Restricted linear dependent types for resource allocation

Fredrik Nordvall Forsberg
(joint work with Dan Ghica)

BLL Workshop
Fontainebleau, 4 December 2013
Resource analysis via types

- Common theme: type systems with resource annotations.
- Annotate terms with the amount of resources they need to execute successfully.
- Resources taken from a semiring.
- Type system generalisation of BLL.
Concrete example

When compiling to hardware circuits, we need to enforce a pipelining discipline for efficient results.

Here resources are “timing schedules”. 
Data dependent resource requirements

- Resource requirements are often data dependent.

- Simple example:

\[
\begin{align*}
\Gamma &\vdash x : \mathbb{N} \\
\Gamma &\vdash e_{\text{cheap}} : \theta \\
\Gamma &\vdash e_{\text{expensive}} : \theta \\
\Gamma &\vdash \text{ifzero } x \text{ then } e_{\text{cheap}} \text{ else } e_{\text{expensive}} : \theta
\end{align*}
\]
Data dependent resource requirements

- Resource requirements are often data dependent.

- Simple example:

\[
\Gamma \vdash x : \mathbb{N} \quad \Gamma \vdash e_{\text{cheap}} : \theta \quad \Gamma \vdash e_{\text{expensive}} : \theta
\]

\[
\Gamma \vdash \text{ifzero } x \text{ then } e_{\text{cheap}} \text{ else } e_{\text{expensive}} : \theta
\]
Data dependent resource requirements (cont.)

Possible if resources can depend on data:

\[ \text{ifzero } x \text{ then } e_{\text{cheap}} \text{ else } e_{\text{expensive}} \]
Possible if resources can depend on data:

- `e_{\text{expensive}}`
- `e_{\text{cheap}}`
- `\text{ifzero 0 then } e \text{ else } e'`
Data dependent resource requirements (cont.)

- Possible if resources can depend on data:

  \[\begin{align*}
  \text{ifzero } (n + 1) \text{ then } \ e & \text{ else } e' \\
  e_{\text{expensive}} & \\
  e_{\text{cheap}}
  \end{align*}\]
Data dependent resource requirements (cont.)

- Possible if resources can depend on data:

  $e_{\text{expensive}}$

  $e_{\text{cheap}}$

  ifzero $(n + 1)$ then $e$ else $e'$

- We want resource annotations to appear in types, and resources to depend on data.

- Hence, we need types to depend on data – dependent types.
A type system combining:

- The parametric resources from BLL over a general semiring (Ghica and Smith (2013), Brunel, Gaboardi, Mazza, and Zdancewic (2013)),
- the dependent linear types from dℓPCF (Dal Lago, Gaboardi, Petit (2011 – 2013)).

A categorical semantics for it.
A type system for resources
Design goals

- Type system with resource annotations and (restricted) dependent types.
- Call by name to fit in with hardware compiler.
- Decidable type inference (do not want to force programmer to deal with complicated types).
DML-style dependent types

Following dℓPCF, we use Dependent ML-style dependent types ("refinement types").

- Index language separate from term language.
- Also means we can safely add non-terminating or side effect-inducing terms to the term language.

- Linked through singleton types \(\text{nat}[i]\):

\[ x : \text{nat}[i] \iff x = i \]

- Here \(x\) is a term and \(i\) is an index term.
Overview

- Four different judgements:
  
  - $\Phi; \chi \vdash i \textbf{ index}$
  
  - $\Phi; \chi \vdash J \textbf{ resource}$
  
  - $\Phi; \chi \vdash \Theta \textbf{ type}$
  
  - $\Phi; \chi; \Gamma \vdash t : \Theta$

  $\Phi$ index context  $\chi$ index constraints  $\Gamma$ term context

- For ease of presentation, we first introduce an affine system (i.e. no contraction) without bounded exponentials.
Like DML, we are almost completely parametric in the index language.

E.g. $(\mathbb{N}, +, 0, 1, <)$ plus a variable schema.

The theory of the index language needs to be decidable for type checking.
Resources

- Resources should form an ordered monoid \((R, \times, 1, \leq)\).
- Also resources depending on index terms, e.g.:

\[
\Phi; \chi \vdash m, n \text{ index } \quad \Phi; \chi \vdash J, K \text{ resource }
\]

\[
\Phi; \chi \vdash (\text{if } m = n \text{ then } J \text{ else } K) \text{ resource }
\]

What happened to +?

“BLL over a semiring” is using + of the semiring for contraction. We will introduce a different bounded exponential for that purpose later.
Types (without exponentials)

- Generated by grammar

\[ \theta ::= \text{nat} \mid \text{nat}[M] \]
\[ \mid \theta \otimes \theta \]
\[ \mid (R \cdot \theta) \rightarrow \theta \]

(M index term, R resource).

