided that $j, j' < |\pi|$, $j - i = j' - i'$ and for all $k : 0 \leq k \leq j - i$,

(1) if $t \in \pi(i+k)$ then $t \in \pi(i'+k)$.
(2) if $q[l, r] \in \pi(i+k)$ then for some r', $q[l, r'] \in \pi(i'+k)$
(3) if $q[(l_1, \ldots, l_m), r] \in \pi(i+k)$ then for some r', $q[(l_1, \ldots, l_m), r'] \in \pi(i'+k)$.
(4) if $q[-, r] \in \pi(i+k)$ then for some r', $q[-, r'] \in \pi(i'+k)$.

Fact 8.4. The relation $\sim$ on intervals is an equivalence relation.

Although this definition abstracts from the look-up tables, it requires agreement on the sequences of nodes of the tree and on the left terms of states. For instance, intervals that are shortest ri plays on a top tile correspond, as shown in the proof of Proposition 6.22. When π' is the play in Figure 8 on the tree in Figure 1, the nri intervals $\pi'(4, 7)$ and $\pi'(10, 13)$ correspond.

In Section 6 we described some uniformity properties of game playing for top and embedded tiles (especially for embedded end tiles). Now our aim is to understand uniformities of play for tiles of arbitrary level. Given a position $\pi(j)$ at a lambda node t', there is an associated (unique) b-partition of $\pi(1, j)$ into intervals that are plays on the simple tiles between the root of the term tree and node t' by Definition 6.14. If more than one position is at a lambda node t' then their associated b-partitions must differ in their plays at an earliest tile with the same starting positions but different end positions.

Proposition 8.5. *Assume* $t' \in \pi(j)$, $t' \in \pi(j')$, $j \neq j'$ *and* t' *is labelled* $\lambda \overline{x}$ *for some* $\overline{x}$. *If* $\pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ *is the b-partition for* $\pi(j)$ *and* $\pi(j_0'), \pi(i_1', j_1'), \ldots, \pi(i_n', j_n')$ *is the b-partition for* $\pi(j')$ *then there is a* $k : 1 \leq k \leq n$ *such that* $i_k = i_k'$ *and* $j_k \neq j_k'$ *and for all* $m < k$, $i_m = i_m'$ *and* $j_m = j_m'$.

Proof. Assume that $\lambda \overline{x}_0, u_1, \lambda \overline{x}_1, \ldots, u_n, \lambda \overline{x}_n$ is the branch from the root of the term tree to t' labelled $\lambda \overline{x}_n$, $t' \in \pi(j)$ and $t' \in \pi(j')$ for $j' \neq j$. Let $\pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ be the b-partition for $\pi(j)$ and $\pi(j_0'), \pi(i_1', j_1'), \ldots, \pi(i_n', j_n')$ be the b-partition for $\pi(j')$. Because $j_0 = j_0' = 1$, $i_1 = i_1' = 2$ and $j \neq j'$, there must be a least k such that $i_k = i_k'$ and $j_k \neq j_k'$. $\square$

The b-partitions of Example 6.15 illustrate this proposition; node (7) of Figure 6 labelled λ belongs to both $\pi(15)$ and $\pi(31)$ of the play in Figure 7. These b-partitions agree on the play $\pi(2, 3)$ on the initial tile $z(\lambda z_1, \lambda z_2)$ that ends at λz_1 and then they have different 1-plays $\pi(4, 5)$ and $\pi(4, 21)$ on the next tile $z(\lambda x_1, \lambda x_2)$.

Definition 8.6. Assume $t' \in \pi(j)$, $t' \in \pi(j')$, $j \neq j'$, t' is labelled $\lambda \overline{x}$ for some $\overline{x}$ and $\lambda \overline{x}_0, u_1, \lambda \overline{x}_1, \ldots, u_n, \lambda \overline{x}_n$ is the branch from the root of the term tree to t'. Let $\pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ be the b-partition for $\pi(j)$ and let $\pi(j_0'), \pi(i_1', j_1'), \ldots, \pi(i_n', j_n')$ be the b-partition for $\pi(j')$ and let $\tau = u_k(\ldots \lambda \overline{x}_k \ldots)$ be the first simple tile for $k \geq 1$ such that $i_k = i_k'$ and $j_k \neq j_k'$. The positions $\pi(j), \pi(j')$ are then said to *vary at* $\pi(j_k), \pi(j_k')$ *with (simple tile)* τ.

Two positions vary at $\pi(j_k), \pi(j_k')$ with τ if they are at the same lambda node and τ is the first simple tile in their b-partitions where there is a difference in play; there are two distinct m-plays on τ for some m, $\pi(i_k, j_k)$ and $\pi(i_k, j_k')$. In the case of Example 6.15 discussed previously, the positions $\pi(15), \pi(31)$ vary at $\pi(5), \pi(21)$ with $z(\lambda x_1, \lambda x_2)$.

Given two positions $\pi(j)$ and $\pi(j')$ at the same node labelled $\lambda \overline{x}$ we are interested in defining when intervals $\pi(j + 1, j + m)$ and $\pi(j' + 1, j' + m)$ correspond in the sense of Definition 8.3. The simplest case is when $\pi(j), \pi(j')$ vary at $\pi(j), \pi(j')$ with $\tau = u_n(\ldots \lambda \overline{x} \ldots)$; their b-partitions agree except on the last simple tile in the branch from the root to $\lambda \overline{x}$. The two intervals $\pi(i_n, j)$ and $\pi(i_n, j')$ are, therefore, both k-plays on τ for some k; both positions $\pi(j)$ and $\pi(j')$ are children of $\pi(i_n)$ and, therefore, must be the results of move C4 of Figure 3. This means that the look-up tables $\theta_{j+1} \in \pi(j + 1)$ and $\theta_{j'+1} \in \pi(j' + 1)$ only differ in their entries for the variables in $\overline{x}$: therefore, for each $m \geq 1$ their continuations $\pi(j + 1, j + m), \pi(j' + 1, j' + m)$ correspond as long as play does not reach children of

$\pi(j)$ and $\pi(j')$; or positions where different $\forall$ choices are exercised; or a position with a final state. Consider next the general situation when $\pi(j), \pi(j')$ vary at $\pi(j_k), \pi(j'_k)$ with $\tau = u_k(\ldots \lambda \overline{x}_k \ldots)$; their b-partitions differ at a simple tile that is earlier in the branch from the root to $\lambda \overline{x}$. What we want to capture is the following uniformity: if the plays $\pi(i_m, j_m)$ and $\pi(i'_m, j'_m)$ on tiles $\tau_m : k < m \leq n$ that do not involve positions that are children of $\pi(j_k)$ and $\pi(j'_k)$ correspond, then the continuations from $\pi(j+1)$ and $\pi(j'+1)$ will also correspond as long as the positions are not descendents of $\pi(j_k)$ or $\pi(j'_k)$ or the result of a different $\forall$-choice, or one has a final state. The issue is how to formally capture this correspondence in terms of the relationship between the look-up tables $\theta \in \pi(j+1)$ and $\theta' \in \pi(j'+1)$. This is the motivation for the following bisimulation like definition.

Definition 8.7. Assume τ is a simple non-constant tile.

(1) Two look-up tables μ, μ' are *n-similar except for τ*, $\mu \sim^n_\tau \mu'$, which is defined iteratively, for $n \geq 0$.
  - $\mu \sim^0_\tau \mu'$ iff $\mu = \mu'$;
  - $\theta \sim^{n+1}_\tau \theta'$ iff (1) for all y. $\theta(y)$ is defined iff $\theta'(y)$ is defined and (2) if $\gamma = y(\ldots)$ is not a dependent of τ and $\theta(y) = l\xi i$ then $\theta'(y) = l\xi' i'$ and $\xi \sim^n_\tau \xi'$;
  - $\xi \sim^{n+1}_\tau \xi'$ iff (1) for all z. $\xi(z)$ is defined iff $\xi'(z)$ is defined and (2) if $\xi(z) = t\theta i$ then $\xi'(z) = t\theta' i'$ and $\theta \sim^n_\tau \theta'$.

(2) Two look-up tables μ, μ' are *similar except for τ*, $\mu \sim_\tau \mu'$, if there is an $n \geq 0$ such that $\mu \sim^n_\tau \mu'$.

Fact 8.8. In the following assume μ, μ' are look-up tables (of the same kind).

(1) $\mu \sim^n_\tau \mu$.
(2) If $\mu \sim^n_\tau \mu'$ then $\mu' \sim^n_\tau \mu$.
(3) If $\mu \sim^n_\tau \mu'$ and $\mu' \sim^n_\tau \mu''$ then $\mu \sim^n_\tau \mu''$.
(4) If $\mu \sim^m_\tau \mu'$ and $n > m$ then $\mu \sim^n_\tau \mu'$.

The key point with the definition of $\theta \sim_\tau \theta'$ is that the entries of θ and θ' should be very similar except in the case that they are labels of nodes of simple tiles that are dependents of τ.

Fact 8.9. If $\pi(i, j)$ and $\pi(i, j')$ are k-plays on the simple tile τ, $\theta \in \pi(j+1)$ and $\theta' \in \pi(j'+1)$ then $\theta \sim_\tau \theta'$.

