A Higher-Order Logical Framework for Reasoning about Programming Languages in Coq

Chelsea Battell

Department of Mathematics and Statistics
University of Ottawa

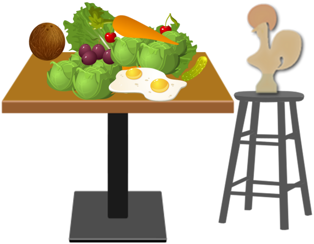
February 12, 2016
McGill University, Montreal, Canada

OBJECTIVE

Mechanize reasoning about programming languages and logics

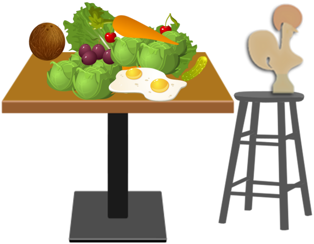
EXAMPLE

For STLC, want to prove $\dfrac{\vdash e \Downarrow v \quad \vdash e : t}{\vdash v : t}$



SOLUTION

Encode the object logic in an existing proof assistant

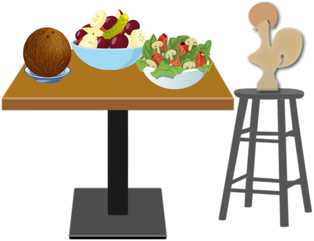


SOLUTION

Encode the object logic in an existing proof assistant

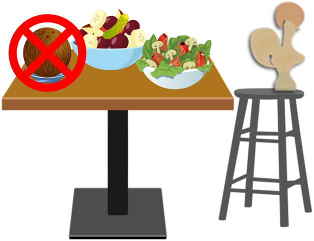
PROBLEM

Many tedious computations for each encoding with binding structures



SOLUTION

Use higher-order abstract syntax

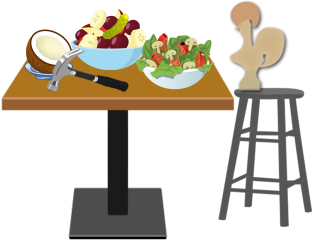


SOLUTION

Use higher-order abstract syntax

PROBLEM

Some judgments cannot be encoded as inductive types in Coq



SOLUTION

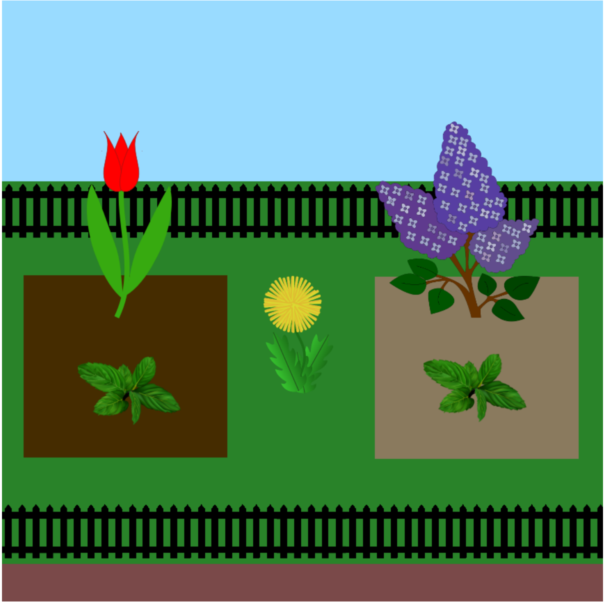
Add intermediate layer called specification logic with parameter for provability in object logic

Object Logic

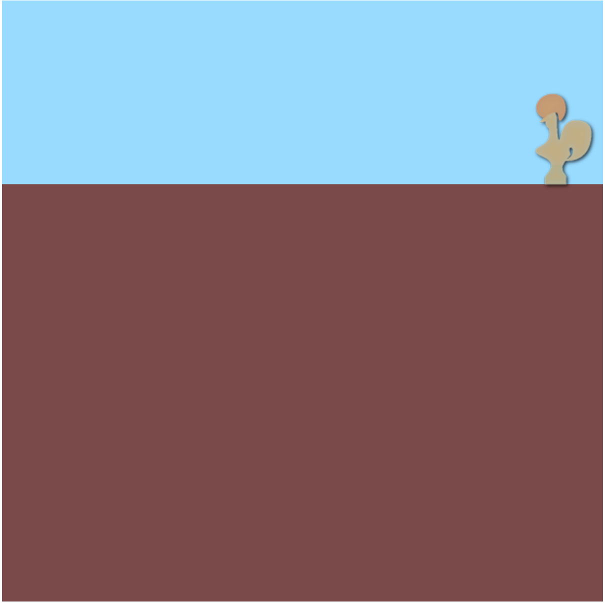
Specification Logic(s)

