Checking Dependently Typed Programs

James M. Chapman
University of Nottingham
Epigram is a dependently typed functional programming language. It has:

- \( \Pi \)-types
- \( \Sigma \)-types
- Inductive Families
- Elaborates to UTT
Epigram programs are:

- total
- can be correct by construction
- are also proofs (Curry-Howard)
How can we believe a program is right?

- Is the program correct?
- Is it the right program?

How can we trust evidence?

- Proof of the Four Colour theorem is 60,000 lines long

Idea: Rather than read the proof write a checker

Paranoia
Checking Epigram Programs

Equipment:
- $\beta\eta$-equality test
- $\beta$-normalization
- An implementation of the typing rules
If $S$, $T$, $s$ and $t$ are terms then so are the following:

- $\star$, Unit, () Constants
- $x$ Variables (e.g. $\Gamma;x:T$)
- $\lambda x:S \Rightarrow t$ Lambda Abstraction
- $t \ s$ Function Application
- $\Pi x:S \Rightarrow T$ Dependent function space
βη-equality testing

Start with a theory with β-equality and add η-reduction:

\[ \lambda x: S \Rightarrow fx \text{ is reduced to } f \]

Does not work for Σ-types

An Algorithm for testing conversion (Coquand, 1991)
Untyped $\beta\eta$-equality

\[
\begin{align*}
\Gamma; x: S \vdash t[x] & \Downarrow v \\
\Gamma; x: S \vdash t'[x] & \Downarrow v'
\end{align*}
\]

\[
\Gamma; x: S \vdash v =_{\beta\eta} v'
\]

\[
\Gamma \vdash \lambda x: S \Rightarrow t =_{\beta\eta} \lambda x: S' \Rightarrow t'
\]

\[
\Gamma; x: S; \Gamma' \vdash x =_{\beta\eta} x
\]

\[
\begin{align*}
\Gamma; x: S \vdash t[x] & \Downarrow v \\
\Gamma; x: S \vdash v =_{\beta\eta} n \; x
\end{align*}
\]

\[
\Gamma \vdash \lambda x: S \Rightarrow t =_{\beta\eta} n
\]

\[
\begin{align*}
\Gamma; x: S \vdash n \; x & =_{\beta\eta} w \\
\Gamma \vdash n & =_{\beta\eta} \lambda x: S \Rightarrow t
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash n & =_{\beta\eta} n' \\
\Gamma \vdash a & =_{\beta\eta} a'
\end{align*}
\]

\[
\Gamma \vdash n \; a =_{\beta\eta} n' \; a'
\]
The untyped algorithm doesn’t work for elements of Unit.

We want:

\[ \frac{}{\Gamma \vdash \nu =_\beta \eta w : \text{Unit}} \]
Type directed $\beta\eta$-equality

\[
\begin{align*}
\Gamma \vdash fx \downarrow v \\
\Gamma \vdash gx \downarrow v' \\
\Gamma; x: S \vdash \nu \equiv_{\beta\eta} \nu' : V
\end{align*}
\]

\[
\Gamma \vdash f \equiv_{\beta\eta} g : \Pi x: S \Rightarrow V
\]
Further work

- Extend type checker to inductive families.
- Prove correctness of type-directed equality.