We now come to a critical uniformity property.

Proposition 8.10. *Assume $\pi(j), \pi(j')$ vary at $\pi(j_k), \pi(j'_k)$ with τ, $\theta \in \pi(j+1)$, $\theta' \in \pi(j'+1)$ and $\theta \sim_\tau \theta'$. If $\pi(j+1, j+m)$ is ri, $t' \in \pi(j+m)$ is labelled $\lambda \overline{x}$ for some $\overline{x}$ and no $\pi(j+l)$, $1 \leq l \leq m$, is a descendent of $\pi(j_k)$ then $\pi(j'+1, j'+m) \sim \pi(j+1, j+m)$.*

Proof. Assume that $\lambda \overline{x}_0, u_1, \lambda \overline{x}_1, \ldots, u_n, \lambda \overline{x}_n$ is the branch from the root of term tree t to $\lambda \overline{x}_n$, $\lambda \overline{x}_n \in \pi(j)$ and $\lambda \overline{x}_n \in \pi(j')$ for $j' \neq j$. Let $\pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ be the b-partition for $\pi(j)$ and $\pi(j'_0), \pi(i'_1, j'_1), \ldots, \pi(i'_n, j'_n)$ be the b-partition for $\pi(j')$. Assume that $\pi(j), \pi(j')$ vary at $\pi(j_k), \pi(j'_k)$ with $\tau = u_k(\ldots \lambda \overline{x}_k \ldots)$ and that $\theta \sim_\tau \theta'$ when $\theta \in \pi(j+1)$ and $\theta' \in \pi(j'+1)$. Assume that $\pi(j+1, j+m)$ is ri, $t' \in \pi(j+m)$ is labelled $\lambda \overline{x}$, no $t'' \in \pi(i+k)$ within this interval is at a dependent of τ. By a routine induction on m, $\pi(j'+1, j'+m) \sim \pi(j+1, j+m)$. The interval $\pi(j+1, j+m)$ only involves moves A3 and C4 of Figure 3 because it is ri and ends at $t' \in \pi(j+m)$ by Fact 5.5. Initially, for some simple non-constant tile $\tau' = y(\lambda \overline{z}_1, \ldots, \lambda \overline{z}_l)$, $y \in \pi(j+1)$ and $y \in \pi(j'+1)$. Therefore,

because $\pi(j+1)$ is not a descendent of $\pi(j_k)$, τ' is not a dependent of τ; by A3 of Figure 3 position $\pi(j+1) = yq[l, r]\theta\xi_{j+1}$ and position $\pi(j'+1) = yq[l, r']\theta'\xi_{j'+1}$ where $\xi_{j+1} \sim_\tau \xi_{j'+1}$. The next position is a result of C4: if $\xi_{j+1}(x) = t'\theta''i$ then $\xi_{j'+1}(x) = t'\theta'''i'$ and $\theta'' \sim_\tau \theta'''$. Consequently, the result follows. $\square$

Remark 8.11. Proposition 8.10 can be strengthened to include corresponding nri plays that involve the same $\forall$-choices. For instance, in Example 6.15 when π is in Figure 7 the positions $\pi(5), \pi(21)$ vary at $\pi(5), \pi(21)$ with $z(\lambda x_1, \lambda x_2)$. Position $\pi(16)$ is the first position after $\pi(5)$ that is a child of $\pi(5)$; therefore, the continuations $\pi(6, 15)$ and $\pi(22, 31)$ correspond even though they pass through the constant tile $h(\lambda)$. $\square$

We shall now extend the definition of the p-partition of a play $\pi \in \mathsf{G}(t, P)$, Definition 7.6, from 3rd-order P to all higher orders. Again, it is defined in stages using simple tiles from t; so $\pi = \pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ for some n and there is an associated sequence of simple tiles $\tau_1, \ldots, \tau_n$ from t such that for each $m : 1 \leq m \leq n$, $t' \in \pi(i_m)$ where t' is the root of τ_m and τ_m occurs directly below $t'' \in \pi(j_{m-1})$ in t. However, the same simple tile may occur more than once in this sequence of tiles; so we adopt the following notation.

Definition 8.12. Assume that $\tau_1, \ldots, \tau_n$ is a sequence of simple tiles associated with a play $\pi \in \mathsf{G}(t, P)$. We write τ_k^π to identify the kth tile τ_k in this sequence and we use the notation $t'@\tau_k^\pi$ for node t' of τ_k^π.

After stage k, the p-partition for π consists of the sequence of tiles $\tau_1^\pi, \ldots, \tau_k^\pi$; the composite tile γ_k consists of these tiles. However, we also provide a more graphical representation of the composite tile γ_k by including a labelled edge for each $m : 1 < m \leq k$ of the form $\tau_m^\pi \xleftarrow{\pi} t'@\tau_l^\pi$ where $l < m$ and t' is an atomic leaf of τ_l^π. The relation $\xleftarrow{\pi}$ between a tile at stage m and an atomic leaf of a tile at an earlier stage l captures control structure in the p-partition that $t'@\tau_l^\pi \in \pi(j_{m-1})$. This linearisation of the p-partition for each π will be useful in the decidability proof; we can, of course, reconstitute the subtree γ_k from its linear presentation after stage k just by viewing $\xleftarrow{\pi}$ as the subtree relation. We assume this extra intensionality is also reflected in the ξ look-up tables for π; in the case of the C moves of Figure 3 if $l = \lambda z_1 \ldots z_j.w$ and $t_m@\tau_k^\pi \downarrow_i t'@\tau_k^\pi$ then $\xi_{m+1} = \xi_m\{(t_1'@\tau_k^\pi)\theta_m m/z_1 \ldots, (t_j'@\tau_k^\pi)\theta_m m/z_j\}$. Consequently, at an application of move C4, if x is the head variable of the left term of the state and $\xi(x) = (t'@\tau_k^\pi)\theta'i$ then the next position is at $t'@\tau_k^\pi$.

Definition 8.13. Assume $\pi \in \mathsf{G}(t, P)$. The *p-partition* of π is defined in stages $1 \leq k \leq n$ for some n as $\pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ where $j_0 = 1$. At each stage k there is
(1) the p-partition up to stage $k - 1$, $\pi(1, j_{k-1}) = \pi(j_0), \ldots, \pi(i_{k-1}, j_{k-1})$;
(2) the composite tile γ_{k-1} consisting of the tiles $\tau_1^\pi, \ldots, \tau_{k-1}^\pi$ with edges $\xleftarrow{\pi}$;
(3) the simple tile τ_k^π which occurs in t directly beneath node $t'@\tau_l^\pi \in \pi(j_{k-1})$ and $\tau_k^\pi \xleftarrow{\pi} t'@\tau_l^\pi$ in $\gamma_k = \gamma_{k-1} \cup \{\tau_k^\pi\}$;
(4) the position $\pi(i_k)$ with $t''@\tau_k^\pi \in \pi(i_k)$ where t'' is the root node of τ_k;
(5) the interval $\pi(i_k, j_k)$ determined as follows:
  • set $j = i_k$
  • ($\ast\ast$) while $j \neq |\pi|$ and $t'@\tau_l^\pi \in \pi(j)$ is not labelled with a lambda do j = j+1;
    − if $j = |\pi|$ or $t'@\tau_k^\pi \in \pi(j)$ then $j_k = j$;

$$\tau_1 = z(\lambda x, \lambda) \qquad \pi'(2,3)$$
$$\tau_2 = f(\lambda) \qquad \pi'(4,5) \qquad \tau_2 \xleftarrow{\pi'} \lambda x @ \tau_1$$
$$\tau_3 = z(\lambda u, \lambda) \qquad \pi'(6,7) \qquad \tau_3 \xleftarrow{\pi'} \lambda @ \tau_2$$
$$\tau_4 = x \qquad \pi'(8,15) \qquad \tau_4 \xleftarrow{\pi'} \lambda u @ \tau_3$$
$$\tau_5 = z(\lambda y, \lambda) \qquad \pi'(16,17) \qquad \tau_5 \xleftarrow{\pi'} \lambda @ \tau_1$$
$$\tau_6 = z(\lambda s, \lambda) \qquad \pi'(18,19) \qquad \tau_6 \xleftarrow{\pi'} \lambda y @ \tau_5$$
$$\tau_7 = s \qquad \pi'(20,23) \qquad \tau_7 \xleftarrow{\pi'} \lambda s @ \tau_6$$
$$\tau_8 = y \qquad \pi'(24,33) \qquad \tau_8 \xleftarrow{\pi'} \lambda @ \tau_6$$
$$\tau_9 = a \qquad \pi'(34,34) \qquad \tau_9 \xleftarrow{\pi'} \lambda @ \tau_5$$

Figure 10: Partition of π' from Figure 8 in Example 8.14

- find a largest $h \geq 0$, if there is one, such that there is a $j' < j$, $\pi(j), \pi(j')$ vary at $\pi(l), \pi(l')$ with τ_m^π in the same family as τ_k^π and $\theta \sim_{\tau_m^\pi} \theta'$ where $\theta \in \pi(j+1)$, $\theta' \in \pi(j'+1)$, and
  * no position in $\pi(j', j'+h)$ is a descendent of $\pi(l')$;
  * if $h > 0$ then $\pi(j+1, j+h) \sim \pi(j'+1, j'+h)$;
- if there is such a $h \geq 0$ set $j = j + (h+1)$ and goto (**) else $j_k = j$.