Higher-Order Abstract Syntax

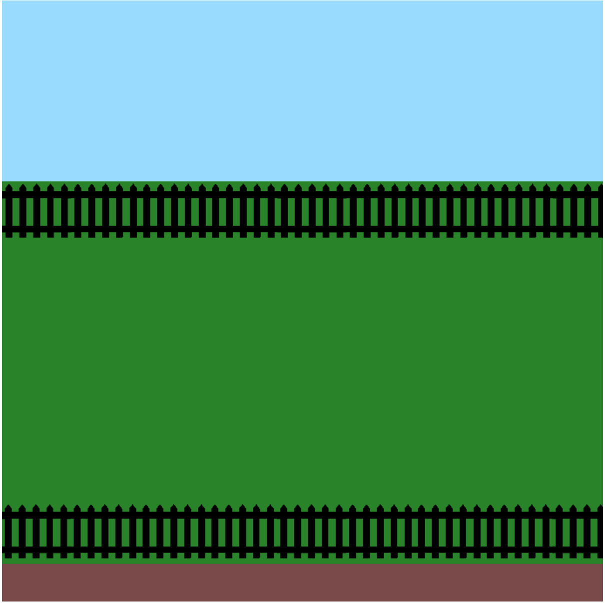
Ambient Logic



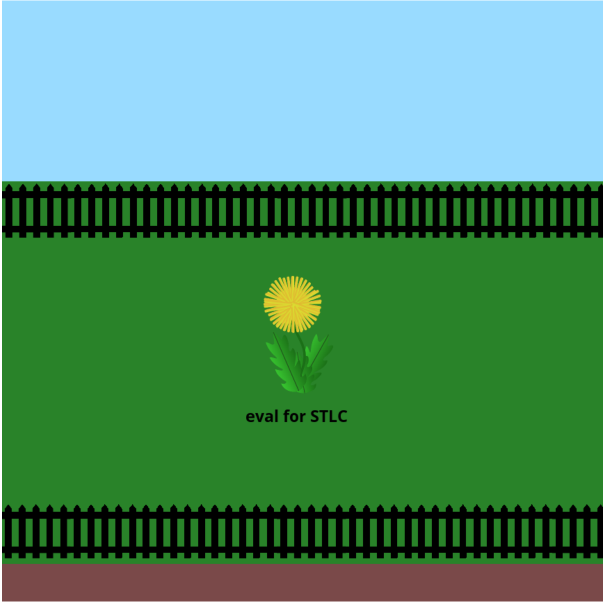
Ambient
Logic



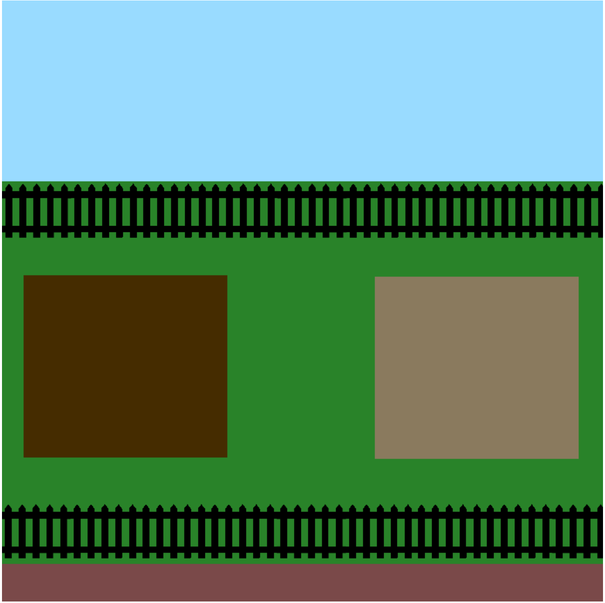
Higher-Order Abstract Syntax



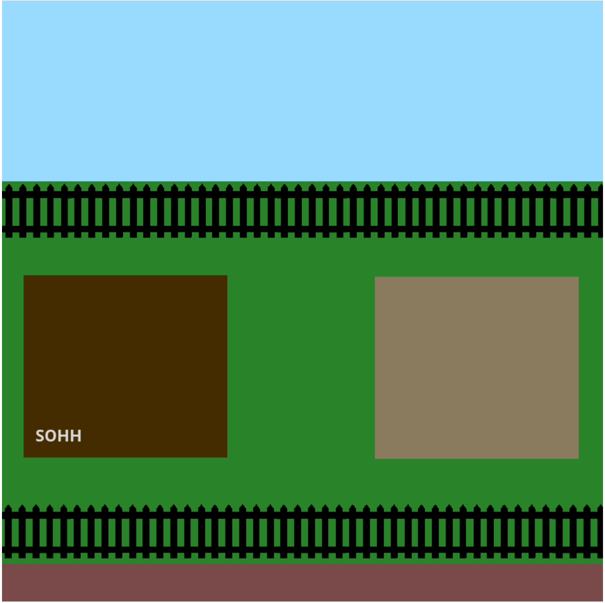
