# Algorithmic specifications in linear logic with subexponentials 

Vivek $\mathrm{Nigam}^{\dagger}$ and Dale Miller<br>INRIA-Saclay \& LIX/École Polytechnique Palaiseau, France

3 December 2010, LIX

Based on a paper with the same title that appeared in PPDP 2010. $\dagger$ Now on a Humboldt Fellowship at L.U.M. Munich.

## Overview of high-level goals

- Use focused proof systems to justify algorithmic interpretation of proof search
- Use subexponentials to locate data and processes.
- Describe a algorithmic programming language based on located multisets
- Speculate on parallel and distributed computing and bigraphs.


## Outline

Synthetic connectives and focused proofs in linear logic

The subexponentials of linear logic

Subexponentials and explicit algorithm specification

## Outline

Synthetic connectives and focused proofs in linear logic

The subexponentials of linear logic

## Subexponentials and explicit algorithm specification

The LLF focused proof system for linear logic
Negative rules

$$
\begin{gathered}
\frac{\Psi ; \Delta \Uparrow L}{\Psi ; \Delta \Uparrow \perp, L}[\perp] \quad \frac{\Psi ; \Delta \Uparrow F, G, L}{\Psi ; \Delta \Uparrow F \ngtr G, L}[8] \quad \frac{\Psi, F ; \Delta \Uparrow L}{\Psi ; \Delta \Uparrow ? F, L}[?] \\
\frac{\Psi ; \Delta \Uparrow T, L}{}[\top] \frac{\Psi ; \Delta \Uparrow F, L \quad \Psi ; \Delta \Uparrow G, L}{\Psi ; \Delta \Uparrow F \& G, L}[\&] \frac{\Psi ; \Delta \Uparrow B[y / x], L}{\Psi ; \Delta \Uparrow \forall x \cdot B, L}[\forall]
\end{gathered}
$$

Positive rules

$$
\begin{gathered}
\frac{\Psi ; \Delta_{1} \Downarrow F \quad \Psi ; \Delta_{2} \Downarrow G}{\Psi ; \Downarrow 1}[1] \quad \frac{\Psi ; \Delta_{1}, \Delta_{2} \Downarrow F \otimes G}{\Psi ; \cdot]} \frac{\Psi ; \Uparrow F}{\Psi ; \cdot \Downarrow!F}[!] \\
\frac{\Psi ; \Delta \Downarrow F_{1}}{\Psi ; \Delta \Downarrow F_{1} \oplus F_{2}}\left[\oplus_{l}\right] \quad \frac{\Psi ; \Delta \Downarrow F_{2}}{\Psi ; \Delta \Downarrow F_{1} \oplus F_{2}}\left[\oplus_{r}\right] \frac{\Psi ; \Delta \Downarrow B[t / x]}{\Psi ; \Delta \Downarrow \exists x \cdot B}[\exists]
\end{gathered}
$$

$\psi$ a set, $\Delta$ a multiset, $L$ a list. There are one-sided sequents!

Identity, Decide, and Reaction rules

$$
\begin{array}{cl}
\overline{\Psi ; A \Downarrow A^{\perp}}\left[I_{1}\right] & \overline{\Psi, A ; \cdot \Downarrow A^{\perp}}\left[I_{2}\right] \\
\frac{\Psi ; \Delta \Downarrow F}{\Psi ; \Delta, F \Uparrow \cdot}\left[D_{1}\right] & \frac{\Psi, F ; \Delta \Downarrow F}{\Psi, F ; \Delta \Uparrow \cdot}\left[D_{2}\right]
\end{array}
$$

$\ln \left[I_{1}\right]$ and $\left[I_{2}\right], A$ is atomic. $\ln \left[D_{1}\right]$ and $\left[D_{2}\right], F$ is positive.

$$
\begin{array}{ll}
\frac{\Psi ; \Delta, F \Uparrow L}{\Psi ; \Delta \Uparrow F, L}[R \Uparrow] & \text { provided that } F \text { is positive or an atom } \\
\frac{\Psi ; \Delta \Uparrow F}{\Psi ; \Delta \Downarrow F}[R \Downarrow] & \text { provided that } F \text { is negative }
\end{array}
$$

An atom $A$ is negative and $A^{\perp}$ is positive.