As with Definition 7.6, if τ_k^π is a top or a constant tile then $\pi(i_k, j_k)$ is internal to it and either ends at one of its atomic leaves, or a final state is reached: via clause (**), j_k will be the least $j > i_k$ such that $t' \in \pi(j)$ is an atomic leaf of τ_k^π or $j = |\pi|$. The new case is when $\tau_k^\pi = y(\lambda \overline{x}_1, \ldots, \lambda \overline{x}_m)$ is neither a constant tile nor a top tile. Position $\pi(i_k)$ is at the root of τ_k^π; the first position $\pi(j)$, if there is one, after $\pi(i_k)$ that is at a lambda node t' need not be internal to τ_k^π; however t' must be an atomic leaf of some tile $\tau_{k'}^\pi$, $k' \leq k$, which belongs to the same family as τ_k^π by Proposition 8.1. If $k' = k$ then $j_k = j$ and $\pi(i_k, j_k)$ finishes at t'. Otherwise $k' \neq k$ and $t'@\tau_{k'}^\pi \in \pi(j)$: one checks whether there are previous positions $\pi(j')$ such that $\pi(j), \pi(j')$ vary at $\pi(l), \pi(l')$ with τ_m^π in the same family as τ_k^π (and, therefore, also, $\tau_{k'}^\pi$) and whether $\theta \sim_{\tau_m^\pi} \theta'$ when $\theta \in \pi(j+1)$ and $\theta' \in \pi(j'+1)$. If there are no such positions, for instance if t' is an atomic leaf of the composite tile γ_k, then $j_k = j$ and $\pi(i_k, j_k)$ finishes at t'. Otherwise there are such positions; we then look for a longest continuation from $\pi(j+1)$ that corresponds to a previous interval from such a $\pi(j'+1)$. If there is no such continuation, so $h = 0$, then $\pi(j+1)$ is at a descendent of $\pi(l)$; so control is at the root of a tile in γ_k in the same family as τ_k^π; and the loop starts again. If there is such a continuation then $h > 0$ and the loop starts again from $\pi(j + (h+1))$; if $\pi(i_k, j_k)$ is ri then $\pi(j + (h+1))$ will also be at the root of a tile in γ_k in the same family as τ_k^π as this position must be a descendent of $\pi(l)$. Consequently, as we shall prove, if $\pi(i_k, j_k)$ is ri then it is guaranteed to finish at an atomic leaf of a tile in the same family as τ_k^π.

Example 8.14. We describe the p-partition of π' in Figure 8 for the term tree in Figure 1 (omitting the initial move). We present tile $\tau_k^{\pi'}$, without its superscript π', the play $\pi'(i_k, j_k)$ and the edge relation $\xleftarrow{\pi'}$ in Figure 10. The intervals for the first three stages are plays on top and constant tiles. Tile $\tau_4 = x$: play jumps at position $\pi'(8)$ to λx of τ_1; the positions

$$\tau_{n'}$$
$$\vdots$$
$$\tau_{m'}$$
$$\vdots$$
$$\tau_k$$

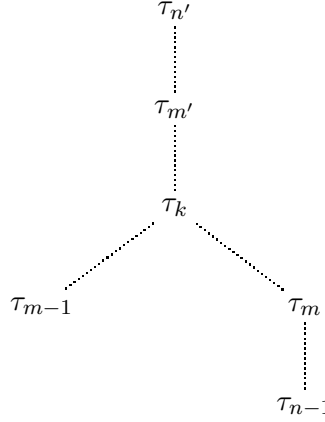$$\tau_{m-1} \qquad\qquad \tau_m$$

$$\tau_{n-1}$$

Figure 11: Illustrating repeating tiles in a p-partition

$\pi'(9),\pi'(3)$ vary at $\pi'(9),\pi'(3)$ with τ_1 and $\theta \sim_{\tau_1} \theta'$ when $\theta \in \pi'(10)$ and $\theta' \in \pi'(4)$; there are the maximal corresponding intervals $\pi'(4,7)$ and $\pi'(10,13)$. Play returns to x and jumps to $\lambda@\tau_1$ which completes stage 4 as $\pi'(15)$ is the first position at this atomic leaf. The p-partition continues with plays on the top tiles τ_5 and τ_6. After stage 6, the subtree γ_6 of Figure 1 associated with the partition consists of tiles $\tau_1-\tau_6$ plus the edges already described (which has atomic leaves $(9)\lambda$, $(19)\lambda$, $(15)\lambda s$ and $(17)\lambda$). Tile $\tau_7 = s$ and $\pi'(21),\pi'(19)$ vary at $\pi'(21),\pi'(19)$ with τ_6 and $\theta \sim_{\tau_6} \theta'$ when $\theta \in \pi'(22)$, $\theta' \in \pi'(20)$; however, there are no corresponding continuations because $\pi'(22)$ and $\pi'(20)$ are children of $\pi'(21)$ and $\pi'(19)$. The interval at stage 7 is, therefore, $\pi'(20,23)$ that finishes at $\lambda@\tau_6$. Tile $\tau_8 = y$ and the play at this stage jumps to λy of τ_5; positions $\pi'(25)$, $\pi'(17)$ vary at $\pi'(25)$, $\pi'(17)$ with τ_5 and $\theta \sim_{\tau_5} \theta'$ where $\theta \in \pi'(26)$ and $\theta' \in \pi'(18)$; $\pi'(26,31)$ corresponds to $\pi'(18,23)$ and then play returns to τ_8 and jumps to $\lambda@\tau_5$; so, $\pi'(24,33)$ is the interval at stage 8. Finally, $\tau_9 = a$ and stage 9 is $\pi'(34,34)$. □

A reason that the same tile may be repeated in a p-partition starts with different $\forall$-choices. For instance, consider a situation where $\tau_m \xleftarrow{\pi} t'@\tau_k$ and $\tau_n \xleftarrow{\pi} t'@\tau_k$ as follows where τ_k is a constant tile with arity 2, $f(\lambda,\lambda)$. Consider Figure 11: the p-partition for π has a play on $\tau_{n'}$, then $\tau_{m'}$ and then τ_k choosing the left branch of f. Tile τ_{m-1} is a dependent of $\tau_{m'}$ and play jumps to its atomic leaf above τ_k and so this position with the earlier one will vary at themselves at $\tau_{m'}$; play thus proceeds to τ_k; but now there is a different $\forall$-choice; so at stage $m-1$, play finishes at the other leaf of τ_k. Play now proceeds through τ_m to τ_{n-1} which is a dependent of $\tau_{n'}$; it jumps to the atomic leaf of $\tau_{n'}$ above $\tau_{m'}$ and again it varies with the earlier position at themselves at $\tau_{n'}$: instead of the corresponding play which passes through τ_k twice, play descends to τ_k and now the $\forall$-choice is the right branch; therefore, play after stage $n-1$ also finishes at $\lambda@\tau_k$ of the second branch. These different $\forall$-choices involve stages of a p-partition that have nri intervals; if an interval is ri then it is well-behaved as Proposition 8.16 shows.

Definition 8.15. Assume $\pi(j_0),\pi(i_1,j_1),\ldots,\pi(i_n,j_n)$ is the p-partition of $\pi \in \mathsf{G}(t,P)$ and $\tau_1^\pi,\ldots,\tau_n^\pi$ is the associated sequence of tiles. The π-*path for* τ_k^π is the sequence of tiles $\tau_{m_1}^\pi,\ldots,\tau_{m_l}^\pi$ such that $m_1 = 1$, $\tau_{m_l}^\pi = \tau_k^\pi$ and $\tau_{m_{j+1}}^\pi \xleftarrow{\pi} t_j@\tau_{m_j}^\pi$ for $1 \le j < l$.

Proposition 8.16. *Assume* $\pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ *is the p-partition of* $\pi \in \mathsf{G}(t, P)$ *and* $\tau_1^\pi, \ldots, \tau_n^\pi$ *is the associated sequence of tiles.*

(1) *If* $m < n$ *and* τ_m^π *is a top or constant tile then* $\tau_{m+1}^\pi \xleftarrow{\pi} t'@\tau_m^\pi$ *for some* t'.

(2) *If* $\pi(i_m, j_m)$ *is ri and* $\tau_{m+1}^\pi \xleftarrow{\pi} t'@\tau_k^\pi$ *then* τ_k^π *belongs to the same family as* τ_m^π.

(3) *If* $\pi(i_m, j_m)$ *is ri and* τ_m^π *is an embedded tile in* t *then* $\tau_{m+1}^\pi \xleftarrow{\pi} t'@\tau_m^\pi$ *for some* t'.

(4) *If* $\pi(i_m, j_m)$ *is ri and* $\tau_{m+1}^\pi \xleftarrow{\pi} t'@\tau_k^\pi$ *then* $t'@\tau_k^\pi$ *is not in the* π*-path of* τ_m.

