Cut Introduction

Cut-Formulas with Multiple Universal Quantifiers

Janos Tapolczai

December 11, 2013

1 Introduction

In [1, Ch. 5], algorithmic cut-introduction is described, albeit restricted to cut-formulas with one universal quantifier. This document extends the mechanism described therein to cut-formulas with an arbitrary number of universal quantifiers.

Specifically, we replace the Δ operator with Δ_G. Both descend through a list of terms in parallel in search for a common teram structure, but whereas Δ is limited to introducing only one variable, Δ_G can introduce an arbitrary number. Also, a specific definition of completeness will be given, namely that, for every set of terms and an unbounded number of variables, every decomposition has a uniqe normal form, which Δ_G computes. Later on, we also give an operator Δ_G^k for a bounded number of variables, and the corresponding completeness result that every decomposition computed by Δ_G^k is in *weak k-normal form*.

2 Generalized Δ-Vector

One need only change the definition of the δ-vector and extend it to deal with vectors of variables instead of a single variable.

First, we define the helper-function `nub`[1] which eliminates duplicates:

$$\texttt{nub}(f(u_1,\ldots,u_m),(\overline{s_1},\ldots,\overline{s_n})) = \texttt{elim}\uparrow\uparrow\infty = (f(u_1',\ldots,u_m'),(\overline{s_1},\ldots,\overline{s_p}))$$

where α_{F+1} is the α with the lowest index which occurs in $u_1,\ldots,u_m$ and

$\texttt{elim}(f(u_1,\ldots,u_m),(\overline{s_1},\ldots,\overline{s_n}) =$
if $[\exists i,j : i<j]\ \overline{s_i} = \overline{s_j}$ then
remove $\overline{s_j}$,
replace α_{j+F} with α_{i+F} and
replace α_k with $\alpha_{k-1}\ \forall k > j+F$ in $u_1,\ldots,u_m$.

We then define the generalized delta-vector Δ_G:

[1] `nub` is the common name for the duplicate elimination function in functional languages.

$$\Delta_G(t_1,\dots,t_n) = \begin{cases} (t_1,()) \text{ if } t_1 = t_2 = \cdots = t_n \text{ and } n > 0 \\ \\ \texttt{nub}(f(u_1,\dots,u_m),(\overline{s_1},\dots,\overline{s_q})) \\ \quad \text{if all} \quad t_i = f(t_1^i,\dots,t_m^i) \text{ and case 1 does not apply,} \\ \quad \text{where} \quad (\overline{s_1},\dots,\overline{s_q}) = \bigsqcup_{1\leq j\leq m} \pi_2(\Delta_G(t_j^1,\dots,t_j^n)) \text{ and} \\ \qquad\qquad\quad u_j = \pi_1(\Delta_G(t_j^1,\dots,t_j^n)) \text{ for all } j \in \{1,\dots,m\} \\ \\ (\alpha_{\text{UNIQUE}},(t_1,\dots,t_n)) \quad \text{otherwise} \end{cases}$$

where $\sqcup$ is list concatenation, π is the tuple projection function and α_{UNIQUE} denotes an instance of α with a globally unique index that starts with 1 and is incremented by 1 with each instantiation left-to-right.

Observe that, due to the incremental assignment of indices to generate α-instances, $f(u_1,\dots,u_m)$ always contains a contiguous set $\{\alpha_k,\dots,\alpha_{k+q}\}$ (for a priori unknown k and q).

Strictly speaking, neither applying `nub` nor case 1 of Δ_G would be necessary, but they minimize the number of α-instances: `nub` merges two α-instances *if their respective s-vectors are identical*, whereas case 1 eliminates an α-instance *if all terms within its s-vector* are identical.

3 Behavior

Δ_G is a generalization of Δ in a certain sense, although not a strict one. Both try to find the maximal common structure in $t_1,\dots,t_n$, but the ability to use an unbounded number of α-instances instead of just one means that Δ_G will recursively descend until it encounters a termset in which at least two terms have different heads, whereas Δ will introduce its α as soon as it is called on two different termsets where not all heads are equal, even if the heads *within* each termset are.

4 Soundness

W.l.o.g. we assume that the set of variables which occur in u is $\{\alpha_1,\dots,\alpha_k\}$.