## An example of a "macro-rule"

Assume that $\Psi$ contains the formula $a^{\perp} \otimes b^{\perp} \otimes c$. A deviation that focuses on that formula must have the following shape.

This derivation is possible only if $\Delta_{1}$ is $\{a\}$ and if $\Delta_{2}$ is $\{b\}$. Thus, the corresponding "macro-rule" is

$$
\frac{\Psi ; c, \Delta_{3} \Uparrow}{\Psi ; a, b, \Delta_{3} \Uparrow}
$$

Thus, selecting this formula corresponds to the "action" of replacing two members of a multiset with one.

## Arithmetic: add fixed points

We shall add the fixed point operators $\mu$ and $\nu$ which will be de Morgan duals of each other ( $\mu$ is positive and $\nu$ is negative).

We shall only adopt the unfolding rules for them, so they will not be distinguishable.

For an approach to induction and co-induction, see David Baelde's recent papers on $\mu M A L L$.

The additional (micro) rules for LLF are the following.

$$
\frac{\Psi ; \Delta \Uparrow B(\nu B) \bar{t}, \Gamma}{\Psi ; \Delta \Uparrow \nu B \bar{t}, \Gamma} \quad \frac{\Psi ; \Delta \Downarrow B(\mu B) \bar{t}}{\Psi ; \Delta \Downarrow \mu B \bar{t}}
$$

## Arithmetic: add equality

Formulas of the form $t=s$ are positive while formulas of the form $t \neq s$ are negative.

$$
\frac{\Psi_{\sigma} ; \Theta \sigma \Uparrow \Gamma \sigma}{\Psi ; \Theta \Uparrow \Gamma, s \neq t} \dagger \quad \overline{\Psi ; \Theta \Uparrow \Gamma, s \neq t} \ddagger \quad \overline{\Psi ; \cdot \Downarrow t=t}
$$

The proviso $\dagger$ requires the terms $s$ and $t$ to be unifiable and for $\sigma$ to be their most general unifier. The proviso $\ddagger$ requires that the terms $s$ and $t$ are not unifiable.

Oddly:

$$
3 \neq 3 \circ-\circ \perp \quad 3 \neq 4 \circ-\circ \top \quad 3=3 \circ-\circ 1 \quad 3=4 \circ-\circ 0
$$

Seems that equality (as a logical connective) can denote an additive and a multiplicative unit.

## A couple of fixed points

Encode the natural numbers using zero 0 and successor s.

$$
\begin{aligned}
x \leq y \quad \triangleq \quad & {[x=0] \oplus } \\
& {\left[\exists x^{\prime} y^{\prime} .\left(x=s x^{\prime}\right) \otimes\left(y=s y^{\prime}\right) \otimes x^{\prime} \leq y^{\prime}\right] . }
\end{aligned}
$$

$$
\begin{aligned}
\text { plus } y w u \triangleq \quad & {[y=0 \otimes w=u] \oplus } \\
& {\left[\exists y^{\prime} u^{\prime} .\left(y=s y^{\prime}\right) \otimes\left(u=s u^{\prime}\right) \otimes \text { plus } y^{\prime} w u^{\prime}\right] . }
\end{aligned}
$$

Notice that the following are "macro-rules."

$$
\overline{\overline{\psi ; \cdot \Downarrow 5 \leq 8}}
$$

$\xlongequal{\Psi ; \Gamma, q(0,3) \Uparrow \cdot \Psi ; \Gamma, q(1,2) \Uparrow \cdot \Psi ; \Gamma, q(2,1) \Uparrow \cdot \Psi ; \Gamma, q(3,0) \Uparrow .}$
( $\ldots$ assuming that formulas of the form $q(\cdot, \cdot)$ are positive).

## Computing the minimum of a multiset

$$
\begin{gathered}
\left\{\exists x \exists y\left[I(x)^{\perp} \otimes I(y)^{\perp} \otimes(x \leq y) \otimes I(x)\right]\right. \\
\left.\exists x\left[I(x)^{\perp} \otimes!\min (x)\right]\right\},
\end{gathered}
$$

The macro rules determined by focusing on these two rules are:

$$
\frac{\Psi ; \Gamma, I(x) \Uparrow \cdot}{\overline{\Psi ; \Gamma, l(x), I(y) \Uparrow \cdot}} \text { provided } x \leq y \quad \xlongequal[\Psi ; \min (x) \Uparrow]{\Psi ; l(x) \Uparrow \cdot}
$$

The use of! in the second clause above is important.
The sequent

$$
\Psi ; \cdot \Uparrow \min (m) \multimap\left[I\left(m_{1}\right) \ngtr \cdots>I\left(m_{n}\right)\right]
$$

is provable if and only if $m$ is the minimum of the (non-empty) multiset $\left\{m_{1}, \ldots, m_{n}\right\}$. [[Also, $\Psi$ contains the min program, etc.]]