Proof. (1) is clear from Definition 8.13. For (2) assume $\tau_{m+1}^\pi \xleftarrow{\pi} t'@\tau_k^\pi$ and $\pi(i_m, j_m)$ is ri. Consider the first position $\pi(j)$ in this interval at a lambda node. By Proposition 8.1, it is an atomic leaf of a tile in the same family as τ_m^π. Either this position is $\pi(j_m)$ and so the result follows, or there are earlier positions $\pi(j')$ and $t' \in \pi(j')$ and $\pi(j), \pi(j')$ vary at $\pi(l), \pi(l')$ at τ' in the same family as τ_m^π and $\theta \sim_{\tau'} \theta'$ when $\theta \in \pi(j+1)$ and $\theta' \in \pi(j'+1)$. One chooses the j' that allows longest corresponding intervals $\pi(j+1, j+h)$ and $\pi(j'+1, j'+h)$ for $h \geq 0$. Using Proposition 8.10, it follows that $\pi(j + (h+1))$ is a descendent of $\pi(l)$ and so is at a tile τ'' in the same family as τ_k^π; the same argument is now repeated. For (3), consider the b-partition for $\pi(j_{m-1})$; because τ_m^π is an embedded tile, there is a tile τ_k^π such that $\tau_k^\pi \equiv \tau_m^\pi$ and τ_m^π is below τ_k^π in t and in the b-partition there is a play on τ_k^π, $\pi(i_k', j_k')$; consider the shortest interval $\pi(i_k', i_k' + j')$ that is a play on τ_k^π. Using Proposition 6.23 and Definition 8.13 it follows that $\pi(i_m, j_m) = \pi(i_m, i_m + j')$ as $t' \in \pi(i_m + j')$ is an atomic leaf of τ_m^π. In the case of (4) assume $\pi(i_m, j_m)$ is ri and $\tau_{m+1}^\pi \xleftarrow{\pi} t'@\tau_k^\pi$ and $t'@\tau_k^\pi$ is in the π-path of τ_m^π; so, τ_k^π and τ_m^π belong to the same family by (2) because $\pi(i_m, j_m)$ is ri. Also, there must be a tile τ_l^π, $k < l \leq m$ such that $\tau_l^\pi \xleftarrow{\pi} t'@\tau_k^\pi$ and either $\tau_m^\pi = \tau_l^\pi$ or τ_l^π is on the π-path for τ_m^π. If τ_k^π is a top tile then consider the b-partition for $\pi(j_{m-1})$ and its interval that is a play on τ_k^π that ends at $t'@\tau_k^\pi$ at position $\pi(j')$; clearly, if position $\pi(j)$ in $\pi(i_m, j_m)$ is at $t'@\tau_k^\pi$ then $\pi(j), \pi(j')$ vary at $\pi(j), \pi(j')$ at τ_k^π which means that $j < j_m$ and so it is impossible that $\tau_{m+1}^\pi \xleftarrow{\pi} t'@\tau_k^\pi$. Otherwise, τ_k^π is a dependent tile; the argument is now similar but more general; if position $\pi(j)$ in $\pi(i_m, j_m)$ is at $t''@\tau_l^\pi$ for τ_l^π that is τ_k^π or a tile that τ_k^π is a dependent of and $t''@\tau_l^\pi$ is in the π-path for τ_m^π then there is a tile $\tau_{m'}^\pi$ such that $\tau_{m'}^\pi$ is a dependent of τ_k^π or of some tile that τ_k^π is a dependent of and the b-partition for $\pi(j_{m'-1})$ which involves a play on τ_l^π is such that it finishes at t'' at position $\pi(j')$ and $\pi(j), \pi(j')$ vary at $\pi(j_1), \pi(j_1')$ at $\tau_{l'}^\pi$ in the same family as τ_m^π; therefore, $\pi(j)$ cannot be a final position for $\pi(i_m, j_m)$. $\qquad\square$

Example 8.17. The p-partition of π in Figure 7 where the term tree is in Figure 6 is presented in Figure 12 (omitting the initial move); where we provide the tile τ_k^π, omitting π, the interval $\pi(i_k, j_k)$ and the edge relation at each stage k. Play proceeds through the top tiles τ_1 and τ_2. Tile $\tau_3 = (6)z_1(\lambda)$ and play at this stage is $\pi(6, 7)$ that ends at $\lambda z_2@\tau_1$. The next tile is a constant tile. Tile τ_5 is z_2 and the pair $\pi(11), \pi(3)$ vary at $\pi(11), \pi(3)$ with τ_1 and $\theta \sim_{\tau_1} \theta'$ for $\theta \in \pi(12)$ and $\theta' \in \pi(4)$; so the interval at stage 5 is $\pi(10, 15)$ that ends at λ of τ_3 as there is no earlier position where control is at this leaf. The tile $\tau_6 = x_1(\lambda)$ and $\pi(16, 17)$ is the interval at this stage. Play then proceeds through a constant tile at stage 7. Tile $\tau_8 = x_2$ and the interval is $\pi(20, 33)$; positions $\pi(21), \pi(5)$ vary at $\pi(21), \pi(5)$ with τ_2 and $\theta \sim_{\tau_2} \theta'$ for $\theta \in \pi(22)$ and $\theta' \in \pi(6)$; the intervals $\pi(6, 15)$ and $\pi(22, 31)$ correspond as described in Remark 8.11. Tile τ_9 is $(10)z_1(\lambda)$; positions $\pi(35), \pi(7)$ vary at $\pi(35), \pi(7)$ with τ_1 and $\theta \sim_{\tau_1} \theta'$ when $\theta \in \pi(36)$ and $\theta' \in \pi(8)$; the intervals $\pi(8, 10)$ and $\pi(36, 38)$ correspond. With the next position $\lambda z@\tau_1 \in \pi(39)$ and $\pi(39), \pi(11)$ vary at $\pi(39), \pi(11)$ with τ_1 and $\theta \sim_{\tau_1} \theta'$ for $\theta \in \pi(40)$ and $\theta' \in \pi(12)$. The intervals and

$$\begin{aligned}
&\tau_1 = z(\lambda z_1, \lambda z_2) &&\pi(2,3) \\
&\tau_2 = z(\lambda x_1, \lambda x_2) &&\pi(4,5) &&\tau_2 \xleftarrow{\pi} \lambda z_1 @ \tau_1 \\
&\tau_3 = (6)z_1(\lambda) &&\pi(6,7) &&\tau_3 \xleftarrow{\pi} \lambda x_1 @ \tau_2 \\
&\tau_4 = h(\lambda) &&\pi(8,9) &&\tau_4 \xleftarrow{\pi} \lambda z_2 @ \tau_1 \\
&\tau_5 = z_2 &&\pi(10,15) &&\tau_5 \xleftarrow{\pi} \lambda @ \tau_4 \\
&\tau_6 = x_1(\lambda) &&\pi(16,17) &&\tau_6 \xleftarrow{\pi} \lambda @ \tau_3 \\
&\tau_7 = g(\lambda) &&\pi(18,19) &&\tau_7 \xleftarrow{\pi} \lambda x_2 @ \tau_2 \\
&\tau_8 = x_2 &&\pi(20,33) &&\tau_8 \xleftarrow{\pi} \lambda @ \tau_7 \\
&\tau_9 = (10)z_1(\lambda) &&\pi(34,43) &&\tau_9 \xleftarrow{\pi} \lambda @ \tau_6 \\
&\tau_{10} = a &&\pi(44,44) &&\tau_{10} \xleftarrow{\pi} \lambda @ \tau_9
\end{aligned}$$

Figure 12: Partition of π in Figure 7 of Example 8.17

$\pi(40,41)$, $\pi(12,13)$ correspond. Therefore at stage 9, the interval is built from two separate subintervals. Finally, stage 10 is the constant tile a. □

9. Unfolding and the small model property

We now prove decidability of higher-order matching at all orders, by showing the small model property; if $t \models P$ then there is a small term $t' \models P$. The proof starts with the tree of tiles that captures the p-partitions of all plays in a game $\mathsf{G}(t, P)$ and then extends it to a tree of basic tiles. We then define unfolding on such trees which underpins the small model property.

As with the 3rd-order case in Section 7, we examine the p-partitions of all plays in $\mathsf{G}(t, P)$. We maintain abuse of notation: if π and π' are two plays we let $\pi(i_k, j_k)$, $\pi'(i_k, j_k)$ be their intervals at stage k irrespective of their ranges. Instead of a sequence of simple tiles there is a tree of simple tiles that is associated with the p-partitions as each p-partition shares the initial tile τ_1 of t. As a representation for this tree of tiles, we let its root be τ_1^Π when Π is the set of all plays in $\mathsf{G}(t, P)$; any other node of this tree has the form $\tau_k^{\Pi'}$ which represents that for each $\pi \in \Pi'$, τ_k^π is its tile at stage k and for all earlier stages $m < k$, every play in Π' also shares the same tile at stage m. Thus, the tree has the form depicted in Figure 13. However, we also assume the induced edge relations $\xleftarrow{\pi}$ within this tree: $\tau_m^{\Pi'} \xleftarrow{\pi} t'@\tau_k^{\Pi''}$ if $\tau_m^\pi \xleftarrow{\pi} t'@\tau_k^\pi$ and $\pi \in \Pi' \cap \Pi''$.

