A complete proof system for nested term graphs

Stefan Blom
Department of Computer Science
Vrije Universiteit Amsterdam
de Boelelaan 1081
1018 HV Amsterdam
The Netherlands

Extended abstract: May 1, 1995

1 Syntax

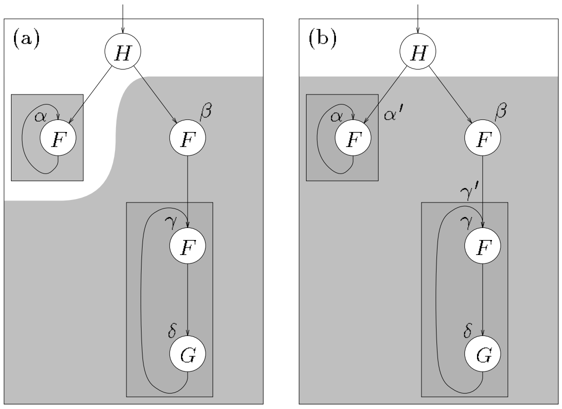
In their report [1], Zena Ariola and Jan Willem Klop represent term graphs as sets of recursive equations. In practice we will want a 'modular' structure as given by the let and letrec constructs. This leads to the notion of Nested Term Graph. Informally a Nested Term Graph, or NTG, is a term where at any point variables may locally be (recursively) defined to have a certain value. We will adopt the convention that $\mathcal{V}$ is a set of variables with typical members $\alpha, \beta, \gamma, \delta, \alpha_1, \beta^2, \ldots$ and that $\mathcal{F}$ is a set of function symbols $f, g, h, F, G, H, c, d, \ldots$ each with a given fixed arity. By $arity(f)$ we mean the arity of the funtion symbol f, constants are function symbols with arity 0. There is a special constant $\bullet$, the *black hole*.

Definition 1.1 A *nested term graph* can be:

- α, for each $\alpha \in \mathcal{V}$
- $f(t_1, \ldots, t_n)$ for each n-ary $f \in \mathcal{F}$ and NTG's $t_1, \ldots, t_n$.
- $\langle s | \alpha_1 = t_1, \ldots, \alpha_n = t_n \rangle$, for NTG's $s, t_1, \ldots, t_n$ and variables $\alpha_1, \ldots, \alpha_n$ which are pairwise different.

We will now introduce some terminology and notation. In stead of $\langle s|E \rangle$ we will often write s^E. Given a nested term graph of the form $\langle s | \alpha_1 = t_1, \ldots, \alpha_n = t_n \rangle$, we refer to the entire NTG as a *box*, to the s as the *external part*, and to the set of equations $\alpha_1 = t_1, \ldots, \alpha_n = t_n$ as the *internal part* or the *environment*. The symbols $E, F, \ldots$ are normally used to match environments. A nested term graph built only with the first two clauses is a *term*. A nested term graph is *guarded* if it is of the form $f(t_1, \ldots, t_n)$. A nested term graph of the form α^E where all right-hand sides of the equations in E are guarded terms is called a *term graph*. So a term graph has no nesting! By $def(E)$ we mean the set of left-hand sides from the environment E and by $free(s)$ we mean the set of variables that has at least one free occurrence in the NTG s.

We say that that an occurrence of a variable is a free occurrence, if the occurrence is not inside a box, where the variable occurs as the left-hand side of an equation in the environment. If a variable occurs as the left-hand side of an equation in the internal part, then that occurrence is a binding occurrence and all free occurrences of the variable in the external part and in the rigth-hand sides of the environment are bound by this left-hand side occurrence.



(a) $\langle H(\langle\alpha|\alpha = F(\alpha)\rangle, \beta)|\beta = F(\langle\gamma|\gamma = F(\delta), \delta = G(\gamma)\rangle)\rangle$
(b) $\langle H(\alpha', \beta)|\alpha' = \langle\alpha|\alpha = F(\alpha)\rangle, \beta = F(\gamma'), \gamma' = \langle\gamma|\gamma = F(\delta), \delta = G(\gamma)\rangle\rangle$

Example 1.1 $\big\langle f(\alpha_1, \big\langle f(\alpha_2, \beta_1) \bigm| \alpha_3 = f(\alpha_4, \gamma), \beta_2 = \bullet\big\rangle) \bigm| \alpha_5 = f(\alpha_6, \beta_3)\big\rangle$

Here the subscripts denote the number of the occurrence. The only occurrence of γ is a free occurrence, the 1st occurrence of α is bound to the 5th occurrence, the 2nd and 4th occurrences of α are bound to the 3rd occurrence, the 1st occurrence of β is bound to the 2nd and the 3rd occurrence of β is free.