Higher-Order Abstract Syntax



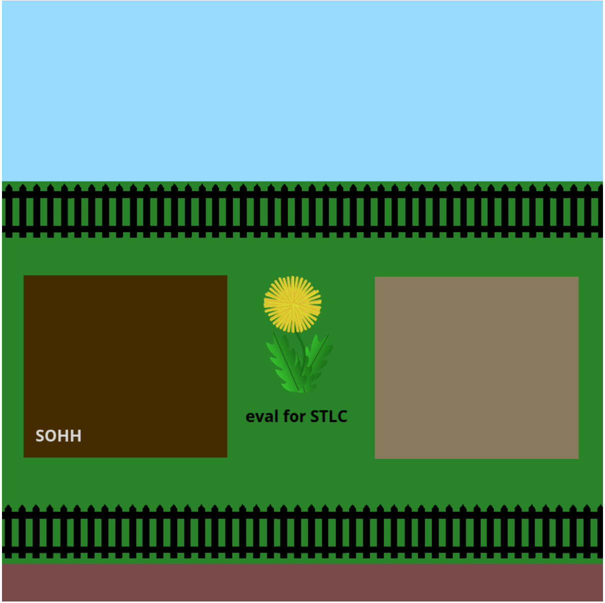
Specification
Logic(s)



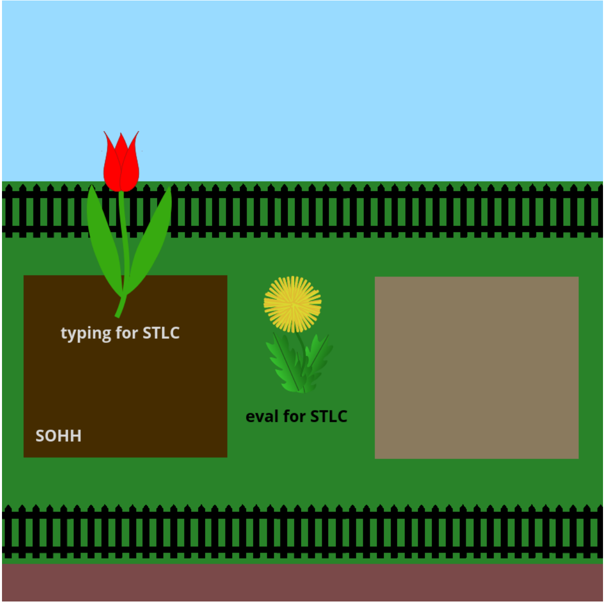
Specification
Logic(s)



Specification Logic(s)



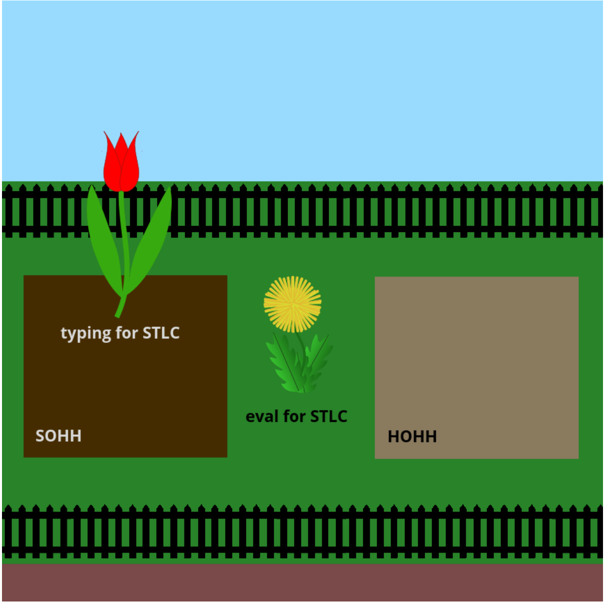
Specification Logic(s)