Let T be the tree of tiles for the p-partitions of all plays in $\mathsf{G}(t, P)$. We drop the superscript Π' from tiles $\tau_k^{\Pi'}$ whenever the context allows. We assume the definition of π-path, Definition 8.15, which picks out the sequence of tiles in t from its root to $\tau_k^{\Pi'}$ when $\pi \in \Pi'$. Also, we shall assume that Definitions 6.3, 6.4 and 6.7 of j-below, below, immediate j-dependent, dependent and embedded apply to tiles in a tree T by examining π-paths and bindings: for instance, τ_m is an immediate j-dependent of τ_k if τ_k is in the π-path for τ_m, the free variable y at the head of τ_m is bound in τ_k and τ_m is j-below τ_k relative to this π-path. However, because of the linear representation of a p-partition there is also the idea that a tile τ_m is *later* than τ_k in T if there is a play π and $\tau_k = \tau_k^\pi$ and $\tau_m = \tau_m^\pi$ and $m > k$; we also say that τ_k is *earlier* than τ_m.

As with the 3rd-order case in Section 7, we identify *special* tiles in the tree.

$$\tau_n^{\Pi_1'}$$
$$\tau_k^{\Pi'} \cdots\cdots\cdots \tau_{n'}^{\Pi_l'}$$
$$\tau_1^{\Pi} \cdots\cdots\cdots \tau_j^{\Pi} \cdots\cdots\cdots \cdots$$
$$\tau_k^{\Pi''} \cdots\cdots\cdots \tau_{m'}^{\Pi_l''}$$
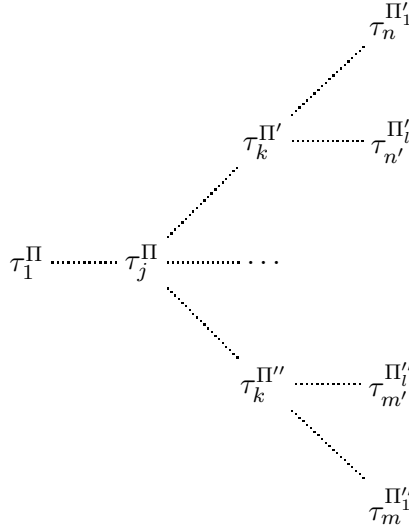$$\tau_m^{\Pi_1''}$$

<div align="center">Figure 13: Tree of tiles in all p-partitions</div>

Definition 9.1. The tile $\tau_k^{\Pi} \in$ T is *special* if it obeys one of the following three conditions;
(1) for some $\pi \in \Pi$, $\pi(i_k, j_k)$ is nri,
(2) for some $\pi \in \Pi$, $q \in \pi(j_k)$ is a final state,
(3) for some π, $\pi' \in \Pi$, $t' \neq t''$ when $t' \in \pi(j_k)$ and $t'' \in \pi'(j_k)$.
A tile is *x-special* if it obeys (1) or (2) of these conditions.

A special tile that is not x-special is a play separator. There is the same upper bounds on the number of special tiles in T as in the 3rd-order case.

Fact 9.2. Assume T is tree of tiles associated with $\mathsf{G}(t, P)$. Within T there are
(1) at most δ (= the right size for P, Definition 3.8) tiles that involve nri intervals;
(2) at most p (= the number of plays in $\mathsf{G}(t, P)$) tiles where play ends;
(3) therefore, at most $\delta + p$ are x-special tiles;
(4) at most $p - 1$ tiles that are play separators.

The proof of the small model property for the 3rd-order case is straightforward: use transformation **T2** of Section 7 to remove any tile that is not special from T and update edges. With higher orders we cannot just omit a tile that is not special. It may have dependents, so its removal would result in a tree that is no longer a closed term. Or an associated interval may finish at an atomic leaf of some other tile in the tree, so its removal may not preserve game playing. Instead, we introduce tile *unfolding* as a transformation on a tree T. We need to generalise the notion of tree to that of a *tree of basic* tiles. We will be interested in a tile $\tau_m \in$ T that is an immediate dependent of a top or embedded tile τ_k which is not x-special and which also does not have x-special later tiles belonging to the same family. Therefore, as we shall see, τ_m can be replaced in the tree by a basic tile that is constructed from τ_k and τ_m; this may require revision of later edges in the tree. Before developing the full account, we shall now briefly illustrate it.

Example 9.3. Consider the p-partition in Figure 10 of Example 8.14 for the play π' in Figure 8 on the term tree in Figure 1. The tree of tiles with the edge relation $\xleftarrow{\pi'}$, with π'
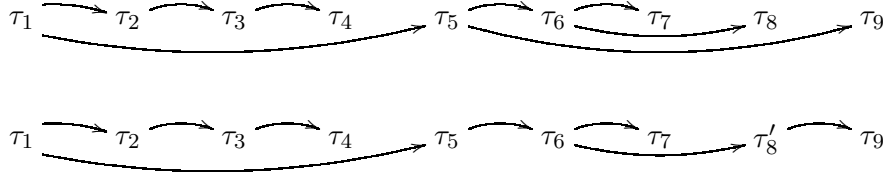
Figure 14: Tree of tiles before and after unfolding in Example 8.14

omitted, is pictured in the top diagram in Figure 14. There are edges, for instance, from different atomic leaves of τ_1 to τ_2 and τ_5. The special (and x-special) tiles are τ_2, τ_4 and τ_9; intervals on the first two of these are nri and the third is where the play π' finishes. Transformation **T2** of Section 7 would allow us to remove the tile τ_3 with the effect that the edge from τ_2 would then be to τ_4: dynamically, in terms of play, this means that the ri interval $\pi'(i_3, j_3)$, a play on τ_3, is omitted and the interval $\pi'(i_4, j_4)$ reduced (by omitting its ri play on τ_3). Tile τ_5 is a top tile with immediate dependent τ_8; neither of these tiles is x-special. However, because of this binding and the fact that it has two outgoing edges from different atomic leaves, τ_5 cannot be omitted (like τ_3); nor can we remove τ_8 because $\pi'(i_8, j_8)$ ends at an atomic leaf of τ_5. Instead we can unfold τ_5 at τ_8: we introduce the basic tile $\tau_8' = z(\lambda y.\tau_8, \lambda)$ which prefaces τ_5 to τ_8; the new interval $\pi'(i_8, j_8)$, in effect, includes an extra interval that corresponds to $\pi'(i_5, j_5)$ as a prefix and then omits the ri plays on τ_6 and τ_7 from the old interval $\pi'(i_8, j_8)$. Play in the new interval $\pi'(i_8, j_8)$ which is still ri now finishes at an atomic leaf of τ_8'; so edges may need to be changed; here, τ_8' now has an edge to τ_9. The effect of this unfold is pictured in the lower diagram of Figure 14. As a consequence, both τ_5 and τ_8' can now be removed by transformation **T2**. As the reader can verify, tile τ_6 can also be unfolded at τ_7. □

We call the process "unfolding" τ at τ' (where τ' is an immediate dependent of τ) because it is analogous to unfolding or unravelling a transition system in modal logic; here, there is the extra dimension of binding. As with unravelling, the purpose of unfolding in T is also to approximate the tree model property. Technically, from a game-theoretic point of view, what will justify unfolding is permuting, repeating and omitting corresponding ri intervals: for instance, with the replacement of τ_8 by τ_8' in Example 9.3 there is a repetition and a permutation of intervals that correspond to the earlier ri interval on τ_5 within τ_8', and then omission of the ri intervals that correspond to $\pi'(i_6, j_6)$, $\pi'(i_7, j_7)$ within $\pi'(i_8, j_8)$. Although initially unfolding increases the size of a tree, its point is to reduce tile levels. Tile τ_8 is level 2 whereas its replacement τ_8' is a level 1 (basic end) tile. Unfolding is not defined as a transformation in the sense of Section 7 because it is not local; edges to later tiles may be revised in its application.

First, we extend the notion of a tree to that of a tree of basic tiles with associated plays that are p-partitioned.

Definition 9.4. The tree of basic tiles T has associated plays Π if Π is the set of plays down the branches of T and for each $\pi \in \Pi$, $\pi = \pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ for some n such that

(1) if $m \leq n$ then $\pi(i_m, j_m)$ is an interval on the tiles $\tau_1^\pi, \ldots, \tau_m^\pi$ in T that starts at the root of τ_m,

(2) if $m < n$ and $t'@\tau_k^\pi \in \pi(j_m)$ then t' is an atomic leaf of τ_k^π and $\tau_{m+1}^\pi \xleftarrow{\pi} t'@\tau_k^\pi$,

(3) if $m < n$ and τ_m^π is a top or constant tile then $\tau_{m+1}^\pi \xleftarrow{\pi} t'@\tau_m^\pi$ for some t',

(4) if $\pi(i_m, j_m)$ is ri and $\tau_{m+1}^\pi \xleftarrow{\pi} t'@\tau_k^\pi$ then τ_k^π belongs to the same family as τ_m^π,

(5) if $\pi(i_m, j_m)$ is ri and τ_m^π is an embedded tile in t then $\tau_{m+1}^\pi \xleftarrow{\pi} t'@\tau_m^\pi$ for some t',

(6) if $\pi(i_m, j_m)$ is ri and $\tau_{m+1} \xleftarrow{\pi} t'@\tau_k$ then $t'@\tau_k$ is not in the π-path of τ_m.