Theorem 1. ***Soundness of*** Δ_G**.** *Let $t_1,\dots,t_n$ be terms. If $\Delta_G(t_1,\dots,t_n) = (u,(\overline{s_1},\dots,\overline{s_p}))$, then $t_i = u[\alpha_1\backslash s_{1,i},\dots,\alpha_p\backslash s_{p,i}]$ (for $1 \leq i \leq n$).*

Proof. We proceed by induction on the depth u.

Base case. u's depth is 0.
If not all $t_1, \ldots, t_n$ are equal, $\Delta_G(t_1, \ldots, t_n) = (\alpha_1, \overline{s_1} = (t_1, \ldots, t_n))$ per definition. $t_i = u[\alpha_1 \backslash s_{1,i}, \ldots, \alpha_p \backslash s_{p,i}] = \alpha_1[\alpha_1 \backslash s_{1,i}] = t_i$.
If all terms are equal, then $\Delta_G(t_1, \ldots, t_n) = (t_1, ())$. We simply get $t_i = t_1$.

Step case. Assume the soundness for a depth of $\leq d$. We show the soundess for depth $d + 1$.

If all terms are equal, $\Delta_G(t_1, \ldots, t_n) = (t_1, ())$, which is obviously correct. If not all terms are equal, $\Delta_G(t_1, \ldots, t_n) = (f(u_1, \ldots, u_m), (\overline{s_1}, \ldots, \overline{s_p}))$.

For the second case, let $\Delta_G(t_j^1, \ldots, t_j^n) = (u_j, (\overline{s_{\alpha_1}} \ldots, \overline{s_{\alpha_k}}))$. Per the IH, the soundness condition holds for $\Delta_G(t_j^1, \ldots, t_j^n)$ $(1 \leq j \leq m)$ — i.e. $t_j^i = u_j[\alpha_1 \backslash s_{\alpha_1,i}, \ldots, \alpha_k \backslash s_{\alpha_k,i}]$. Per the uniqueness constraint on instances of α, u_{j_1} and u_{j_2} will contain non-intersecting sets of α iff $j_1 \neq j_2$. If we therefore take $f(u_1, \ldots, u_m)$ and the concatenation of all $\pi_2(\Delta_G(t_j^1, \ldots, t_j^n))$, the soundness condition will still hold.

It may, however, be the case that for two different α_{j_1} and α_{j_2} $(j_1 < j_2)$, $\overline{s_{j_1}} = \overline{s_{j_2}}$. In this case, `elim` replaces α_{j_2} with α_{j_1} in $u_1, \ldots, u_m$ and deletes $\overline{s_{j_2}}$, effectively merging α_{j_1} and α_{j_2}. For all $j_3 > j_2$, α_{j_3} is renamed to α_{j_3-1}, preserving the 1-to-1 correspondence between α_i and $\overline{s_j}$ for all $j \in \{1, \ldots, n-1\}$. Since only the superfluous α_{j_2} was eliminated, the substituiton $u[\alpha_1 \backslash s_{1,i}, \ldots, \alpha_p \backslash s_{p,i}]$ (for any i) yields the same result as before and the soundness condition is still fulfilled.

`nub` then merely repeats this soundness-preserving operation until no more duplicates can be eliminated.

□

5 Completeness

To define the sense in which Δ_G is complete, we must first introduce the notion of a normal form for decompositions. For a single α, this was done in [2, Sect. 4], where a calculus for decompositions into u, S was presented. The full calculus, as well as the theoretical results of that paper will not be replicated for the case of multiple α here, but we will make use of a few analogous notions.

5.1 A calculus of decompositions for multiple α

Definition 1. ***Decomposition and substitution.*** *A term u and a list of vectors S are a decomposition and $u \circ S$ is a substitution for a set of terms*

$\{t_1, \dots, t_n\}$ *if*

$$\{t_1, \dots, t_n\} = u \circ S = \{u[\overline{\alpha} \backslash s_{i,i}, \dots, s_{q,i} | \ 1 \leq i \leq n]\}$$

where $\overline{\alpha} = (\alpha_1, \dots, \alpha_q)$, $u[\overline{\alpha} \backslash (s_{1,j}, \dots, s_{q,j})] = u[\alpha_1 \backslash s_{1,j}, \dots, \alpha_q \backslash s_{q,j}]$, *s.t.*

1. *S does not contain any* α_i *and*
2. *the variables occurring in* u *are numbered* $\alpha_1, \dots, \alpha_q$ *left-to-right.*

