Explicit substitution.

Note: using deBruijn indices to avoid names.

1. Syntax.

$$e ::= 0 \mid e_1\ e_2 \mid \lambda e \mid e[\sigma]$$
$$\sigma ::= id \mid \uparrow \mid \sigma, e \mid \sigma \circ \tau$$

Note: the n^{th} deBruijn index can be represented by $0[\uparrow^n]$.

- Substitutions. ~~ate~~
  - id: the identity. $e[id] = e$. $i \longmapsto i$.
  - $\uparrow$: the weakening. $i \longmapsto i+1$.
  - $\_,\_$: the extension. $(\sigma, e)(i) = \begin{cases} e & \text{if } i = 0 \\ \sigma(i) & \text{if } i = i+1 \end{cases}$
  - $\_\circ\_$: the composition. $e[\sigma \circ \tau] = (e[\sigma])[\tau]$.

    Hence, $(\sigma \circ \tau)(i) = \sigma(i)[\tau]$

- Calculus.

$$\uparrow, 0 = id. \qquad \uparrow \circ (\sigma, e) = \sigma. \qquad (\sigma, e) \circ t = \sigma \circ t, e[t].$$
$$(\sigma \circ \tau) \circ \rho = \sigma \circ (\tau \circ \rho). \qquad 0[\sigma, t] = t. \qquad id \circ \sigma = \sigma = \sigma \circ id.$$

Note: $\uparrow \circ \sigma$ is like "taking the tail of σ". 0 is the head of σ.

$\_\circ\_$ is associative, and it distributes over $\_,\_$.

Hence, we have: $\underbrace{\uparrow \circ \sigma}_{\text{tail}} \underbrace{,}_{\text{assemble}} \underbrace{0[\sigma]}_{\text{head}} = \underbrace{\sigma}_{\text{the original}}$.

2. Evaluate substitutions.

- $0[\sigma, e] = e$.
- $(e[\sigma])[\tau] = e[\sigma \circ \tau]$.
- $(e_1\ e_2)[\sigma] = e_1[\sigma]\ e_2[\sigma]$.
- ※ $(\lambda e)[\sigma] = \lambda(e[\sigma \circ \uparrow, 0])$.

We have to weaken the substitution, since we're now one-abstraction deeper.

We must avoid substituting the 0^{th} index, since it is bounded.

β-reduction:

$$(\lambda e_1)\, e_2 = e_1[id, e_2].$$

3. Simple types.

Note: of course, we have $\overline{\Gamma, A \vdash 0 : A}$.

※ Judgement: $\boxed{\Gamma \vdash \sigma \rhd \Gamma'}$

Meaning: σ is a list of terms for substitutions.
Γ' is a list of types.
$\Gamma \vdash \sigma \rhd \Gamma'$ means, ~~that~~ the list of terms in σ has ~~that~~ the list of types in Γ'.

$$\frac{}{\Gamma \vdash id \rhd \Gamma}\ (id) \qquad \frac{\Gamma \vdash e : A \quad \Gamma \vdash \sigma \rhd \Gamma'}{\Gamma \vdash \sigma, e \rhd \Gamma', A}\ (-,-)$$

$$\frac{}{\Gamma, A \vdash \uparrow \rhd \Gamma}\ (\uparrow) \qquad \frac{\Gamma \vdash \sigma \rhd \Gamma' \quad \Gamma'' \vdash \tau \rhd \Gamma}{\Gamma'' \vdash \sigma \circ \tau \rhd \Gamma'}\ (-\circ-)$$

- Type judgement for substitution.

$$\frac{\Gamma \vdash \sigma \rhd \Gamma' \quad \Gamma' \vdash e : A}{\Gamma \vdash e[\sigma] : A}\ (subst)$$

- Intuition.
  - A substitution is "well-typed" if it provides an interpretation of the free variables in the context.

    In the rule (subst), e has fvs in Γ'; which might be $\{n : A_n, \ldots, 1 : A_1, 0 : A_0\}$.
    σ is a list of terms that matches the types of fvs in Γ'.
    Hence, this substitution is well-typed, and the result now lives in σ's context, Γ.

  - A composition of substitutions is well-typed if every component $\sigma_i[\tau]$ is well-typed.
    We know that, $\Gamma \vdash \sigma \rhd \Gamma'$, so $\Gamma \vdash \sigma_i : \Gamma'_i$. Substitutions doesn't change types. So, we are looking for:

$$\frac{\Gamma \vdash \sigma_i : \Gamma'_i \quad ? \vdash \tau \rhd \Gamma}{? \vdash \sigma_i[\tau] : \Gamma'_i}$$

and this $?$ can be any context.
That is why we say $\Gamma'' \vdash \tau \rhd \Gamma$ in the ~~premises~~ premises of rule $(-\circ-)$.

4. Dependent types.

Things are more-or-less the same, but more tricky.

- The syntax is standard. The equations for substitutions still hold.

- Judgements. $\boxed{\Gamma \vdash \sigma \rhd \Gamma'}$

$$\frac{}{\Gamma \vdash id \rhd \Gamma}(id) \qquad \frac{}{\Gamma, A \vdash \uparrow \rhd \Gamma}(\uparrow) \qquad \frac{\Gamma \vdash \sigma \rhd \Gamma' \quad \Gamma'' \vdash \tau \rhd \Gamma}{\Gamma'' \vdash \sigma \circ \tau \rhd \Gamma'}(-\circ-)$$

The above three rules are unchanged. Note the extension rule.

$$\frac{\Gamma \vdash e : A[\sigma] \qquad \Gamma \vdash \sigma : \Gamma'}{\Gamma \vdash \sigma, e : \Gamma', A}(-,-)$$

To have this, we must have $\Gamma' \vdash A : Set_i$. Hence the type of e is $A[\sigma]$, as $\Gamma \vdash A[\sigma] : Set_i$. If we set the type of e be A, then there's no way to get same $\Gamma' \vdash A' : Set_i$ from $\Gamma \vdash A : Set_i$.

$\boxed{\Gamma \vdash e : A}$

$$\frac{}{\Gamma, A \vdash 0 : A[\uparrow]}(0)$$

Note that we have to weaken A.
E.g. the counterpart of $\cdot, \cancel{Set} A : Set, x : A \vdash x : A$
is $\cdot, Set, 0 \vdash 0 : 1$, not $\ldots \vdash 0 : 0$.

$$\frac{\Gamma, A \vdash e : B}{\Gamma \vdash \lambda e : \Pi A B}(\lambda)$$

This is the same as usual.

$$\frac{\Gamma \vdash e_1 : \Pi A B \quad \Gamma \vdash e_2 : A}{\Gamma \vdash e_1\ e_2 : B[id, e_2]}(app)$$

Same as usual, except using explicit substitutions.

$$\frac{\Gamma \vdash e : A \qquad \Gamma' \vdash \sigma \rhd \Gamma}{\Gamma' \vdash e[\sigma] : A[\sigma]}(subst)$$

Note that we have to substitute into the type of e as well.