## There is only one multiset, and it is global!

Performing interesting computations with multiset will almost certainly require many of them. Linear logic (Girard 1987) only provides one and its global.

Are we stuck? Are we doomed to go the route of Logical Algorithms by McAllester and Ganzinger and introduce lots of non-logical features into "proof systems"?

Not necessarily: the exponentials of linear logic have a feature ("bug") that allows us to enrich our computing paradigm with multiple multisets.

## Outline

## Synthetic connectives and focused proofs in linear logic

The subexponentials of linear logic

## Subexponentials and explicit algorithm specification

## Connectives can be canonical

The pair of connectives $\otimes$ and $\ngtr$ are canonical since

$$
\vdash B^{\perp}>C^{\perp}, B \otimes C \quad \text { and } \quad \vdash B^{\perp} \ngtr C^{\perp}, B \otimes C
$$

follow from the inference rules:

$$
\begin{array}{ll}
\frac{\vdash \Gamma, B, C}{\vdash \Gamma, B \gtrdot C} & \frac{\vdash \Gamma_{1}, B \quad \vdash \Gamma_{2}, C}{\vdash \Gamma_{1}, \Gamma_{2}, B \otimes C} \\
\frac{\vdash \Gamma, B, C}{\vdash \Gamma, B \gtrdot C} & \frac{\vdash \Gamma_{1}, B}{\vdash \Gamma_{1}, \Gamma_{2}, B \otimes C}
\end{array}
$$

All connectives of linear logic are canonical except for the exponentials.

## The exponentials are not canonical

The promotion rule is

$$
\frac{\vdash ? \Gamma, B}{\vdash ? \Gamma,!B}
$$

To prove! $B \multimap$ ! $B$ we have

$$
\frac{\text { oops }}{\vdash ? B^{\perp},!B}
$$

Since the exponentials are not canonical, there are numerous ways to develop them. E.g., lite, elementary, soft linear logics. We pursue another avenue.

## Subexponentials

Danos, Joinet, and Schellinx ("The structure of exponentials: Uncovering the dynamics of linear logic proofs", Kurt Gödel Colloquium, 1993). (We introduce the term subexponential.)

A subexponential signature is a tuple $\langle I, \preceq, \mathcal{W}, \mathcal{C}\rangle$ where

- $I$ is a set of indexes (naming the subexponentials),
- $\preceq$ is a pre-order on $I$,
- $\mathcal{W}$ and $\mathcal{C}$ are both subsets of $I$, and
- $\mathcal{W}$ and $\mathcal{C}$ are upward closed wrt $\preceq$.

If $k \in I$ then $?^{k}$ and ! ${ }^{k}$ are two dual subexponentials.
If $k \in \mathcal{W}$ then $?^{k}$ admit weakening (on the right).
If $k \in \mathcal{C}$ then $?^{k}$ admit contraction (on the right).

## Proof rules for subexponentials

Assume that $x \in I, y \in \mathcal{C}$, and $z \in \mathcal{W}$.

$$
\frac{\vdash C, \Delta}{\vdash ?^{x} C, \Delta} D \frac{\vdash ?^{y} C, ?^{y} C, \Delta}{\vdash ?^{y} C, \Delta} C \frac{\vdash \Delta}{\vdash ?^{z} C, \Delta} W
$$

The promotion rule in particularly interesting:

$$
\frac{\vdash ?^{x_{1}} C_{1}, \ldots, ?^{x_{n}} C_{n}, C}{\vdash ?^{x_{1}} C_{1}, \ldots, ?^{x_{n}} C_{n},!^{a} C}!^{a}
$$

provided $a \preceq x_{i}$ for all $i=1, \ldots, n$.
Refer to this proof system as $\mathrm{SELL}_{\Sigma}$ (linear logic over the subexponential signature $\Sigma$ ).