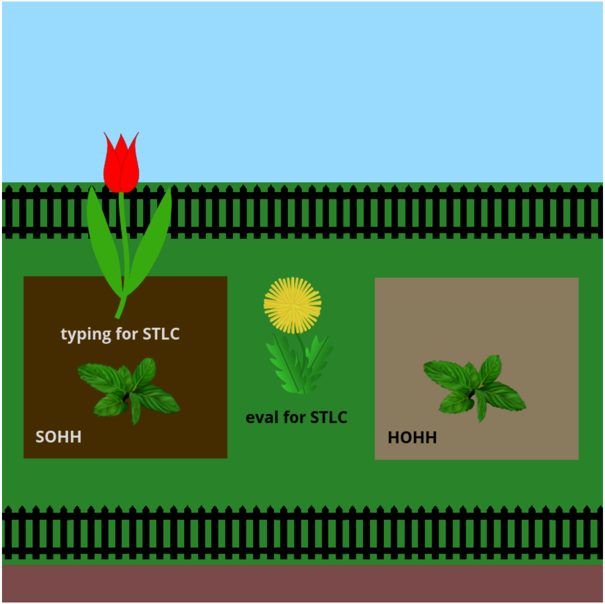
Subject Reduction

$$\frac{e \Downarrow v \qquad \rhd\langle\, e : t \,\rangle}{\rhd\langle\, v : t \,\rangle}$$

Specification Logic(s)



Specification Logic(s)



$$\frac{A \mathrel{:-} G \quad \Gamma \rhd G}{\Gamma \rhd \langle\, A \,\rangle}\ \text{s\_bc}$$

$$\frac{F \in \Gamma \quad \Gamma, [F] \rhd A}{\Gamma \rhd \langle\, A \,\rangle}\ \text{s\_init}$$

$$\vdots$$

$$\frac{\Gamma, G_1 \rhd G_2}{\Gamma \rhd G_1 \to G_2}\ \text{s\_imp}$$

$$\frac{\texttt{proper}\, x \to \Gamma \rhd E\, x}{\Gamma \rhd \forall^{expr} E}\ \text{s\_all}$$

$$\vdots$$

$$\frac{}{\Gamma, [\langle\ A\ \rangle] \rhd A}\ \text{b\_match}$$

$$\frac{\Gamma \rhd G \quad \Gamma, [F] \rhd A}{\Gamma, [G \to F] \rhd A}\ \text{b\_imp}$$

$$\vdots$$

```
Inductive seq : context -> oo -> Prop :=
...
| s_imp :
 forall (G1 G2 : oo) (L : context),
 seq (L, G1) G2
  -> seq L (G1 ---> G2)
| s_all :
 forall (E : expr -> oo) (L : context),
 (forall x : expr, proper x -> seq L (E x))
  -> seq L (All E)
...
with bch : context -> oo ->  atm -> Prop :=
| b_match :
 forall (A : atm) (L : context),
  bch L (< A >) A
| b_imp :
 forall (F G : oo) (A : atm) (L : context),
 seq L G -> bch L F A
  -> bch L (G ---> F) A.
...
```

STRUCTURAL RULES

Weakening, Contraction and Exchange corollaries of:

$$\frac{\Gamma_1 \subseteq \Gamma_2 \qquad \boxed{\Gamma_1 \rhd o}}{\Gamma_2 \rhd o} \wedge \frac{\Gamma_1 \subseteq \Gamma_2 \qquad \boxed{\Gamma_1, [o] \rhd a}}{\Gamma_2, [o] \rhd a}$$

PROOF

By mutual structural induction over sequent premises

$P_1 := \lambda \Gamma_2 \,.\, \lambda o \,.\, \Gamma_1 \subseteq \Gamma_2 \to \Gamma_2 \rhd o$
$P_2 := \lambda \Gamma_2 \,.\, \lambda o \,.\, \lambda a \,.\, \Gamma_1 \subseteq \Gamma_2 \to \Gamma_2, [o] \rhd a$

