## Löb's Theorem

# A functional pearl of dependently typed quining

Name1
Affiliation1
Email1

Name2 Name3
Affiliation2/3
Email2/3

Categories and Subject Descriptors CR-number [subcategory]: third-level

General Terms Agda, Lob, quine, self-reference

Keywords Agda, Lob, quine, self-reference

#### **Abstract**

This is the text of the abstract.

If P's answer is 'Bad!', Q will suddenly stop. But otherwise, Q will go back to the top, and start off again, looping endlessly back, till the universe dies and turns frozen and black.

Excerpt from Scooping the Loop Snooper, by Geoffrey K. Pullum (http://www.lel.ed.ac.uk/~gpullum/loopsnoop.html())

#### 1. TODO

- cite Using Reflection to Explain and Enhance Type Theory?

## 2. Introduction

Löb's thereom has a variety of applications, from proving incompleteness of a logical theory as a trivial corrolary, to acting as a no-go theorem for a large class of self-interpreters (TODO: mention F<sub>o</sub>mega?), from allowing robust cooperation in the Prisoner's Dilemma with Source Code (), to curing social anxiety ().

"What is Löb's theorem, this versatile tool with wonderous applications?" you may ask.

Consider the sentence "if this sentence is true, then you, dear reader, are the most awesome person in the world." Suppose that this sentence is true. Then you, dear reader are the most awesome person in the world. Since this is exactly what the sentence asserts, the sentence is true, and you, dear reader, are the most awesome person in the world. For those more comfortable with symbolic logic, we can let X be the statement "you, dear reader, are the most awesome person in the world", and we can let A be the statement "if this sentence is true, then X". Since we have that A and  $A \to B$  are the same, if we assume A, we are also assuming  $A \to B$ , and

hence we have B, and since assuming A yields B, we have that  $A \to B$ . What went wrong?<sup>1</sup>

It can be made quite clear that something is wrong; the more common form of this sentence is used to prove the existence of Santa Claus to logical children: considering the sentence "if this sentence is true, then Santa Claus exists", we can prove that Santa Claus exists. By the same logic, though, we can prove that Santa Claus does not exist by considering the sentence "if this sentence is true, then Santa Claus does not exist." Whether you consider it absurd that Santa Claus exist, or absurd that Santa Claus not exist, surely you will consider it absurd that Santa Claus both exist and not exist. This is known as Curry's paradox.

Have you figured out what went wrong?

The sentence that we have been considering is not a valid mathematical sentence. Ask yourself what makes it invalid, while we consider a similar sentence that is actually valid.

Now consider the sentence "if this sentence is provable, then you, dear reader, are the most awesome person in the world." Fix a particular formalization of provability (for example, Peano Arithmetic, or Martin–Löf Type Theory). To prove that this sentence is true, suppose that it is provable. We must now show that you, dear reader, are the most awesome person in the world. If provability implies truth, then the sentence is true, and then you, dear reader, are the most awesome person in the world. Thus, if we can assume that provability implies truth, then we can prove that the sentence is true. This, in a nutshell, is Löb's theorem: to prove X, it suffices to prove that X is true whenever X is provable. Symbolically, this

$$\Box(\Box X - > X) \to \Box X$$

where  $\Box X$  means "X is provable" (in our fixed formalization of provability).

Let us now return to the question we posed above: what went wrong with our original sentence? The answer is that self-reference with truth is impossible, and the clearest way I know to argue for this is via the Curry–Howard Isomorphism; in a particular technical sense, the problem is that self-reference with truth fails to terminate.

The Curry–Howard Isomorphism establishes an equivalence between types and propositions, between (well-typed, terminating, functional) programs and proofs. See Table?? for some examples. Now we ask: what corresponds to a formalization of provability? If a proof of P is a terminating functional program which is well-typed at the type corresponding to P, and to assert that P is provable is to assert that the type corresponding to P is inhabited, then an encoding of a proof is an encoding of a program. Although mathematicians typically use Gödel codes to encode propositions and

[Copyright notice will appear here once 'preprint' option is removed.]

 $<sup>\</sup>overline{\ }^1$  Those unfamiliar with conditionals should note that the "if ... then ..." we use here is the logical "if", where "if false then X" is always true, and not the counterfactual "if".

proofs, a more natural choice of encoding programs will be abstract syntax trees. In particular, a valid syntactic proof of a given (syntactic) proposition corresponds to a well-typed syntax tree for an inhabitant of the corresponding syntactic type.