When t is obtained from s by replacing in s a binding occurrence of a variable and all occurrences bound to that occurrence by a variable that does not occur is s, we say that $s =_\alpha t$. We denote by the same $=_\alpha$ the transitive reflexive closure of the relation and we name it *α-conversion*. A nested term graph is maximally α-converted if no variable occurs both free and as a binder and no variable occurs twice as a binder. Unless stated otherwise we will not distinguish between two α-convertible nested term graphs.

A context $C[.]$ is a NTG with one occurrence of a constant replaced by $\Box$. By $C[s]$ we mean $C[.]$ with the box replaced by s. So $\big\langle \Box \bigm| \alpha = f(\alpha)\big\rangle$, $\big\langle \alpha \bigm| \alpha = f(\Box)\big\rangle$ and $f(\Box)$ are contexts, but $\big\langle \Box \bigm| \alpha = f(\Box)\big\rangle$ and $\big\langle \alpha \bigm| \Box = f(\alpha)\big\rangle$ are not.

A *substitution* is a partial function σ from $\mathcal{V}$ to the set of NTG's, which extends to NTG's by the following rules:

$$\begin{aligned}
\alpha\sigma &= \begin{cases} \sigma(\alpha) & \text{, if } \alpha \in D_\sigma \\ \alpha & \text{, otherwise} \end{cases} \\
f(t_1, \ldots, t_n)\sigma &= f(t_1\sigma, \ldots, t_n\sigma) \\
\langle s|\alpha_1 = t_1, \ldots, \alpha_n = t_n\rangle\sigma &= \langle s\sigma|\alpha_1 = t_1\sigma, \ldots, \alpha_n = t_n\sigma\rangle
\end{aligned}$$

under the restriction that a substitution is only applicable to a NTG if no variable that occurs free in some NTG in the image of the substitution occurs as a binder in the NTG. This is easily guaranteed using α-conversion.

2 Bisimulation

We now introduce a semantical notion of equivalence between NTG's. Roughly this notion is tree equivalence, i.e. NTG's are equivalent when they unwind to the same (possibly infinite) tree. It turns out (see [1]) that this notion is conveniently described by the notion of bisimulation, well-known in process algebra. We do not want bisimulation to be influenced by differences in boxing. So as far as bisimulation is concerned the two graphs in the picture are supposed to be the same.

Definition 2.1 We define the equivalence relation $\sim$ on occurrences of variables and function symbols as the least equivalence such that:

- $\alpha_1 \sim \alpha_2$, if both occurrences of α are free.
- $\alpha_1 \sim \alpha_2$, if the first occurrence is bound to the second.
- If we have $\alpha = t$ somewhere then $\alpha \sim head(t)$.

Subscripts denote different occurrences and $head(t)$ is simply the leftmost function or variable symbol. Examples of all cases are:

- $\alpha_1 \sim \alpha_2$, in $f(\alpha_1, \alpha_2)$.
- $\alpha_1 \sim \alpha_2$, in $\langle \alpha_1 | \alpha_2 = \bullet \rangle$
- $\alpha_2 \sim \bullet$ in $\langle \alpha_1 | \alpha_2 = \bullet \rangle$

On the equivalence classes we define:

$$\begin{aligned} label(C) &= f\text{, if an occurrence of the function symbol } f \text{ is in } C \\ label(C) &= \alpha\text{, if a free occurrence of the variable } \alpha \text{ is in } C \\ label(C) &= \bullet\text{, otherwise} \end{aligned}$$

There is at most one occurrence of a function symbol in a class. If two free occurrences of a variable are in the same class, the variables are the same. If a function symbol occurrence is in a class there is no occurrence of a free variable in that class.

If $f(t_1, \ldots, t_n)$ is a subterm occurrence then for $i = 1, \ldots, n$:

$[f] \rightarrow_i [head(t_i)]$

A relation R on the equivalence classes of NTG's s and t is a bisimulation if

- $[head(s)]R[head(t)]$
- $cRd \Rightarrow label(c) = label(d)$
- $c_1 \rightarrow_i c_2$, $c_1 R d_1$ and $d_1 \rightarrow_i d_2$ implies that $c_2 R d_2$.

If a bismulation R between s and t exists we write $s \underline{\leftrightarrow} t$.

3 The proof system

We assume that the expressions occurring in the proof system are maximally α-converted, i.e. all variables occurring as left-hand sides are pairwise different, also if they occur in different environments, and there is no overlap between free and bound variables. The system is in the same style as the usual system for equational logic.

Definition 3.1 We say that $\models s = t$ if $s \underline{\leftrightarrow} t$ and that $\vdash s = t$ if there is a list of equations $E_1, E_2, \ldots E_n$, where $E_n \equiv s = t$, such that each equation E_i is either an instantiation of an axiom in the proof system or the result of an instantiation of a rule where the instantiations of the premises of the rule are present in the list prior to E_i itself.