- Formally derivations of the form \( \Phi; \chi \vdash \theta \) type.
Terms

Term contexts are sets $\Gamma = \{x_1 : R_1 \cdot \theta_1, \ldots, x_n : R_n \cdot \theta_n\}$.

- $\Phi, \chi \vdash \theta$ type
  $\frac{\Phi; \chi; x : 1 \cdot \theta \vdash x : \theta}{\Phi; \chi; x : 1 \cdot \theta \vdash x : \theta}$ (axiom)

- $\Gamma \vdash t : \theta'$
  $\frac{\Gamma \vdash t : \theta'}{\Gamma, x : R \cdot \theta \vdash t : \theta'}$ (weakening)

- $\Gamma, x : Q \cdot \theta \vdash t : \theta'$
  $\frac{Q \leq R}{\Gamma, x : R \cdot \theta \vdash t : \theta'}$ (resource weakening)

- $\Gamma \vdash s : \theta$
  $\frac{\Gamma' \vdash t : \theta'}{\Gamma, \Gamma' \vdash \langle s, t \rangle : \theta \otimes \theta'}$ (pairing)
Terms (cont.)

$$\vdash n : \text{nat}[n]$$

$$\Phi; \chi; \Gamma \vdash b : \text{nat}[B] \quad \Phi; \chi, B = 0; \Gamma' \vdash s : \theta \quad \Phi; \chi, B > 0; \Gamma' \vdash t : \theta$$

$$\Phi; \chi; \Gamma, \Gamma' \vdash \text{ifzero } b \text{ then } s \text{ else } t : \theta$$

$$\Gamma, x : J \cdot \theta \vdash t : \theta'$$

$$\Gamma \vdash \lambda x . t : J \cdot \theta \multimap \theta'$$

(abstraction)

$$\Gamma \vdash s : R \cdot \theta \multimap \theta' \quad \Gamma' \vdash t : \theta$$

(application)

$$\Gamma, R \cdot \Gamma' \vdash st : \theta'$$

where

$$R \cdot \{x_1 : S_1 \cdot \theta_1, \ldots, x_n : S_n \cdot \theta_n\} :=$$

$$\{x_1 : (R \times S_1) \cdot \theta_1, \ldots, x_n : (R \times S_n) \cdot \theta_n\}.$$
Example: short-circuit multiplication constant

\[ \ast : \text{nat} \to \text{nat} \to \text{nat} \]

- If the first argument is zero, no need to evaluate second argument.
- Simple notion of resource: number of threads needed.
- We can give \( \ast \) type

\[ \ast : 1 \cdot \text{nat}[A] \to R \cdot \text{nat} \to \text{nat} \]

where \( R \equiv \text{if } A = 0 \text{ then } 0 \text{ else } 1 \).
Example: a caching memory controller

- Caching behaviour of a memory controller: if we request the same address twice consecutively, the second roundtrip will be faster than requesting another address.

- A family of constants

  \[
  \text{mem}_i : R_i \cdot (\text{bool}[B_i] \otimes \text{nat}[A_i] \otimes \text{nat}) \rightarrow \text{nat}
  \]

- To write \( v \) at address \( a \), we call

  \[
  a \leftarrow_i v := \text{mem}_i(w, a, v)
  \]

  and ignore the result (here \( w := \text{fff} \)).

- To read the current value at address \( a \), we call

  \[
  !_i a := \text{mem}_i(r, a, \_)
  \]

  where \( \_ \) is arbitrary (here \( r := \text{tt} \)).
Example: a caching memory controller (cont.)

\[ \text{mem}_i : R_i \cdot (\text{bool}[B_i] \otimes \text{nat}[A_i] \otimes \text{nat}) \rightarrow \text{nat} \]

- The resource \( R_i \) depends on the data.
- For simplicity, let’s use “cost 1” or “cost 0”.

\[
R_0 \equiv 1 \\
R_{n+1} \equiv \text{if } B_{n+1} = r \text{ then (if } A_{n+1} = A_n \text{ then } 0 \text{ else } 1) \text{ else } 1
\]
Adding contraction

- "BLL over a semiring" use the additive structure of the semiring for contraction:

\[
\Gamma, x : R_1 \cdot \theta, y : R_2 \cdot \theta \vdash t : \theta' \\
\Gamma, x : (R_1 + R_2) \cdot \theta \vdash t\{x/y\} : \theta'
\]