Definition 2. ***Left-shifting.*** *A decomposition may be left-shifted if, for some* α_i*, all terms in the list* s_i *start with a common function symbol* f *of arity* r*. Let* $s_i = (f(a_{1,1}, \dots, a_{1,r}), \dots, f(a_{n,1}, \dots, a_{n,r}))$*. Then, left-shifting for* α_i *is defined as:*

$$\frac{u \circ S}{u[\alpha_i \backslash f(\alpha_i^1, \dots, \alpha_i^r)] \circ S[\overline{s_i} \backslash (a_{1,1}, \dots, a_{n,1}), \dots, (a_{1,r}, \dots, a_{n,r}))]} \leftarrow$$

where $\alpha_i^1, \dots, \alpha_i^r$ *are fresh variable names.*

Example. *Let* $\{t_1, \dots, t_n\} = \{f(g(a,b),x), f(g(c,d),y)\}$ *and let* $u = f(\alpha_1, \alpha_3)$, $S = (\overline{s_1}, \overline{s_2})$ *with* $\overline{s_1} = (g(a,b), g(c,d)$, $\overline{s_2} = (x,y)$.
We can left-shift for α_1*:*

$$\frac{f(\alpha_1, \alpha_3) \circ (\overline{s_1}, \overline{s_2})}{f(g(\alpha_1, \alpha_2), \alpha_3) \circ ((a,c),(b,d),(x,y))} \leftarrow$$

Definition 3. ***Merging & splitting*** α***.*** *Suppose that there exist* α_i *and* α_j *in* u *(*$i \neq j$*) s.t.* $\overline{s_i} = \overline{s_j}$*. Then we can replace* α_j *with* α_i *and* α_k *with* α_{k-1} $(k > max\{i,j\})$ *in* u *and delete* $\overline{s_j}$ *from* S*. Conversely, we can rename multiple occurrences of* α_i *to* $\alpha_{i_1}, \dots, \alpha_{i_n}$ *and duplicate* $\overline{s_i}$ n *times. These operations are called merging & splitting and obviously preserve the result of* $\circ_{\overline{\alpha}}$*.*

Definition 4. α***-elimination.*** *Suppose that, for a decomposition* u, S*, there exists an* α_i *in* u *s.t. all terms in* $\overline{s_i}$ *are equal, i.e.* $\overline{s_i} = (s_{i,1}, \dots, s_{i,1})$*. Then we can eliminate* α_i *by replacing* u *with* $u[\alpha_i \backslash s_{i,1}]$ *and deleting* $\overline{s_i}$ *from* S *without changing the result of* $\circ_{\overline{\alpha}}$*.*

Definition 5. ***Normal form.*** *A decomposition* u, S *is in normal form iff no left-shift, no merging and no* α*-elimination are possible.*

Theorem 2. *Every decomposition* u, S *has a unique normal form.*

Proof. 1. Observe that the order in which merging operations and α-eliminations are applied is irrelevant.

2. The order in which two left-shift operations are applied is irrelevant. Suppose α_{j_1} and α_{j_2} occur in u and we can left-shift for both. Left-shifting for α_{j_1} only replaces α_{j_1} in u with a term containing fresh $\alpha^1_{j_1}, \dots, \alpha^r_{j_1}$ and the vector $\overline{s_{j_1}}$ in S with new vectors $\overline{s^1_{j_1}}, \dots, \overline{s^r_{j_1}}$. The applicability of a left-shift for α_{j_2} remains unaffected.

3. Left-shifting and α-elimination commute. Suppose both left-shifting and α-elimination can be performed on α_j, i.e. $\overline{s_j} = (f(s_{j,1}, \dots, s_{j,r}), \dots, f(s_{j,1}, \dots, s_{j,r}))$. α-elimination replaces α_j with $f(s_{j,1}, \dots, s_{j,r})$ in u and deletes $\overline{s_j}$ in S. Left-shifting replaces α_j with $f(\alpha^1_j, \dots, \alpha^r_j)$ in u and $\overline{s_j}$ with $(s_{j,1}, \dots, s_{j,1}) \dots, (s_{j,r}, \dots, s_{j,r})$. α-elimination can then be performed on $\alpha^1_j, \dots, \alpha^r_j$, giving the same result.

