POPLmark 1a with Named Bound Variables

Aaron Stump

Computer Science and Engineering
Washington University
St. Louis, Missouri, USA

Why Named Bound Variables?

1. Closer to the real language:
    - Languages are presented with named variables.
    - Tools support named variables.
    - There is a gap if meta-theory done with de Bruijn indices.
2. De Bruijn indices considered tedious.

Why Not Named Bound Variables?

Named bound variables considered **more** tedious.

- Capture-avoiding substitution requires (**safe**) renaming:

  $$[N/X](\forall Y <: T_1.T_2) = \forall Z <: [N/X]T_1.[N/X][Z/X]T_2$$

  if $X \neq Y$ and $Z \notin FV(T2) \cup FV(N)$
- Some complexity to phrase this primitive recursively.
  - $[Z/X]T_2$ is not a subterm of $\forall Y <: T_1.T_2$.
  - Coq syntactically enforces structural decrease.

Techniques to Tame Named Bound Variables

1. Barendregt variable convention.
   - Free and bound variables drawn from disjoint sets.
   - So free variables cannot possibly be captured when substituting under a binder.
   - Capture-avoiding substitution becomes grafting.
2. De Bruijn levels for free variables (aka *constants*).

$$\frac{\Gamma \vdash \mathsf{T}_1 \mathrel{<:} \mathsf{S}_1 \qquad \Gamma, X <: \mathsf{S}_1 \vdash [\mathsf{X}/\mathsf{X}_1]\mathsf{S}_2 <: [\mathsf{X}/\mathsf{X}_2]\mathsf{T}_2}{\Gamma \vdash \forall \mathsf{X}_1 \mathrel{<:} \mathsf{S}_1.\mathsf{S}_2 \mathrel{<:} \forall \mathsf{X}_2 \mathrel{<:} \mathsf{T}_1.\mathsf{T}_2} \ \textsc{SA-All}$$

Question: How to choose X?
Answer: Canonically.

- Constants are nats, introduced in order.
- No need for a renaming lemma.
- Indeed, next constant is just $|\Gamma|$.