- Does not work for us because of higher-order dependent types:

\[
\Gamma, f : R_1 \cdot (\text{nat}[A] \rightarrow \text{nat}), g : R_2 \cdot (\text{nat}[A] \rightarrow \text{nat}) \vdash f(0) + g(1) : \text{nat} \\
\Gamma, f : (R_1 + R_2) \cdot (\text{nat}[A] \rightarrow \text{nat}) \vdash f(0) + f(1) : \text{nat}
\]

would require \(0 = A = 1\).
Bounded exponentials

- Following dℓPCF, we use a bounded exponential $!_{i < N} \theta$.
- Here, $i$ is an index term, and $\theta$ can depend on $i$.
- Hence each argument can get its own index $A_i$:

$$ f : !_{i < 2} (R_i \cdot (\text{nat}[A_i] \rightarrow \text{nat})) $$

- In some sense a separate kind of resource (similar to the $\mathbb{N}_f<\mathcal{R}$ construction).
- Change contexts to be sets of the form

$$ \Gamma = \{x_1 : !_{i_1 < N_1} (R_1 \cdot \theta_1), \ldots, x_n : !_{i_n < N_n} (R_n \cdot \theta_n)\} $$
Modified types

- Types now given by the grammar

\[ \theta ::= \text{nat} \mid \text{nat}[N] \mid \theta \otimes \theta \mid (\!i<N (R \cdot \theta)) \rightarrow \theta \]

- Detailed rule for the function type:

\[ \Phi; \chi \vdash N \text{ index} \quad \Phi, i; \chi, i < N \vdash R \text{ resource} \quad \Phi; \chi \vdash \theta' \text{ type} \]

\[ \Phi; \chi \vdash (\!i<N R \cdot \theta) \rightarrow \theta' \text{ type} \]

- Note dependency of \( R \) and \( \theta \) on \( i < N \).

- Important point: \( R \) should only be allowed to depend on data \( A_i \) (as in \( \text{nat}[A_i] \)), not on the index \( i \) itself (does not make sense in call by name setting).
Contraction rule

\[
\frac{
\Phi; \chi; \Gamma, x : ! i < N \quad R \cdot \theta, \ y : ! i' < M ((R \cdot \theta)\{N + i'/i\}) \vdash t : \theta'}{
\Phi; \chi; \Gamma, x : ! i < N + M \quad R \cdot \theta \vdash t\{x/y\} : \theta'}
\] (contraction)

- Intuitively: split the \(N + M\) occurrences of \(x\) into the first \(N\), followed by the next \(M\).

\[
(R \cdot \theta)\{0\}, \ldots, (R \cdot \theta)\{N - 1\}, (R \cdot \theta)\{N\}, \ldots (R \cdot \theta)\{N + M - 1\}
\]

- \(\! i < N \quad R \cdot \theta\)

- \(\! i' < M ((R \cdot \theta)\{N + i'/i\})\)

- Other rules need to be adapted as well.

- In particular, resource action \(R \cdot \Gamma\) needs to be generalised to \(\! N \quad R \cdot \Gamma\) (following dℓPCF).
Modified rules

\[
\Phi, i, \chi, i < 1 \vdash \theta \quad \text{(type)}
\]

\[
\Phi; \chi; x : !i_{<1} 1 \cdot \theta \vdash x : \theta\{0/i\} \quad \text{(axiom)}
\]

\[
\Gamma, x : !i_{<N} Q \cdot \theta \vdash t : \theta' \quad Q \leq R
\]

\[
\Gamma, x : !i_{<N} R \cdot \theta \vdash t : \theta'
\]

\[
\Gamma, x : !i_{<N} (J \cdot \theta) \vdash t : \theta'
\]

\[
\Gamma \vdash \lambda x. t : !i_{<N} (J \cdot \theta) \rightarrow \theta'
\]

\[
\Phi; \chi; \Gamma \vdash f : !i_{<N} (R \cdot \theta) \rightarrow \theta'
\]

\[
\Phi, i; \chi, i < N; \Gamma' \vdash t : \theta
\]

\[
\Phi; \chi; \Gamma, ! R \cdot \Gamma' \vdash f t : \theta'
\]

\[
\text{(application)}
\]
Categorical semantics
Overview

- Category $\mathcal{B}$ of index contexts.

- Indexed category $\mathcal{I} : \mathcal{B}^{\text{op}} \to \text{Cat}$ of index terms.

- Indexed category $\mathcal{R} : \mathcal{B}^{\text{op}} \to \text{Cat}$ of resources.

- Indexed category $\mathcal{C} : \mathcal{B}^{\text{op}} \to \text{Cat}$ of computations.
Index contexts and index terms

- Index contexts $\Phi; \chi$ can be interpreted as the conjunction of the constraints in $\chi$.