Cut and initial elimination holds for $\operatorname{SELL}_{\Sigma}$.

## The promotion rule as a gate keeper

Notice that if $x \npreceq y$ then the promotion rule can be applied to the sequent

$$
\vdash ?^{x} C_{1}, \ldots, ?^{x} C_{n}, ?^{y} D_{1}, \ldots, ?^{y} D_{m},!^{y} C
$$

only if $n=0$ : the premise of that promotion rule would then be $\vdash ?^{y} D_{1}, \ldots, ?^{y} D_{m}, C$.

The promotion rule can then be seen as a guard that allows a certain proof-search reduction only when the collection $\left\{C_{1}, \ldots, C_{n}\right\}$ located at $x$ is empty.

The promotion rule will allow us to test some locations for emptiness: computation can then move to another phase.

Eg: select the minimum from one location, then process the graph edges placed in another location, etc.

## Why the name?

We use the term subexponential since the equivalence

$$
?(A \oplus B) \circ-\circ(? A \gg B)
$$

-which relates the exponential ?, the additive $\oplus$, and the multiplicative $>$-fails when ? does not admit contraction and/or weakening.

## Outline

## Synthetic connectives and focused proofs in linear logic

The subexponentials of linear logic

Subexponentials and explicit algorithm specification

## A couple of assumptions for the rest of this talk

Neither

- the "affine" subexponential $(i \in \mathcal{W}$ and $i \notin \mathcal{C})$ nor
- the "relevant" subexponential $(i \notin \mathcal{W}$ and $i \in \mathcal{C})$ will be used here.

Instead, we shall assume that

- $\mathcal{W}=\mathcal{C}$ (only bounded and unbounded subexponentials) and
- the pre-ordered set $\langle I, \preceq\rangle$ has a maximum element, written $\infty$, which is unbounded.


## Further restrictions: no sub-locations

A complemented subexponential signature on set $I$ is the signature

$$
\langle\{\infty\} \cup I \cup \hat{l}, \preceq,\{\infty\},\{\infty\}\rangle
$$

where

- $\hat{l}$ is a copy of the index set $I$ containing elements of the form $\hat{l}$ whenever $I \in I$,
- the order relation $\preceq$ does not relate any two members in I nor any two members of $\hat{l}$, and
- $\hat{l} \preceq k$ such that $I$ and $k$ are distinct members of $I$.

The promotion rule with the subexponential ! $\hat{\text { I }}$ succeeds only if the indexed context is empty at location l: all other locations need not be considered.

We shall not use ?': that is, no data will not be "stored" in complemented locations.

## The specification language BAG

$$
\begin{aligned}
& t \quad::=c \in \mathbb{C} \mid \text { var } \quad \text { tup }::=\left\langle t_{1}, \ldots, t_{n}\right\rangle(n \geq 0) \\
& \text { pat } \quad::=\text { tup } \mid \lambda \text { var.pat } \\
& \operatorname{cond}_{a}::=t_{1} \boldsymbol{\nabla} t_{2} \quad \text { cond }_{1}::=\text { is_empty } \text { loc }_{b} \\
& \text { cond }::=\text { cond }_{a} \mid \text { cond, } \\
& \text { prog ::= load tup loc prog } \mid \text { unload }_{i} \text { loc pat bprog } \\
& \text { while cond }{ }^{\text {a }} \text { ( } \lambda \text { K.prog) prog } \\
& \mathbf{l o o p}_{i} \operatorname{loc}_{b} \text { kprog prog } \mid \text { new loc } \lambda L \text {.prog } \\
& \text { if cond prog }|\operatorname{prog} \rrbracket p r o g| K \mid \text { end } \\
& \text { bprog }::=\operatorname{prog} \mid \lambda \text { var.bprog } \\
& \text { kprog }::=\lambda \text { K.prog } \mid \lambda \text { var.kprog } \\
& \text { lprog }::=\lambda K . p r o g|\lambda L . l p r o g| \lambda \text { var.lprog }
\end{aligned}
$$

## The definition of BAG primitives

Define the delays: $\delta^{-}(C)$ is $C \ngtr \perp$ and $\delta^{+}(C)$ is $C \otimes 1$.