4. Left-shifting and merging commute. Suppose that α_{j_1} and α_{j_2} occur in u s.t. α_{j_1} and α_{j_2} can be merged and left-shifting is possible on both. Since merging is possible, $\overline{s_{j_1}} = \overline{s_{j_2}} = (s_{j,1}, \dots, s_{j,n})$. Merging α_{j_1} and α_{j_2} replaces u with $u[\alpha_{j_2} \backslash \alpha_{j_1}]$ and deletes $\overline{s_{j_2}}$ from S. If we left-shift on α_{j_1} and α_{j_2}, we introduce $\alpha^1_{j_1}, \dots, \alpha^r_{j_1}$ and $\alpha^1_{j_2}, \dots, \alpha^r_{j_2}$ with corresponding s-vectors $\overline{s^1_{j_1}}, \dots, \overline{s^r_{j_1}}$ and $\overline{s^1_{j_2}}, \dots, \overline{s^r_{j_2}}$ s.t. $\overline{s^i_{j_1}} = \overline{s^i_{j_2}}$ $(1 \leq i \leq r)$. Merging $(\alpha^1_{j_1}, \alpha^1_{j_2}), \dots, (\alpha^r_{j_1}, \alpha^r_{j_2})$ then delivers the same result as merging $(\alpha_{j_1}, \alpha_{j_2})$ did.

$\square$

Corollary 1. *If $t_1 = \dots = t_n$ and $n > 0$, the normal form of a decomposition for $(t_1, \dots, t_n)$ is $t_1, ()$.*

Proof. Take the trivial decomposition $\alpha_1, (t_1, \dots, t_n)$ and perform α-elimination. $\square$

5.2 Completeness proof

Lemma 1. *nub* performs all merges. *W.l.o.g. let $u, S = (\overline{s_1}, \dots, \overline{s_q})$ be a decomposition in which the variables $\alpha_1, \dots, \alpha_q$ occur. Then $\texttt{nub}(u, S) = u', S'$ such that u', S' equals u, S modulo merging and that no merge is possible in u', S'.*

Proof. We show that if a merge is possible, `elim` performs it.

Per definition, merging two distinct variables α_i, α_j is possible iff $\overline{s_i} = \overline{s_j}$. This is equivalent to the condition for the application of `elim`: $[\exists i, j : i < j]\ \overline{s_i} = \overline{s_j}$. If this condition holds, `elim` performs the two actions given in the definition of merging: it

1. removes $\overline{s_j}$ and

2. replaces α_j with α_i and α_k with α_{k-1} $(k > j)$ in u.

`nub` is the least fixed point of `elim`, and since `elim` merges two variables α_i and α_j iff they can be merged, it is shown that `nub` performs exactly every possible merge in u, S. □

Theorem 3. ***Completeness of*** Δ_G**.** *If* u, S *is a decomposition for a non-empty set of terms* $\{t_1, \ldots, t_n\}$*, then* $\Delta_G(t_1, \ldots, t_n) = (u', S')$ *s.t.* u', S' *is the normal form of* u, S*.*

Proof. W.l.o.g. we assume a contiguous numbering of the α occurring in u. We proceed by induction on the depth of $t_1, \ldots, t_n$.

Base case. Let all t_i $(1 \leq i \leq n)$ have a depth of 0. We have two sub-cases:

1. All terms are equal. This is the condition of the first case of Δ_G, which, per Corollary 1, is the normal form of u, S.

2. Not all terms are equal. The terms can only be constants. In either case, the only possible decomposition is $u = \alpha, S = (t_1, \ldots, t_n)$. This is exactly the "otherwise"-case of Δ_G.

Step case. Assume that Δ_G is complete if all t_i $(1 \leq i \leq n)$ have a depth $\leq d$. We again have two cases:

1. All terms are equal. As in the base case, Δ_G computes the normal form of u, S, per Corollary 1.

2. Not all terms are equal. If not all terms begin with a common function symbol with arity m, the only possible decomposition is $u = \alpha, S = (t_1, \ldots, t_n)$. This is again exactly the "otherwise"-case of Δ_G. If all terms do begin with a common function symbol of arity m, then $t_i = f(t^i_1, \ldots, t^i_m)$ for $(1 \leq i \leq n)$. This is the condition for the second case of Δ_G. u can take one of two forms:

   (a) $u = \alpha$. Since a left-shift is possible due to the common function symbol f, such a decomposition is not in normal form. If we left-shift, case 2.(b) applies.