Subcase $\dfrac{\Gamma \rhd G \quad \Gamma, [F] \rhd A}{\boxed{\Gamma, [G \to F] \rhd A}}$ b_imp :

Then $\Gamma_1 = \Gamma, o = G \to F$ and $a = A$.

$$\frac{\begin{array}{c} H_1 : \Gamma \rhd G \\ IH_1 : P_1\ \Gamma\ G \\ H_2 : \Gamma, [F] \rhd A \\ IH_2 : P_2\ \Gamma\ F\ A \end{array}}{P_2\ \Gamma\ (G \to F)\ A}$$

CUT RULE

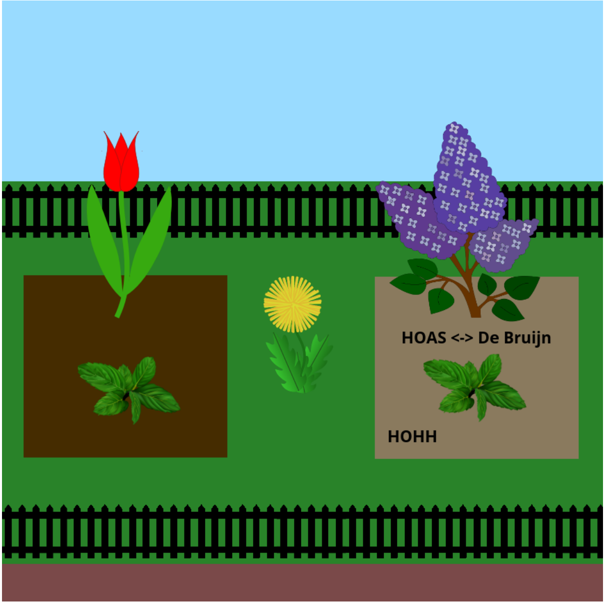
$$\frac{\boxed{\Gamma, \mathbf{o_1} \rhd o_2} \quad \Gamma \rhd \mathbf{o_1}}{\Gamma \rhd o_2} \wedge \frac{\boxed{\Gamma, \mathbf{o_1}, [o_2] \rhd a} \quad \Gamma \rhd \mathbf{o_1}}{\Gamma, [o_2] \rhd a}$$

structural induction over $\mathbf{o_1}$

mututal structural induction over $\Gamma, \mathbf{o_1} \rhd o_2$ and $\Gamma, \mathbf{o_1}, [o_2] \rhd a$

98 subcases (91 proven automatically)

Object
Logic



$$
\begin{array}{lcl}
\texttt{hApp} & : & \texttt{tm} \to \texttt{tm} \to \texttt{tm} \\
\texttt{hAbs} & : & (\texttt{tm} \to \texttt{tm}) \to \texttt{tm} \\
\texttt{dApp} & : & \texttt{dtm} \to \texttt{dtm} \to \texttt{dtm} \\
\texttt{dAbs} & : & \texttt{dtm} \to \texttt{dtm} \\
\texttt{dVar} & : & \mathbb{N} \to \texttt{dtm}
\end{array}
$$

$$
\frac{\Gamma \vdash H_1 \equiv_n D_1 \quad \Gamma \vdash H_2 \equiv_n D_2}{\Gamma \vdash \texttt{hApp}\ H_1\ H_2 \equiv_n \texttt{dApp}\ D_1\ D_2}\ \text{hodb\_app}
$$

$$
\frac{\Gamma, (\forall k, x \equiv_{n+k} \texttt{dVar}\ k) \vdash H \equiv_{n+1} D}{\Gamma \vdash \texttt{hAbs}\ (\lambda x.H) \equiv_n \texttt{dAbs}\ D}\ \text{hodb\_abs}
$$

[Wang, Chaudhuri, Gacek, Nadathur; PPDP-13]

DEMO

Prove $\lambda x.x \equiv_0 \lambda.1$ using Hybrid

TODO

$$\frac{\triangleright\langle\ \mathtt{hodb}\ H_1\ n\ D\ \rangle \quad \triangleright\langle\ \mathtt{hodb}\ H_2\ n\ D\ \rangle}{H_1 = H_2}$$

and

$$\frac{\triangleright\langle\ \mathtt{hodb}\ H\ n\ D_1\ \rangle \quad \triangleright\langle\ \mathtt{hodb}\ H\ n\ D_2\ \rangle}{D_1 = D_2}$$

Thank you!

Questions? Comments?