TODO: Table of CH-Equivalence: Type<->set of proofs<->Proposition, Term/(Terminating, Well-typed, functional) Program<->Proof, Function Type<->set of functions<->Implication, Pairing<->cartesian product<->Conjunction, Sum Type<->disjoint union<->disjunction

Unless otherwise specified, we will henceforth consider only well-typed, terminating programs; when we say "program", the adjectives "well-typed" and "terminating" are implied.

Before diving into Löb's theorem in detail, we'll first visit a standard paradigm for formalizing the syntax of dependent type theory. (TODO: Move this?)

#### 3. Quines

What is the computational equivalent of the sentence "If this sentence is provable, then X"? It will be something of the form "???  $\rightarrow X$ ". As a warm-up, let's look at a Python program that returns a string representation of this type.

To do this, we need a program that outputs its own source code. There are three genuinely distinct solutions, the first of which is degenerate, and the second of which is cheeky (or sassy?). These "cheating" solutions are:

- The empty program, which outputs nothing.
- The program print(open(\_\_file\_\_, 'r').read()), which
  relies on the Python interpreter to get the source code of the
  program.

Now we develop the standard solution. At a first gloss, it looks like:

```
(lambda my_type: '(' + my_type + ') -> X') "???"
```

Now we need to replace "???" with the entirety of this program code. We use Python's string escaping function (repr) and replacement syntax ("foo %s bar" % "baz" becomes "foo baz bar"):

This is a slight modification on the standard way of programming a quine, a program that outputs its own source-code. Suppose we have a function  $\square$  that takes in a string representation of a type, and returns the type of syntax trees of programs producing that type. Then our Löbian sentence would look something like (if  $\rightarrow$  were valid notation for function types in Python)

```
(lambda my_type: □ (my_type % repr(my_type)) → X)
  ("(lambda my_type: □ (my_type %% repr(my_type)) → X)\n
```

Now, finally, we can see what goes wrong when we consider using "if this sentence is true" rather than "if this sentence is provable". Provability corresponds to syntax trees for programs; truth corresponds to execution of the program itself. Our pseudo-Python thus becomes

```
(lambda my_type: eval(my_type % repr(my_type)) \rightarrow X) ("(lambda my_type: eval(my_type %% repr(my_type)) \rightarrow X)\n
```

This code never terminates! So, in a quite literal sense, the issue with our original sentence was that, if we tried to phrase it, we'd never finish.

Note well that the type ( $\square X \to X$ ) is a type that takes syntax trees and evaluates them; it is the type of an interpreter. (TODO: maybe move this sentence?)

# **4.** Abstract Syntax Trees for Dependent Type Theory

The idea of formalizing a type of syntax trees which only permits well-typed programs is common in the literature. (TODO: citations) For example, here is a very simple (and incomplete) formalization with  $\Pi$ , a unit type ( $\top$ ), an empty type ( $\bot$ ), and lambdas. (FIXME: What's the right level of simplicity?)