   (b) $u = f(u_1, \ldots, u_m)$. In this case, u_i, S_i $(1 \leq i \leq m)$ must be a decomposition for the terms $t^1_i, \ldots, t^n_i$. For these decomposition, completeness of Δ_G holds per the IH. Now let $\widehat{u} = f(\widehat{u_1}, \ldots, \widehat{u_m}), \widehat{S} = \bigsqcup_{1 \leq i \leq m} S_i$ be the decomposition created by simply concatenating all u_i, S_i, where $\widehat{u_i}$ is u_i, with its variables renamed to avoid clashes between any two u_{i_1}, u_{i_2}. Per definition of Δ_G, $\widehat{u}, \widehat{S}$ is precisely the result of the second case of Δ_G, without the application of `nub`. Furthermore, per Theorem 2, u, S and $\widehat{u}, \widehat{S}$ have the same normal form.

Due to the IH, $\widehat{u}, \widehat{S}$ can only differ from u', S' through merging. Per Lemma 1, the application of `nub` to $\widehat{u}, \widehat{S}$ performs all possible merges, which results in the unique normal form of both $\widehat{u}, \widehat{S}$ and u, S.

□

Corollary 2. *Every non-empty set of terms $\{t_1, \ldots, t_n\}$ has exactly one decomposition in normal form and Δ_G computes it.*

Proof. Follows from Theorems 1, 2 & 3 and from Δ_G being a total function. □

6 Δ-Table

The Δ-table, as described in [1], is compatible with the Δ_G-vector and, save for substituting for multiple α instead of one, can be left as is.

7 Remarks

The introduction of a global, contiguous numbering of α-instances is very procedural and bloats the definition almost to the point of being pseudocode, but the precise definition of the semantics of Δ_G and the continuous & unique labeling and and re-labeling of α-instances regrettably make such an algorithmic approach necessary.

8 Extensions

For theoretical reasons, we might be interested in limiting the number of allowed α (bounded Δ_G).

8.1 Bounded generalized Δ-Vector

Δ_G will compute a unique decomposition that will preserve as much of the common structure of $t_1, \ldots, t_n$ as possible, employing as many α-instances as needed, but in some cases, it may be desirable to limit the number of such α-instances that it may use. Consider the following example:

$$\Delta_G(f(g(a,b),g(c,d),e),f(g(x,y),g(u,v),w)) =$$

$$(f(g(\alpha_1,\alpha_2),g(\alpha_3,\alpha_4),\alpha_5);(a,x),(b,y),(c,u),(d,v),(e,w))$$

This illustrates two points:

1. There is a non-deterministic choice: we could restrict the number of α-instances to, say, 4. This can be achieved by right-shifting α_1 and α_2 into a new α' (with the terms $(g(a,b),g(x,y))$) or α_3 & α_4 (with the terms $(g(c,d),g(u,v))$).

2. Generally, we cannot say that, for any k, decompositions with exactly k variables exist. For example, if we restrict the number of α-instances to at most 2 in u, we only get the trivial decomposition $(\alpha;f(g(a,b),g(c,d),e),f(g(x,y),g(u,v),w))$.

We can see that we can specify upper, but not lower bounds on the number of variables which may be used, although this burdens us with a non-deterministic choice as to for which α-instances to right-shift.

Calculating the set of all decompositions with at most k variables

Δ_G can be slightly altered to compute decompositions with at most k variables. We do this by introducing a parameter k and a non-deterministic boolean variable **RANDOM**, creating Δ_G^k:

$$\Delta_G^k(t_1,\dots,t_n) = \begin{cases} (t_1,()) \text{ if } t_1 = t_2 = \cdots = t_n \text{ and } n > 0 \\[1em] \texttt{nub}(f(u_1,\dots,u_m),(\overline{s_1},\dots,\overline{s_q})) \\ \quad \text{if all} \quad t_i = f(t_1^i,\dots,t_m^i)\text{, case 1 does not apply, and} \\ \qquad\qquad ((\textbf{RANDOM} = \text{true and } |\{\overline{s_1},\dots,\overline{s_q}\}| \le k) \text{ or} \\ \qquad\qquad m = 1) \\[1em] \quad \text{where} \quad (\overline{s_1},\dots,\overline{s_q}) = \bigsqcup_{1\le j\le m} \pi_2(\Delta_G^k(t_j^1,\dots,t_j^n)) \text{ and} \\ \qquad\qquad u_j = \pi_1(\Delta_G^k(t_j^1,\dots,t_j^n)) \text{ for all } j \in \{1,\dots,m\} \\[1em] (\alpha_{\text{UNIQUE}},(t_1,\dots,t_n)) \quad \text{otherwise} \end{cases}$$

The side-condition for the instantation of α_{UNIQUE} is the same as in Δ_G: the leftmost occurrence is α_1 and all instances are numbered incrementally left-to-right.