Proof system I: A proof system for NTG's

A1	$t = t$	*reflexivity*
A2	$\langle s \mid \alpha = \alpha, E\rangle = \langle s \mid \alpha = \bullet, E\rangle$	*black hole*
A3	$C[\alpha] = C[t]$, if $C[\alpha] = \langle s \mid E\rangle$ with $\alpha = t \in E$	*substitution*
A4	$\langle s \mid \rangle = s$	*box removal*
A5	$f(s_1, \ldots\ldots, s_n)^E = f(s_1^E, \ldots\ldots, s_n^E)$	*unwinding*
A6	$s^{E,F} = s^E$, if $def(F) \cap free(s^E) = \emptyset$	*garbage collection*
R1	$\dfrac{s = t}{t = s}$	*symmetry*
R2	$\dfrac{s = t \quad t = u}{s = u}$	*transitivity*
R3	$\dfrac{s = t}{C[s] = C[t]}$	*context*
R4	$\dfrac{s_1 = t_1\{\vec{s}/\vec{\alpha}\} \ldots\ldots s_n = t_n\{\vec{s}/\vec{\alpha}\}}{s_1 = \langle \alpha_1 \mid \vec{\alpha} = \vec{t}\rangle}$, for guarded $t_1, \ldots\ldots, t_n$	*folding*

Example 3.1 We will prove within the system that:
$\langle f(\alpha) \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle = \langle \alpha \mid \alpha = f(\alpha)\rangle$

1	$\langle f(\alpha) \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle = f(\langle \alpha \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle)$	unwinding
2	$\langle \alpha \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle = \langle\langle \beta \mid \beta = f(\alpha)\rangle \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle$	substitution
3	$\langle \beta \mid \beta = f(\alpha)\rangle = \langle f(\alpha) \mid \beta = f(\alpha)\rangle$	substitution
4	$\langle f(\alpha) \mid \beta = f(\alpha)\rangle = \langle f(\alpha) \mid \rangle$	garbage collection
5	$\langle f(\alpha) \mid \rangle = f(\alpha)$	box removal
6	$\langle \beta \mid \beta = f(\alpha)\rangle = f(\alpha)$	transitivity, 3, 4, 5
7	$\langle\langle \beta \mid \beta = f(\alpha)\rangle \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle = \langle f(\alpha) \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle$	context, 6
8	$\langle f(\alpha) \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle = f(\langle \alpha \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle)$	unwinding
9	$\langle \alpha \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle = f(\langle \alpha \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle)$	transitivity, 2, 7, 8
10	$\langle \alpha \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle = \langle \delta \mid \delta = f(\delta)\rangle$	folding, 9
11	$f(\langle \alpha \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle) = f(\langle \delta \mid \delta = f(\delta)\rangle)$	context, 10
12	$\langle f(\alpha) \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle = f(\langle \delta \mid \delta = f(\delta)\rangle)$	transitivity, 1, 11
13	$\langle \alpha \mid \alpha = f(\alpha)\rangle = \langle f(\alpha) \mid \alpha = f(\alpha)\rangle$	substitution
14	$\langle f(\alpha) \mid \alpha = f(\alpha)\rangle = f(\langle \alpha \mid \alpha = f(\alpha)\rangle)$	unwinding
15	$\langle \alpha \mid \alpha = f(\alpha)\rangle = f(\langle \alpha \mid \alpha = f(\alpha)\rangle)$	transitivity, 13, 14
16	$f(\langle \alpha \mid \alpha = f(\alpha)\rangle) = \langle \alpha \mid \alpha = f(\alpha)\rangle$	symmetry, 15
17	$\langle f(\alpha) \mid \alpha = \langle \beta \mid \beta = f(\alpha)\rangle\rangle = \langle \alpha \mid \alpha = f(\alpha)\rangle$	transitivity, 12, 16

Theorem 3.1 $\vdash s = t$ *iff* $\models s = t$.

Soundness is proven for axioms by giving bisimulations for the left- and right-hand sides and for rules by giving bisimulation in terms of the bisimulations for the premises, these constructions can be given formally and illustrated by pictures, both ways of description too big to fit here.

Completeness is proven following the general pattern already present in the example: finding a NTG to wich both NTG's can be proven equal. In the general case this will be done by first proving both NTG's equal to a term graph and then proving these term graphs equal. This last step is performed by constructing a third term graph from the bisimulation between the first two and proving each of them equal to the third.

References

[1] Z.M. Ariola and J.W. Klop. Equational Term Graph Rewriting. CWI-report CS-R95xx.