We begin with some standard data type declarations.

```
open import Agda. Primitive public
                   using (Level; ___; lzero; lsuc)
infix| 1 _,_
infixr 2 _x_
  infixl 1 _≡_
 record \top \{\ell\} : Set \ell where
                constructor tt
 data \perp \{\ell\}: Set \ell where
  record \Sigma \{\ell \ell'\} (A : \mathsf{Set} \ \ell) \ (P : A \to \mathsf{Set} \ \ell') : \mathsf{Set} \ (\ell \sqcup \ell') \ \mathsf{where}
                  constructor __,_
                  field
                                     proj_1: A
                                     proj_2 : P proj_1
 data Lifted \{a \ b\} (A : \mathsf{Set} \ a) : \mathsf{Set} \ (b \sqcup a) where
                  \mathsf{lift} : A \to \mathsf{Lifted}\,A
  lower: \forall \{a \ b \ A\} \rightarrow \mathsf{Lifted} \{a\} \{b\} A \rightarrow A
 lower (lift x) = x
             X \times \underline{\hspace{0.1in}} : \forall \; \{\ell \; \ell'\} \; (A : \mathsf{Set} \; \ell) \; (B : \mathsf{Set} \; \ell') \to \mathsf{Set} \; (\ell \sqcup \ell')
\overline{A} \times \overline{B} = \sum A (\lambda) \rightarrow B
data \equiv \{\ell\} \{A : \mathsf{Set} \ \ell\} \ (x : A) : A \to \mathsf{Set} \ \ell \ \mathsf{where}
 svm refl = refl
trans: \{A : \mathsf{Set}\} \to \{x \ y \ z : A\} \to x \equiv y \to y \equiv z \to x \equiv z
trans refl refl = refl
transport : \forall \{A : \mathsf{Set}\} \{x : A\} \{y : A\} \rightarrow (P : A \rightarrow \mathsf{Set}) \rightarrow x \equiv y \rightarrow P x 
transport P refl v = v
                   mutual
(%s)ih}ix∣2_⊳_
                                     data Context : Set where
                                                      \varepsilon: Context

hd \ dash \ : (\Gamma : \mathsf{Context}) 	o \mathsf{Type} \ \Gamma 	o \mathsf{Context}
                                     data Type : Context \rightarrow Set where
                                                          `\top": orall \left\{ \Gamma 
ight\} 
ightarrow \mathsf{Type} \ \Gamma
                (\%s)^{1}': \forall \{\Gamma\} \rightarrow \mathsf{Type}\ \Gamma
                                                      \Pi': orall \left\{\Gamma\right\} 
ightarrow (A:\mathsf{Type}\;\Gamma) 
ightarrow \mathsf{Type}\; (\Gamma 
hd A) 
ightarrow \mathsf{Type}\; \Gamma
                                     data Term : \{\Gamma: \mathsf{Context}\} 	o \mathsf{Type} \ \Gamma \ 	o \mathsf{Set} \ \mathsf{where}
                                                       \mathsf{'tt'}:\forall\ \{\Gamma\}\to\mathsf{Term}\ \{\Gamma\}\ \mathsf{'}\top\mathsf{'}
```

2 2016/2/26

 $\text{`}\lambda\text{'}:\forall \left\{\Gamma\,A\,B\right\} \to \mathsf{Term}\,\left\{\Gamma\triangleright A\right\}\,B \to \mathsf{Term}\,\left\{\Gamma\right\}\left(\text{`}\Pi\text{'}\,A\,B\right)$ 

An easy way to check consistency of a syntactic theory which is weaker than the theory of the ambient proof assistant is to define an interpretation function, also commonly known as an unquoter, or a denotation function, from the syntax into the universe of types. Here is an example of such a function:

# 

 $\sqsubseteq$  ' $\Pi$ '  $A B \sqcup \top \sqsubseteq \Gamma \sqcup = (x : \sqsubseteq A \sqcup \top \sqsubseteq \Gamma \sqcup) \rightarrow \sqsubseteq B \sqcup \top (\sqsubseteq \Gamma \sqcup , x)$ 

## 5. This Paper

In this paper, we make extensive use of this trick for validating models. We formalize the simplest syntax that supports Löb's theorem and prove it sound relative to Agda in 13 lines of code; the understanding is that this syntax could be extended to support basically anything you might want. We then present an extended version of this solution, which supports enough operations that we can prove our syntax sound (consistent), incomplete, and nonempty. In a hundred lines of code, we prove Löb's theorem under the assumption that we are given a quine; this is basically the well-typed functional version of the program that uses open(\_\_file\_\_, 'r').read(). Finally, we sketch our implementation of Löb's theorem (code in an appendix) based on the assumption only that we can add a level of quotation to our syntax tree; this is the equivalent of letting the compiler implement repr(), rather than implementing it ourselves. We close with an application to the prisoner's dilemma, as well as some discussion about avenues for removing the hard-coded repr.

#### 6. Prior Work

TODO: Use of Löb's theorem in program logic as an induction principle? (TODO)

TODO: Brief mention of Lob's theorem in Haskell / elsewhere / ? (TODO)