The second case of Δ_G^k now has an additional condition: first, either the function symbol f at the head of $t_1, \dots, t_n$ is unary, or **RANDOM** is true and, after the application of `nub`, there are at most k s-vectors. If we have a unary function symbol, we can always choose the second case because doing so does not immediately necessitate more than one variable. If the function symbol is not unary, we have two conditions: **RANDOM** has to be true and the returned decomposition may not contain more than k variables. The purpose of this latter is obvious, whereas **RANDOM** accounts for the non-deterministic nature of right-shifting: randomly terminating our search for a common term structure early is equivalent to first computing Δ_G and then non-deterministically right-shifting to reduce the number of variables.

It is easy to see that decompositions computed by Δ_G^k need not be in normal form. A simple example is the following:

$$\Delta_G^\infty(f(a), f(b)) = \{(\alpha_1, ((f(a)), (f(b)))), (f(\alpha_1), ((a), (b)))\}$$

Due to its non-determinism, Δ_G^∞ computed the trivial decomposition, even though a left-shift is possible on it. To talk about the result of Δ_G^k, we therefore need the notion of a weaker normal form:

Definition 6. ***Weak k-normal form.*** *A decomposition u, S is in weak k-normal form if $\leq k$ variables occur in u, merging and α-elimination are not possible, and any left-shifting would result in $> k$ variables in u.*

Thus equipped, we can give the following correctness result:

Theorem 4. ***Soundness of Δ_G^k.*** *Let $t_1, \dots, t_n$ be terms and let $k \geq 1$. If $u, S \in \Delta_G^k(t_1, \dots, t_n)$, then u, S is in weak l-normal form (for some $l \in \{1, \dots, k\}$) and $t_i = u[\alpha_1 \backslash s_{1,i}, \dots, \alpha_p \backslash s_{p,i}]$ ($1 \leq i \leq n$).*

Proof. The proof is largely analogous to that given in Theorem 1. Since Δ_G^k only differs from Δ_G in that it non-deterministically chooses its "otherwise"-case, we need only show two additional claims:

1. u, S is in weak l-normal form for some $1 \leq l \leq k$ and
2. choosing the "otherwise"-case where Δ_G would choose its second case still results in a decomposition.

For the first claim, note that the condition $|\{\overline{s_1}, \dots, \overline{s_q}\}| \leq k$ in the second case of Δ_G^k forces the "otherwise-case" to be chosen whenever more than k distinct variables occur in some subtree of u, including u itself. Therefore, Δ_G^k will never return a decomposition with more than k variables. Now let l be the number of variables occurring in u.

We also annot perform a left-shift without increasing the number of variables in u, as is evident by examining the s-vectors of S. We distinguish three cases for the form of $s \in S$:

1. $s = (c_1, \ldots, c_n)$ where not all c_i have a common function symbol of the same arity. In this case, no left-shift is possible at all.

2. $s = (f(s_1), \ldots, f(s_n))$. Such an s-vector would permit a left-shift without increasing the number of variables. However, such an s cannot exist because it would imply that $\Delta_G^k(f(s_1), \ldots, f(s_n))$ chose its "otherwise"-case. The condition $m = 1$ in the second case of Δ_G^k precludes that.

3. $s = (f(s_1^1, \ldots, s_1^r), \ldots, f(s_n^1, \ldots, s_n^r))$ $(r > 1)$. If we were to left-shift for the corresponding variable, we would increase the number of variables by $r - 1$, resulting in at least $l + 1$ variables. Consequently, u, S would no longer be in weak l-normal form.