$$
\begin{aligned}
& \text { load }\langle\bar{t}\rangle l \operatorname{prog} \quad \triangleq ?^{l} l(\bar{t}) \gamma \delta^{+}(\text {prog }) \\
& \text { unload }{ }^{1} \text { pat bprog } \triangleq I(\text { pat } \bar{v})^{\perp} \otimes\left[\delta^{-}(\text {bprog } \bar{v})\right] \\
& \mathbf{l o o p}_{i} 1 \mathrm{kp} \text { prog } \quad \triangleq\left[l(\bar{v})^{\perp} \otimes \delta^{-}\left((\mathrm{kp} \overline{\mathrm{v}})\left(\mathbf{l o o p}_{i} 1 \mathrm{kp} \text { prog }\right)\right)\right] \oplus!^{\hat{l}}(\text { prog }) \\
& \operatorname{prog}_{1} \square \operatorname{prog}_{2} \quad \triangleq \operatorname{prog}_{1} \oplus \operatorname{prog}_{2} \\
& \text { if (is_empty l) prog } \triangleq!^{\hat{1}}(\text { prog }) \\
& \text { if }\left(t_{1} \backslash t_{2}\right) \text { prog } \quad \triangleq t_{1} \nabla t_{2} \otimes \delta^{-}(\operatorname{prog}) \\
& \text { new loc lprog } \quad \triangleq \cap_{\mathrm{l}} \text { lprog } \\
& \text { end } \quad \triangleq \perp
\end{aligned}
$$

while $\left(t_{1} \nabla t_{2}\right)$ kprog prog $\triangleq$
$\left[\left(t_{1} \nabla t_{2}\right) \otimes \delta^{-}\left(\operatorname{kprog}\left(\right.\right.\right.$ while $\left.\left.\left.\left(t_{1} \nabla t_{2}\right) k p r o g ~ p r o g\right) ~\right)\right] \oplus\left[\left(t_{1} \widetilde{\nabla} t_{2}\right) \otimes \delta^{-}(\operatorname{prog})\right]$ Here, $\boldsymbol{\nabla}$ is a binary test (say, $=, \leq,>$ ) and $\widetilde{\boldsymbol{\nabla}}$ is its complement.

## To delay or not to delay?

The size of a macro rule is related to how difficult it will be to implement. Macro connectives can be broken by inserting delays. Notice that there are no delays written into the definition of the [] operator since we wish that the choice provided by that operator is merged with choices in the instructions it accumulates.
For example, the instructions

$$
\text { (if } \left.\left.(x \leq y) \operatorname{prog}_{1}\right) \rrbracket(\text { if (is_empty } I) \operatorname{prog}_{2}\right)
$$

are equated to the formula

$$
\left((x \leq y) \otimes \delta^{-}\left(\operatorname{prog}_{1}\right)\right) \oplus!^{\hat{\prime}} \operatorname{prog}_{2}
$$

This synthetic connective combines internally the test $x \leq y$ with the emptiness check of location $l$.

## Size of macro rules

$$
\exists x \exists y \cdot\left[I(x)^{\perp} \otimes \delta^{+}\left(I(y)^{\perp} \otimes \delta^{+}((x \leq y) \otimes \text { cont })\right)\right]
$$

Represents three separate actions before you can pursue the continuation: read a value for $x$; read a value for $y$; and check that $x \leq y$. Easy to implement but one might need to "backtrack".
Better is the following:
$\exists x \exists y \cdot\left[I(x)^{\perp} \otimes \delta^{+}\left(I(y)^{\perp} \otimes \delta^{+}\left(\left((x \leq y) \otimes\right.\right.\right.\right.$ cont $\left._{1}\right) \oplus(x>y) \otimes$ cont $\left.\left.\left.\left._{2}\right)\right)\right)\right]$

$$
\exists x \exists y \cdot\left[I(x)^{\perp} \otimes I(y)^{\perp} \otimes(x \leq y) \otimes \text { cont }\right]
$$

Represents the single action of reading two numbers $x$ and $y$ such that $x \leq y$. Probably very hard to implement.

## Parallel and distributed computing?

Multifocusing for parallel execution. In the minimum example, one needs only $\log (n)$ sequential multifocused sets.

Distributed computing: the emptiness check can be limited to processes: it is no longer a global locking mechanism.

Is there an interesting bigraph here? The tree structure would be the collection of locations marked by their subexponentials. The other graph is links between multiset elements that share common constants (common names allow synchronization).

