Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science 72 (2007) 75–86
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Package Duplication in Interaction Nets and Weak Head Reduction in the lambda-calculus

Sylvain Lippi[1]

Institut de Mathématiques de Luminy
Marseille, France

Abstract

We present a simple implementation of weak head reduction in the λ-calculus with interaction nets using package duplication.

Keywords: Weak head reduction, interaction nets, lambda calculus

1 Introduction

Interaction nets introduced by Yves Lafont [5] can be considered as a programming language with a real interpreter [9] or as a graphical model of computation. In the second point view, a typical application is the encoding of optimal reduction for the λ-calculus [1]. Those translations are usually complicated giving quite complex nets during the reduction. We propose to focus on weak head reduction. The consequence is that we shift the problem to the duplication of some classes of nets: packages. We present two results on the duplication of packages: a negative one which sets the intrinsic limits of the interaction net paradigm and a positive one by exhibiting a very simple translation of λ-terms.

2 The interaction net paradigm

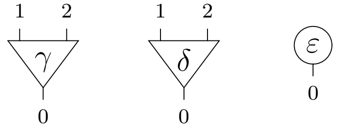
We present the interaction net paradigm in a practical way by introducing a simple example: the interaction combinators. For a more detailed presentation see [5].

[1] lippi@iml.univ-mrs.fr

1571-0661
doi:10.1016/j.entcs.2002.09.006

2.1 Graphical syntax

- The basic ingredient is a *symbol* with its *arity*. Our example system has three symbols: δ and γ of arity 2 and ε of arity 0.
- Occurrences of symbols are called *cells*. Cells have one *principal port* and their number of *auxiliary ports* is given by the arity of their corresponding symbol. The γ-, δ- and ε-cells are pictured like this:



  γ- and δ-cells have three *ports*: one *principal port* (0) and two *auxiliary ports* (1 and 2). ε-cells have only one (principal) port. For esthetic reasons, cells with no auxiliary port are pictured with a circle; the important point is to distinguish the principal port of a cell.

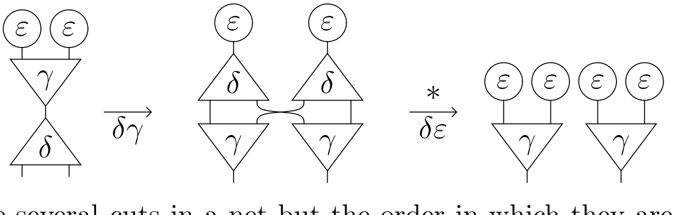
  Convention. Auxiliary ports are not interchangeable. For instance, one can number them from 1 to n, keeping 0 for the principal port. In practice, the ports will always be implicitly numbered in clockwise order.

- A *net* is a graph built with cells and *free ports*. Ports (principal ports, auxiliary ports or free ports) are connected pairwise by *wires*. An example of a net built with the combinators can be found in figure 1

- A *rule* is a pair of nets (*left member* $\rightarrow$ *right member*) with the same number of free ports. The important restriction is that the left member must be built with two cells connected by their principal ports; such a net is called a *cut*. [2] There is, at most, one rule for each pair of symbols. Figure 2 gives the rules for the combinators.

2.2 Execution

Once a set of symbols and rules has been fixed, we can apply one of those rules to a net obtaining another net and so on until we have reached an irreducible one. The *reduction* relation is denoted by $\rightarrow$ and its reflexive and transitive closure by $\xrightarrow{*}$. Here is an example of reduction where a net is duplicated by a δ-cell:



There may be several cuts in a net but the order in which they are eliminated does not matter. The reason is that two instances of a cut are necessarily disjoint,

[2] if the two cells of the left member share the same symbol, there is also a symmetry condition on the right member (see [6]).