Initially, when T is the tree of simple tiles constructed from the p-partitions of the plays in $\mathsf{G}(t, P)$ then T has associated plays $\mathsf{G}(t, P)$; parts (1) and (2) of Definition 9.4 follow from Definition 8.13 of p-partition, the remainder from Proposition 8.16. We assume that the definitions of b-partition, dependent, special, x-special, π-path and so on are extended to basic tiles in a tree of basic tiles.

Definition 9.5. Assume T is a tree of basic tiles with associated plays. Tile τ_k *is unfoldable at* τ_m if

(1) τ_k is a top or embedded tile,

(2) τ_m is the first tile in $\tau_{k+1}, \ldots, \tau_m$ that is a dependent of τ_k,

(3) τ_k and no later tile that is in the same family as τ_k is x-special.

If τ_k is unfoldable at τ_m then we define the unfolding of τ_k at τ_m in T as the tree T′ with the same nodes as T except that τ_m is replaced by a basic tile τ_m' that is a composition of τ_k and τ_m. (Edges in T may also be changed in T′.) In the following we define the associated plays on T′ from those on T: the definition uses the notion of corresponding positions as defined (for intervals) in Definition 8.3. However, because T′ is different from T, the notion of correspondence is slightly weakened in specific circumstances to allow that a sequence of positions may correspond to a single position.

Definition 9.6. Assume $\tau_k = y(\ldots \lambda \overline{x} \ldots)$ is unfoldable at $\tau_m = \tau_m^{\Pi'}$ in T and for each $\pi \in \Pi'$, $\tau_{k+1} \xleftarrow{\pi} \lambda \overline{x} @ \tau_k$. The *unfolding of* τ_k *at* τ_m *in* T is the tree T′ where T′ has the same tiles as T except that τ_m is replaced by $\tau_m' = y(\ldots \lambda \overline{x}.\tau_m \ldots)$. For each play $\pi = \pi(j_0), \pi(i_1, j_1), \ldots, \pi(i_n, j_n)$ on T there is a play $\sigma = \sigma(j_0), \sigma(i_1, j_1), \ldots, \sigma(i_n, j_n)$ on T′, defined as follows in stages and top down:

(1) if $\pi \notin \Pi'$ or $\pi \in \Pi'$ and $l < m$ then $\sigma(i_l, j_l) = \pi(i_l, j_l)$ and $\tau_{l+1} \xleftarrow{\sigma} t'@\tau_{k'}$ iff $\tau_{l+1} \xleftarrow{\pi} t'@\tau_{k'}$;

(2) if $\pi \in \Pi'$, $l \geq m$, $t'@\tau_j \in \pi(j_{m-1})$ and $t'@\tau_j \in \sigma(j_{m-1})$ then $\sigma(i_l, j_l)$ is the continuation from the head of τ_l in $\gamma_l = \tau_1, \ldots, \tau_l$ with edges $\xleftarrow{\sigma}$. Any position $\sigma(i')$ in $\sigma(i_l, j_l)$ corresponds to $\pi(i')$ in $\pi(i_l, j_l)$ in the sense of Definition 8.3 except in the following circumstances where the notion of correspondence is weakened:

   - *the positions are at* τ_m' *and* τ_m: if $t'@\tau_j \in \sigma(i'-1)$ and $t'@\tau_j \in \pi(i'-1)$ then $\sigma'\sigma(i')$ corresponds to $\pi(i')$ where σ' is a shortest play on τ_k in τ_m' that ends at $\lambda \overline{x} @ \tau_m'$,
   - *there is a jump into* τ_m' *and* τ_k: as a result of move C4 of Figure 3, $t''@\tau_k \in \pi(i')$ and $t''@\tau_m' \in \sigma(i')$; then $\sigma(i', i'+i'')$ corresponds to $\pi(i', i'+i'')$ if these intervals are internal to τ_m' and τ_k,
   - *the positions are at* $\lambda \overline{x} @ \tau_m'$ *and* $\lambda \overline{x} @ \tau_k$: $\lambda \overline{x} @ \tau_m' \in \sigma(i')$, $\lambda \overline{x} @ \tau_k \in \pi(i')$ and $\pi(i'')$ is the first later position such that $t'@\tau_j \in \pi(i'')$ and $\pi(l, i')$ is the play on τ_k in the b-partition for $\pi(i'')$; then $\sigma(i')$ corresponds to $\pi(i', i'')$.

   For the edges: if τ_l is τ_m or is below τ_m and $\tau_{l+1} \xleftarrow{\pi} t''@\tau_k$ then $\tau_{l+1} \xleftarrow{\sigma} t''@\tau_m'$; otherwise, $\tau_{l+1} \xleftarrow{\sigma} t''@\tau_p$ if $\tau_{l+1} \xleftarrow{\pi} t''@\tau_p$.

We say that σ on T′ is *the companion* of π on T and that T′ is *an unfolding* of T.

In Example 9.3, τ_5 is unfolded at τ_8; T is the upper and T$'$ the lower tree in Figure 14. The companion play σ' on T$'$ of π' on T has the same intervals up to and including stage 7. In defining the weakened correspondence, play in σ' is at τ_8' in T$'$ and in π' at τ_8 in T: so $\sigma''\sigma'(i_8)$ where σ'' is the initial play on τ_5 in τ_8' now corresponds to $\pi'(i_8)$; next there is a jump into τ_8' and τ_5 by move C4 which is to $\lambda y@\tau_8'$ and $\lambda y@\tau_5$ and so this σ' position corresponds to the interval that is from $\lambda y@\tau_5$ to the lambda node directly above τ_8; so then the next positions will again correspond. In both these cases, where an interval corresponds to a position, the interval must be ri.

Proposition 9.7. *Assume* T *has associated plays* Π. *If* T$'$ *is an unfolding of* T *then* T$'$ *has associated plays* $\Sigma = \{\sigma \,|\, for\ some\ \pi \in \Pi, \sigma\ is\ a\ companion\ of\ \pi\}$.

Proof. Assume that T has associated plays Π and T$'$ is the unfolding of τ_k at $\tau_m^{\Pi'}$ in T. By definition, if $\pi \notin \Pi'$ and σ is its companion then $\sigma = \pi$ is a play on a branch of T$'$ as required. Otherwise, assume $\tau_k = y(\ldots\lambda\overline{x}\ldots)$, $\tau_{k+1} \xleftarrow{\pi} \lambda\overline{x}@\tau_k$ for each $\pi \in \Pi'$, $t'@\tau_j \in \pi(j_{m-1})$ and $\tau_m' = y(\ldots\lambda\overline{x}.\tau_m\ldots)$. We now show that we can find corresponding positions $\sigma(i')$ in $\sigma(i_l, j_l)$ and $\pi(i')$ in $\pi(i_l, j_l)$ as described in Definition 9.6 for each $l \geq 1$. For $l < m$, this holds because $\sigma(i_l, j_l) = \pi(i_l, j_l)$ and $\tau_{l+1} \xleftarrow{\sigma} t''@\tau_{k'}$ iff $\tau_{l+1} \xleftarrow{\pi} t''@\tau_{k'}$. Consider next the case that corresponding positions are at τ_m' and τ_m: $t'@\tau_j \in \sigma(i'-1)$ and $t'@\tau_j \in \pi(i'-1)$. We examine the b-partitions for $\sigma(i'-1)$ and $\pi(i'-1)$ and their component plays σ', π' on τ_k; we show that σ' and π' correspond, that they are ri and that they are shortest plays on τ_k (that end at the leaf $\lambda\overline{x}@\tau_k$). We prove this by induction on corresponding positions $\sigma(i'-1)$, $\pi(i'-1)$, $i' \geq i_m$. The base case is when $i' = i_m$. At this point the b-partitions for $\sigma(i'-1)$ and $\pi(i'-1)$ are the same. Let σ' be the component interval in this b-partition on τ_k. That σ' is ri and is a shortest play on τ_k that ends at $\lambda\overline{x}$ follows from the fact that τ_m is the first tile that is a dependent of τ_k in the sequence $\tau_{k+1}, \ldots, \tau_m$, that τ_k is a top or embedded tile and that τ_k and all later tiles in the same family are not x-special; consequently, any tile between τ_k and τ_m in the same family as τ_k has an associated ri interval and, therefore, if play in such an interval were at a different atomic leaf of τ_k than $\lambda\overline{x}$, then τ_m would not be a dependent of τ_k. For the inductive step the argument is similar after noting the following property: if corresponding positions are at τ_m' and τ_m then it is not possible that play was in a dependent tile $\tau_{m'}$ of τ_k that is below τ_m before jumping back into τ_k, then to $\lambda\overline{x}@\tau_k$ and then proceeding to τ_m because in σ the simulating position would be at τ_m in τ_m' (because in T$'$, $\tau_{m'}$ is a dependent of τ_m' and edges are updated in T$'$). So this property holds. Therefore, returning to the main argument, the continuation from $\sigma(i'-1)$ consists first of a sequence of moves σ'' that corresponds to σ' except it is on τ_k within τ_m'; therefore, $\sigma''\sigma(i')$ weakly corresponds to $\pi(i')$. Consider next corresponding positions such that at the next positions they jump into τ_m' and τ_k; $t''@\tau_k \in \pi(i')$ and $t''@\tau_m' \in \sigma(i')$. Then the intervals $\pi(i', i'+i'')$, $\sigma(i', i'+i'')$ that are internal to these tiles correspond except that they take place in τ_k and τ_m'. The final circumstance to examine is that corresponding positions are $\sigma(i')$, $\pi(i')$ such that $\lambda\overline{x}@\tau_m' \in \sigma(i')$ and $\lambda\overline{x}@\tau_k \in \pi(i')$. We show that there is a first position such that $t'@\tau_j \in \pi(i'')$ and for some l, $\pi(l, i')$ is the play on τ_k in the b-partition for $\pi(i'')$ and the b-partition for $\sigma(i')$ contains the ri interval $\sigma(i'', j'')$ which corresponds to $\pi(i', i'')$. Positions $\sigma(i')$ and $\pi(i')$ must be the result of a C4 move. However, the look-up table where this entry is defined must be at a position within a tile $\tau_{m'}$ that is a dependent of τ_k in T and of τ_m' in T$'$ that is, below τ_m. Now the result follows as τ_k and all later tiles in the same family are not x-special. Clearly, T$'$ has no other