- Then interpreted in the standard categorical logic way in any Cartesian (well-powered) category $\mathcal{B}$.

- There is a morphism $\llbracket \Phi, \chi \rrbracket \to \llbracket \Phi', \chi' \rrbracket$ if $\chi$ implies $\chi'$ in the model.

- Any indexed category $\mathcal{I} : \mathcal{B}^{\text{op}} \to \text{Cat}$ supporting the constants can now interpret the index terms.
Similarly, resources can be interpreted in an indexed category $\mathcal{R} : \mathcal{B}^{\text{op}} \to \text{Cat}$ supporting the right constants.

In particular, $\mathcal{R}$ should be “indexed monoidal”, i.e. each fibre $\mathcal{R}(\Phi, \chi)$ should have a monoidal tensor $(\odot, 1)_{\Phi, \chi}$ (compatible with index substitution).

Morphisms $[R] \to [S]$ corresponds to inequalities $R \leq S$. 
Types and terms

- Indexed symmetric monoidal closed category $\mathcal{C} : \mathcal{B}^{\text{op}} \to \text{Cat}$ (i.e. each fibre $\mathcal{C}(\Phi, \chi)$ symmetric monoidal closed).

- Furthermore the unit $l_{\Phi, \chi}$ should be terminal in each fibre. This implies that we have projections $\pi : A \otimes B \to A$.

- Types $\Phi; \chi \vdash \theta \text{ type}$ are interpreted as objects $[\theta]$ in $\mathcal{C}(\Phi; \chi)$.

- Terms $\Phi; \chi; \Gamma \vdash M : \theta$ are interpreted as morphisms $[M] : [\Gamma] \to [\theta]$ in $\mathcal{C}(\Phi; \chi)$ where

\[
[\Gamma] = [x_1 : !_{a_1 < N_1} J_1 \cdot \theta_1, \ldots, x_n : !_{a_n < N_n} J_n \cdot \theta_n]
\]

\[:= !_{N_1 J_1 \cdot [\theta_1] \otimes \ldots \otimes !_{N_n J_n \cdot [\theta_n]}}\]

with $!_{N J \cdot X}$ explained soon.
We further require a natural transformation $\cdot : \mathcal{R} \times \mathcal{C} \to \mathcal{C}$, i.e.

$$\cdot_{\Phi;\chi} : \mathcal{R}(\Phi;\chi) \times \mathcal{C}(\Phi;\chi) \to \mathcal{C}(\Phi;\chi)$$

Satisfying the following equations (fibrewise):

1. $1 \cdot_{\Phi;\chi} \theta = \theta$
2. $(R \circ S) \cdot_{\Phi;\chi} \theta = R \cdot_{\Phi;\chi} (S \cdot_{\Phi;\chi} \theta)$.
Bounded exponential

The bounded exponential $!_{i<N}$ should be a functor

$$!_{i<N}: \mathcal{C}(\Phi, i; \chi, i < N) \rightarrow \mathcal{C}(\Phi; \chi)$$

Which one? Kind of a bounded universal quantification...
The bounded exponential $!_{i < N}$ should be a functor

$$!_{i < N} : \mathbb{C}(\Phi, i; \chi, i < N) \to \mathbb{C}(\Phi; \chi)$$

Which one? Kind of a bounded universal quantification...

Right adjoint to weakening!

We have a “projection” $\pi_{\Phi, \chi, i, i < N} : [\Phi, i; \chi, i < N] \to [\Phi; \chi]$ in $B$.

Gives rise to $\text{wk}_{i < N} := \mathbb{C}(\pi_{\Phi, \chi, i, i < N}) : \mathbb{C}(\Phi; \chi) \to \mathbb{C}(\Phi, i; \chi, i < N)$.

$\text{wk}_{i < N} \dashv !_N$.

$!_N$ also needs to be monoidal.
For each index term \( N \), the natural transformation \( \!
abla_N \cdot \! \) with components

\[
(\!
abla_N \cdot \!)_{\Phi;\chi} : \mathcal{R}(\Phi, i; \chi, i < N) \times \mathcal{C}(\Phi, i; \chi, i < N) \to \mathcal{C}(\Phi; \chi)
\]

can be defined by composing \( \!
abla_{i < N} \) and \( \cdot \).