The second claim is clearly true: if we choose the "otherwise"-case and instantiate every new variable α_i with its s-vector $(t_1, \ldots, t_n)$, then $(\alpha_i \backslash t_1, \ldots, \alpha_i \backslash t_n) = (t_1, \ldots, t_n)$ and $u \circ S = (t_1, \ldots, t_n)$. $\square$

Theorem 5. ***Completeness of*** Δ_G^k***.*** *If we evaluate for all possible assignments for* ***RANDOM****,* Δ_G^k *returns the set of all decompositions which are in weak* l*-normal form (for some* $l \in \{1, \ldots, k\}$*).*

Proof. Let $l \in \{1, \ldots, k\}$ and let u, S be a decomposition in weak l-normal form.

We first note that the condition $|\{\overline{s_1}, \ldots, \overline{s_q}\}| \leq k$ is *necessary* for weak l-normal forms: if there are more than k distinct variables (evidenced by the distinct s-vectors), these cannot be merged with each other, and the decomposition cannot be in weak l-normal form $(1 \leq l \leq k)$.

Now we proceed by induction on the maximal depth of $t_1, \ldots, t_n$. The proof is entirely analogous to that for Theorem 3, except for point 2 of the step case, which will be given here.

The induction hypothesis is that Δ_G^k is complete for depth $\leq d$. In point 2 of the step case, assuming that not all terms are equal, we distinguish two sub-cases:

1. $t_1, \ldots, t_n$ start with a common unary function symbol, i.e. $m = 1$ and $t_1, \ldots, t_n = f(t_1^1, \ldots, t_n^1)$. Then, u can only be of the form $u = f(u_1)$, since it wouldn't be in weak l-normal form otherwise. This corresponds to the second case of Δ_G^k, since $m = 1$ is true. Per the IH, Δ_G^k is complete for $t_1^1, \ldots, t_1^n$.

2. $t_1, \ldots, t_n$ start with a common function symbol of arity $m > 1$. u may either be of the form $u = \alpha$ or $u = f(u_1, \ldots, u_m)$. The first form

corresponds to the "otherwise"-case. The second form corresponds to the second case of Δ_G^k and the IH applies in the same manner as it applied in the step case of Δ_G's completeness proof. Due to **RANDOM** being non-deterministic, both forms are chosen and hence Δ_G^k is complete.

$\square$

Unlike the normal form, weak k-normal forms are not unique, as the following example shows:

$$\Delta_G^\infty(f(g(a,b),g(c,d),e),f(g(x,y),g(u,v),w)) =$$

$$\{(f(\alpha_1,\alpha_2,\alpha_3);((g(a,b),g(x,y)),(g(c,d),g(u,v)),(e,w)),\dots\}$$

As we can see, this is the same termset as in the previous example, but here, we have terminated the search early in the case of α_1 and α_2. The decomposition shown has 3 variables. If we want to get to weak 4-normal form, we can left-shift either for α_1 and α_2, but not for both. Therefore, we can create two different weak normal forms.

Another property is that one weak k-normal form might reduce to another weak k-normal form, without the intermediate decompositions being in weak k-normal form themselves. This simple example illustrates:

$$(f(\alpha_1,\alpha_3),((g(a,c),g(b,d)),(a,b))),$$
$$(f(g(\alpha_1,\alpha_2),\alpha_1),((a,b),(c,d)),$$

Both decompositions are in weak 2-normal form. However, by left-shifting for α_1 in the first one, we get

$$(f(g(\alpha_1,\alpha_2),\alpha_3),((a,b),(c,d),(a,b))$$

and, after merging α_1 and α_3:

$$(f(g(\alpha_1,\alpha_2),\alpha_1),((a,b),(c,d))$$

The intermediate $(f(g(\alpha_1,\alpha_2),\alpha_3),((a,b),(c,d),(a,b))$, however, is not in weak 2-normal form.

The definition of weak normal forms could be altered to forbid the possibility of such reductions instead of simply the possibility of left-shifts, but a conscious choice was made against it: decompositions such as the two in this example, though related, are materially different, and, for the purpose of finding "intuitive" cut-formulas, it might be advantageous to consider both.

Bibliography

[1] Stefan Hetzl, Alexander Leitsch, Giselle Reis, and Daniel Weller. Algorithmic introduction of quantified cuts. submitted to Theoretical Computer Science, 2013.

[2] Stefan Hetzl, Alexander Leitsch, and Daniel Weller. Towards algorithmic cut-introduction. In *Proceedings of the 18th international conference on Logic for Programming, Artificial Intelligence, and Reasoning*, LPAR'12, pages 228–242, Berlin, Heidelberg, 2012. Springer-Verlag.