plays than the companions of Π. Moreover, each companion play σ obeys the six conditions in Definition 9.4 given that they hold for each π on T. $\qquad\qquad\square$

To prove the small model property, assume a smallest term t such that $t \models P$ and let T be its tree of simple tiles that captures the p-partitions of every $\pi \in \mathsf{G}(t, P)$. First, we describe the proof for a particular case of T, a *general atoms case*, that obeys the following condition: if τ_k is not a constant tile then it is not x-special. What this means is that every top and embedded tile is unfoldable at a first dependent[3]. The decidability proof now reduces to the 3rd-order case as the tree-model property is regained.

Initially, restricting further, assume P is 5th-order: there are, therefore, only two levels of non-constant tiles, top and end tiles. Starting top down with $\mathrm{T}_0 = \mathrm{T}$, at each stage T_i, a top tile τ that is closest to the root and that has dependents is unfolded at a first dependent τ'. The construction finishes at some stage n, when T_n has no unfoldable tiles; that is, when it has no top tiles with dependents. This means that T_n only consists of constant tiles and basic top tiles that are also end tiles. Once the tree is in this form, only the special tiles (those that are constant tiles and play separators) need to be kept: the remainder are redundant using transformation **T2**. To obtain a small term, the initial $\lambda\overline{y}$ of t is placed at its top and the constant $d : \mathbf{0}$ is placed below every leaf $\lambda\overline{z}$. The bound on the size of t is larger than in Fact 7.10 because the units are now *basic* tiles instead of simple tiles. In the worst case, each whole tile consists of one simple top tile and at most α simple end tiles in any branch (where α is the arity of P). Therefore, using this construction, we obtain the following bound where the measures are all from P: $|t| \le (\alpha + 1) \times (\delta + (2p - 1))$.

For higher-orders, the bounds become even larger. Initially, all top and embedded tiles with dependents are unfoldable. Now unfolding is iterated. At each stage T_i, a top or embedded tile τ with dependents which has the *greatest* level is unfolded; if there is more than one such tile then one that is closest to the root is chosen to be τ and it is unfolded at a first dependent to give T_{i+1}. Consider what may happen when P is 7th-order. There are now three levels of non-constant tiles: top, middle and end. Unfolding reconciles embedded middle tiles with their immediate end dependents, which may in turn create larger embedded middle tiles or end tiles. For instance, assume the following branch of tiles in T_0

$$\tau \quad \tau_1 \quad \tau_2 \quad \tau_{21} \quad \tau_3 \quad \tau_{31}$$

where τ is a top tile, τ_1, τ_2, τ_3 are dependents of τ, τ_2 and τ_3 are both embedded middle tiles because of τ_1 and τ_{21} is a dependent of τ_2 and τ_{31} of τ_3. First, τ_2 is unfolded at τ_{21} and then τ_3 is unfolded at τ_{31} to give the following sequence.

$$\tau \quad \tau_1 \quad \tau_2 \quad \tau_2\tau_{21} \quad \tau_3 \quad \tau_3\tau_{31}$$

The situation has reduced to the 5th-order case, as there are now only two levels of tiles. So, the complete unfolding is the following sequence of top tiles that are also end tiles.

$$\tau \quad \tau\tau_1 \quad \tau\tau_1\tau_2 \quad \tau\tau_1\tau_2(\tau_2\tau_{21}) \quad \tau\tau_1\tau_2(\tau_2\tau_{21})\tau_3 \quad \tau\tau_1\tau_2(\tau_2\tau_{21})\tau_3(\tau_3\tau_{31})$$

Therefore, using **T2** we can remove any non-special tiles from the unfolded tree and any embedded (basic) end tiles within special tiles. The following is a very crude size bound on a smallest term that solves the problem P of order $2n + 1$ in this restricted case: $|t| \le g(n) \times (\delta + (2p - 1))$ where $g(1) = 1$ and $g(k + 1) = (\alpha + 1)^{g(k)}$.

---

[3]This feature, that every top and embedded tile is unfoldable at a first dependent, is true in the atoms case where $\delta = 0$ even though some of these tiles may be x-special.

Remark 9.8. In the general atoms case there is a bounded size solution term with a simple form that is a *transferring* term [12]. For instance, consider the definition of transferring in [9]; for every subterm of $\lambda x_1 \ldots x_n.t$ of the form $x_i s_1 \ldots s_k$ the free variables of any s_j belong to $\{x_1, \ldots, x_n\}$. □

Let us return to the unrestricted case. We show that there is a bounded size term that *almost* has the tree model property. Not all top or embedded tiles with dependents can be unfolded because of their contribution to solving P; for instance, in Example 9.3, tile τ_1 cannot be unfolded at τ_4. There is also a further issue that does not occur with the general atoms case. After unfolding, we need to extract a term from the unfolded tree. The intention is that the edge relation $\xleftarrow{\pi}$ should be the subtree relation. However, there can be multiple edges of the form $\tau_m^\pi \xleftarrow{\pi} t'@\tau_k$ and $\tau_n^{\pi'} \xleftarrow{\pi'} t'@\tau_k$: so, we need to guarantee that the "subterms" rooted at τ_m^π and $\tau_n^{\pi'}$ are compatible.

Definition 9.9. The tree T has the *subterm property* if there is a smallest equivalence relation $\cong$ on its tiles such that whenever $\tau_m \cong \tau_n$,

(1) they are syntactically the same tile,

(2) if $\tau_k \xleftarrow{\pi} t''@\tau_m$ and $\tau_l \xleftarrow{\pi'} t''@\tau_n$ then $\tau_k \cong \tau_l$.

If T has the subterm property then its *extraction* is defined top-down. For any edge $\tau_m \xleftarrow{\pi} t'@\tau_k$ the tile τ_m is moved to be directly below $t'@\tau_k$. The initial $\lambda\overline{y}$ of t is placed at the top and the constant $d : \mathbf{0}$ is placed below every leaf $\lambda\overline{z}$. Next we compute the smallest equivalence relation $\cong$ starting with the identity, for each tile $\tau_m \cong \tau_m$ and then closing up inductively under the conditions in Definition 9.9: tiles $\tau_m \cong \tau_n$ are identified.

Fact 9.10. If T has has the subterm property then its extraction is a term.

To obtain the small model property, we show that given the initial tree there is a way of unfolding such that the resulting tree has the subterm property. We then examine the extraction and apply the transformation **T2** to remove any redundant tiles. The upper bound is very crude.