Also need a family of natural transformations \( \delta_{N,M} \) corresponding to contraction

\[
\delta_{N,M,X} : \!
abla_{N+M}(X) \to \!
abla_N (X) \otimes \!
abla_M (X\{i/N + j\}) \]

and dereliction

\[
\iota_X : \!
abla_1 (X) \to X
\]

(natural as usual).
The interpretation

\[
[[\Gamma \vdash t : \theta]] : [[\Gamma]] \to [[\theta]]
\]

\[
[[x : !_{i<1} 1 \cdot \theta \vdash x : \theta\{0/i\}]] = \nu \circ \epsilon\{0/i\}
\]

\[
[[\Gamma, x : !_{i<\mathbb{N}} R \cdot \theta \vdash t : \theta']] = [[\Gamma \vdash t : \theta']] \circ \pi
\]

\[
[[x : !_{i<\mathbb{N}} R \cdot \theta \vdash t : \theta']] = [[\Gamma, x : !_{i<\mathbb{N}} Q \cdot \theta \vdash t : \theta']] \circ (\text{id}_\Gamma \otimes !_{\text{id}_\mathbb{N}} [[Q \leq R] \cdot \text{id}_\theta])
\]

\[
[[\Gamma, \Gamma' \vdash \langle t, t' \rangle : \theta \otimes \theta']] = [[\Gamma \vdash t : \theta]] \otimes [[\Gamma' \vdash t' : \theta']]
\]

\[
[[\Gamma \vdash \lambda x. t : !_{i<\mathbb{N}} R \cdot \theta \to \theta']] = \Lambda([[\Gamma, x : !_{i<\mathbb{N}} R \cdot \theta \vdash t : \theta']])
\]

\[
[[\Gamma, !_{\mathbb{N}} R \cdot \Gamma' \vdash f \ t : \theta']] = \text{eval} \circ [[\Gamma \vdash f : !_{i<\mathbb{N}} R \cdot \theta \to \theta']] \otimes !_{\mathbb{N}} R \cdot [[\Gamma' \vdash t : \theta]]
\]

\[
[[\Gamma, x : !_{i<\mathbb{N}+\mathbb{M}} R \cdot \theta \vdash t\{x/y\} : \theta']] = [[\Gamma, x : \ldots, y : \ldots \vdash t : \theta']] \circ (\text{id}_\Gamma \otimes \delta_{\mathbb{N}, \mathbb{M}, R \cdot \theta})
\]
Coherence

- Just like for the non-dependent system, the use of weakening and contraction introduces multiple derivations.

- Coherence can be proven in the same way:
  1. Transform derivations into a normal form, with weakening and contractions performed as late as possible.
  2. Show that the transformation is meaning-preserving.
  3. Show that derivations in normal form are unique.

- Crucial is the fact that resource dependency is uniform: it is possible to depend on data $A_i$, but not on the index $i$ itself.
Combining BLL over a semiring and restricted dependent linear types.

Coherent categorical semantics by using indexed categories.

Thank you!
The action $!_N R \cdot \Gamma'$

- Simplified:

$$!_N R \cdot (x : !_{j<M} (S \cdot \theta)) = x : !_{j<N \cdot M} (R \times S \cdot \theta)$$

- Need $N$ copies of the context, and each copy needs $M$ occurrences of $x$, so need $N \cdot M$ copies.

- For each copy, multiply resource requirements.

- But this does not take index dependency into account; e.g. $M$ can depend on $i < N$.

- So use $\sum_{i<N} M$ instead of $N \cdot M$. 
More dependencies

- Simplified:

\[
!_N R \cdot (x : !_{j<M} (S \cdot \theta)) = x : !_{k<\sum_{i<N} M} (R \times S \cdot \theta)
\]

- Similarly, dependencies of \( R \) on \( i < N \) and of \( S \) on \( j < M \).

- Each \( k < \sum_{i<N} M \) can be written as \( k = \sum_{d<i} M\{d/i\} + j \) with \( i < N \) and \( j < M \).

\[
M(0) + M(1) + M(2) + M(3) + \ldots + M(N-1)
\]

\[
\sum_{d<i} M(d) \quad j
\]

- Multiply \( R(i) \) and \( S(j) \) for these \( i \) and \( j \).
Formally

Given

\[ \Phi; \chi \vdash N \text{ index} \]
\[ \Phi, i; \chi, i < N \vdash R \text{ resource} \]
\[ \Phi, i; \chi, i < N \vdash M \text{ index} \]
\[ \Phi, i, j; \chi, i < N, j < M \vdash K \text{ resource} \]

and

\[ A = \diamond_{j < M} (K \cdot \theta\{\left(\sum_{d < i} M\{d/i\} + j\}/k\}) \]

we define

\[ \diamond N R \cdot A = \diamond_{k < \sum_{i < N} M} (R \times K) \cdot \theta \]