Theorem 9.11. *If t is a smallest solution of P of order at most $2n + 1$, then $|t| \leq g(n) \times ((p^2 \times \delta \times N(n)) + p - 1)$ where α is the arity of P, $N(n) = p \times \sum\{\alpha^i : 1 \leq i \leq g(n)\}$, $g(1) = 1$ and $g(k + 1) = (\alpha + 1)^{g(k)}$.*

Proof. Assume t is a smallest solution to P of order $2n + 1$. Let T_0 be the tree of simple tiles for the p-partitions of each play in $\mathsf{G}(t, P)$. By Fact 9.2 there are at most $\delta + (2p - 1)$ special tiles in the tree. The issue is to define a largest sequence of unfolds such that afterwards the resulting tree has the subterm property. If every top and embedded tile is unfoldable then at each stage $i + 1$ those top or embedded tiles with the largest level that have dependents are chosen; one of them that is closest to the root is then unfolded at a first dependent; so T_{i+1} is an unfolding of T_i. Unfolding is continued until T_n for some n when there are no more unfoldable tiles; an easy argument shows that there is such an n as the unfolding reduces tile level. By Proposition 9.7, it follows that companion plays preserve final states. After unfolding, every tile in the resulting tree is a basic top tile or a constant tile and, consequently, obeys the subterm property. So, its extraction t' is a term. Except for the special tiles in t', the remainder are redundant using transformation **T2**. Therefore, this produces a term t'' where $|t''| \leq g(n) \times (\delta + (2p - 1))$. Because t is a smallest term that solves P, t obeys this bound. Otherwise, some top or embedded tiles

must be excluded from being unfolded. In the worst case there are at most $\delta + p$ x-special tiles in T_0 and, therefore, at most $(\delta + p) \times N(n)$ distinct tiles in T_0 that are x-special or dependents of x-special tiles and that are also not embedded tiles. Moreover, there are at most a further $p - 1$ tiles that are special. We now proceed with a sequence of unfoldings as follows: find a highest level unfoldable tile that is closest to the root and unfold it at a first dependent; and keep repeating this until there are no more unfoldable tiles. If the resulting tree T_n fails to have the subterm property then at some earlier stage j there are tiles τ_k and τ_l in T_j that are syntactically the same and one of them, say τ_k is later unfolded whereas τ_l is not and both have edges to the same atomic leaf: a failure of condition (2) of Definition 9.9. Both these tiles are dependents of the same tile in T_j. Therefore, we exclude the tile that τ_k is a dependent of from being unfolded and then examine a complete sequence of unfoldings without it. By repeating this argument, after the sequence of unfoldings the resulting tree has the subterm property; the number of distinct tiles that are excluded from being unfolded is at most $(\delta + p) \times N(n)$. By Proposition 9.7 its extraction t' solves P. To begin with there are $((\delta + p) \times N(n)) + p - 1$ tiles that are either special or dependents of x-special tiles. Again, we wish to apply the transformation **T2** to remove redundant end tiles from t'. The question is how many extra unfolded tiles are also special because they have become separators. This depends on the number of plays p. If $p = 1$ then no extra tiles are needed. If $p > 1$ then for each of the $((\delta + p) \times N(n))$ tiles, there could be at most $p - 1$ new play separators in the full unfolding. Therefore, the result follows as each tile has bounded length $g(n)$.   □

10. Conclusion

Although we have shown that higher-order matching is decidable, the upper size bound is very coarse and more work needs to be done to make it more accurate. Although our complexity analysis is in terms of term size, it crudely agrees with the known non-elementary complexity lower bound based on [17]. As order of a problem increases, so does its level and, therefore, the size of a smallest solution term according to our analysis increases (exponentially). Implicit in the analysis are positive sensible algorithms for solving dual interpolation problems. The game-theoretic characterisation of dual interpolation allows us to examine incomplete terms. As a first step, given a problem P its component set of simple tiles can be defined from the subtypes and the constants (including forbidden constants and the new constant d of Section 4). From these, we can then define varieties of basic tiles that have no embedded end tiles. We can then proceed to construct a term tree in stages, first by seeing if there are basic top tiles that separate plays, and, thereby, continue recursively. Otherwise, we need to check if there must be non-top basic tiles, and so on. In the worse case, we need to examine all possible terms whose size is bounded by Theorem 9.11. It may be worth investing effort to implement a tool that builds such terms.

An open question is whether the set of all solutions of an interpolation problem is independently characterisable. For instance, Comon and Jurski define tree automata that characterise all solutions to a 4th-order problem [2]. In Section 8 of their paper they describe two problems with extending their automata beyond order 4. The first is that states of an automaton are constructed out of the observational equivalence classes of terms. Up to a 4th-order problem, one only needs to consider finitely many terms. With 5th and higher orders, this is no longer true and one needs to quotient the potentially infinite terms into their respective observational equivalence classes in order to define only finitely many states:

however as Padovani shows this procedure is, in fact, equivalent to the matching problem itself [13]. The second problem is the term trees that the automata recognise. For a 4th-order problem, there is an automaton that recognises its full set of solutions (up to α-equivalence) even though the syntax may be infinite. Comon and Jurski define a special kind of automata, $\square$-automata, to achieve this [2]. The occurrence of a leaf $\square$ in a term tree represents any (syntactically correct) subtree. A $\square$ cannot contribute to the solution of a matching problem. This is no longer true at 5th-order, as illustrated by their example $x\,\lambda yz.y(\lambda z'.zz') = a$. Solutions of this problem include the terms

$$\lambda x_1.x_1(\lambda y_1.x_1(\ldots x_1(\lambda y_n.y_{i_1}(y_{i_2}(\ldots (y_{i_k}a)\ldots)))u_n\ldots)u_2)u_1$$

where for some $m \leq k$, every u_{i_j}, $j \leq m$, is the identity and $u_{i_{m+1}}$ is the constant function a. Because k is arbitrarily large, one cannot use a bounded number of variables to capture all these terms [2]. In the general case, by iterated unfolding, any solution term can be transformed into another solution term that only uses (and reuses) boundedly many variables. Whether this is capturable using some kind of automaton (such as a transducer) is open.

In subsequent work [21], we are able to overcome the first problem of Comon and Jurski at higher orders but not the second. We provide a tree-automata characterisation relative to a *finite* alphabet: given a problem P, a finite set of variables and constants the (possibly infinite) set of terms that are built from those components and that solve P is regular. The states of the automaton are built from abstractions of sequences of moves in a (variant version of the) game which works for all orders. Although there is active research extending automata on words and trees to infinite alphabets which preserve "good" properties, such as decidability of non-emptiness, see [16] for a recent survey, the results do not yet apply to the case caused by higher-order binding.

As we briefly mentioned in Section 4, Ong has shown that the tree-checking game can be presented using game-semantics and innocent strategies. Another question is whether this framework provides an alternative basis for understanding higher-order matching.

References

[1] Barendregt, H. Lambda calculi with types. In *Handbook of Logic in Computer Science, Vol 2*, ed. Abramsky, S., Gabbay, D. and Maibaum, T., Oxford University Press, 118-309, (1992).

[2] Comon, H. and Jurski, Y. Higher-order matching and tree automata. *Lecture Notes in Computer Science*, **1414**, 157-176, (1997).

[3] Dougherty, D. and Wierzbicki, T. A decidable variant of higher order matching. *Lecture Notes in Computer Science*, **2378**, 340-351, (2002).

[4] Dowek, G. Third-order matching is decidable. *Annals of Pure and Applied Logic*, **69**, 135-155, (1994).

[5] Dowek, G. Higher-order unification and matching. In *Handbook of Automated Reasoning, Vol 2*, ed. Robinson, A. and Voronkov A., North-Holland, 1009-1062, (2001).

[6] Goldfarb, W. The undecidability of the second-order unification problem. *Theoretical Computer Science*, **13**, 225-230, (1982).

[7] Huet, G. *Rèsolution d'èquations dans les langages d'ordre 1, 2, ... ω*. Thèse de doctorat d'ètat, Universitè Paris VII, (1976).

[8] Joly, T. On λ-definability I: the fixed model problem and generalizations of the matching problem. *Fundamenta Informaticae*, **65**, 135-151, (2005).

[9] Loader, R. Unary PCF is decidable. *Theoretical Computer Science*, **206**, 317-329, (1998).

[10] Loader, R. Higher-order β-matching is undecidable. *Logic Journal of the IGPL*, **11(1)**, 51-68, (2003).

[11] Ong, C.-H. L. On model-checking trees generated by higher-order recursion schemes. *Proceedings of 21st Annual IEEE Symposium on Logic in Computer Science*, (LICS), 81-90, (2006). (Longer version available from Ong's web page, 55 pages preprint.)

[12] Padovani, V. Decidability of all minimal models. *Lecture Notes in Computer Science*, **1158**, 201-215, (1996).

[13] Padovani, V. Decidability of fourth-order matching. *Mathematical Structures in Computer Science*, **10(3)**, 361-372, (2001).

[14] Schubert, A. Linear interpolation for the higher-order matching problem. *Lecture Notes in Computer Science*, **1214**, 441-452, (1997).

[15] Schmidt-Schauβ, M. Decidability of arity-bounded higher-order matching. *Lecture Notes in Artificial Intelligence*, **2741**, 488-502, (2003).

[16] Segoufin, L. Automata and logics for words and trees over an infinite alphabet. *Lecture Notes in Computer Science*, **4207**, 41-57, (2006).

[17] Statman, R. The typed λ-calculus is not elementary recursive. *Theoretical Computer Science*, **9**, 73-81, (1979).

[18] Statman, R. Completeness, invariance and λ-definability. *The Journal of Symbolic Logic*, **47**, 17-26, (1982).

[19] Stirling, C. *Modal and Temporal Properties of Processes*. Texts in Computer Science, Springer, (2001).

[20] Stirling, C. Higher-order matching and games. *Lecture Notes in Computer Science*, **3634**, 119-134, (2005).

[21] Stirling, C. Higher-order matching, games and automata. *Proceedings of 22nd Annual IEEE Symposium on Logic in Computer Science*, (LICS 2007), 326-335, (2007).

[22] Støvring, K. Higher-Order beta matching with solutions in long beta-eta normal form. *Nordic Journal of Computing*, **13**, 117-126, (2006).

[23] Wierzbicki, T. Complexity of higher-order matching. *Lecture Notes in Computer Science*, **1632**, 82-96, (1999).