#### 7. Trivial Encoding

```
infixr 1 _'→'_

data Type : Set where

_'→'_ : Type \rightarrow Type \rightarrow Type

'□' : Type \rightarrow Type

data \square : Type \rightarrow Set where

\square is Type \rightarrow Set

\square : Type \rightarrow Set
```

```
|\ddot{o}bf = \bot \ddot{c}bf \rfloor t
```

infixr 1 \_'→'\_

# 8. Encoding with Soundness, Incompleteness, and Non-Emptyness

```
mutual
     data Type : Set where
             ' \! 	o ' \quad : \mathsf{Type} 	o \mathsf{Type} 	o \mathsf{Type}
          \overline{} '\square' : \overline{\mathsf{Type}} \to \mathsf{Type}
          '\top': Type
          '⊥': Type
         \mathsf{L\ddot{o}b}: \forall \{X\} \rightarrow \square (`\square` X`\rightarrow` X) \rightarrow \square X
          'tt' : □ 'T
mutual
    \bot \bot: Type \rightarrow Set
    \ \ \ A \ `\rightarrow `B \ \ = \ \ A \ \ \ \rightarrow \ \ \ B \ \ \ \ 
    \square '\square' T \square = \square T
    \Box 'T' \Box = T
    \bot \bott: \forall \{T: \mathsf{Type}\} \rightarrow \Box T \rightarrow \bot T \bot
    \Box (Löb \Box 'X'\rightarrowX) \Boxt = \Box 'X'\rightarrowX \Boxt (Löb \Box 'X'\rightarrowX)
     \bot 'tt' \bott = tt
\neg \quad : \mathsf{Set} \to \mathsf{Set}
\neg T = T \rightarrow \bot
'\neg': Type \rightarrow Type
'\neg \ddot{T} = T'\rightarrow '\bot
|\ddot{\mathsf{o}}\mathsf{b}: \forall \{ `X' \} \to \Box \ (`\Box' \ `X' \ `\to' \ `X') \to \llcorner \ `X' \ \lrcorner
|\ddot{\mathsf{o}}\mathsf{b}\,f = \mathsf{L}\,\ddot{\mathsf{o}}\mathsf{b}\,f\,\mathsf{J}\mathsf{t}
incompleteness : \neg \Box ('\neg' ('\Box' '\bot'))
incompleteness = löb
soundness: \neg \Box '\bot '
soundness x = \lfloor x \rfloor t
non-emptyness : \square '\top'
non-emptyness = 'tt'
```

### 9. Encoding with Quines

3

```
module lob-by-quines where infix! 2 \_ \triangleright \_ infix! 3 \_ ``\_ infix! 3 \_ ``\_ infix! 3 \_ ``a\_ infix! 3 \_ ``a\_ infix! 3 \_ w````a\_ infix: 2 \_ `o`\_ mutual data Context: Set where \varepsilon : \mathsf{Context} \_ \triangleright \_ : (\Gamma : \mathsf{Context}) \to \mathsf{Type} \ \Gamma \to \mathsf{Context}
```

```
data Type : Context \rightarrow Set where
                                                                                                                                                                    \mathsf{Term} \Downarrow (\mathsf{quine} \leftarrow \{\phi\}) \ \Gamma \Downarrow x = x
            W : \forall \{\Gamma A\} \rightarrow \mathsf{Type} \ \Gamma \rightarrow \mathsf{Type} \ (\Gamma \triangleright A)
                                                                                                                                                                    \mathsf{Term} \Downarrow (`\lambda \bullet `f) \; \Gamma \Downarrow x = \mathsf{Term} \Downarrow f \; (\Gamma \Downarrow , x)
            \Box : orall \left\{ \Gamma A 
ight\} 	o \mathsf{Type} \left( \Gamma 	riangle A 
ight) 	o \mathsf{Term} \left\{ \Gamma 
ight\} A 	o \mathsf{Type} \left[ \Gamma 
ight]
                                                                                                                                                                    \mathsf{Term} \Downarrow (\leftarrow \mathsf{SW1SV} \rightarrow \mathsf{W} f) = \mathsf{Term} \Downarrow f
            'Typeε' : \forall \{\Gamma\} 	o \mathsf{Type}\ \Gamma
                                                                                                                                                                    \mathsf{Term} \Downarrow (\rightarrow \mathsf{SW1SV} \rightarrow \mathsf{W} f) = \mathsf{Term} \Downarrow f
            \Box': \forall \{\Gamma\} \rightarrow \mathsf{Type} \ (\Gamma \rhd `\mathsf{Type}\epsilon')
                                                                                                                                                                    \mathsf{Term} \Downarrow (\mathsf{w} \ x) \ \Gamma \Downarrow = \mathsf{Term} \Downarrow x \ (\Sigma.\mathsf{proj}_1 \ \Gamma \Downarrow)
               '	o' : orall \left\{\Gamma
ight\} 	o \mathsf{Type} \; \Gamma 	o \mathsf{Type} \; \Gamma
                                                                                                                                                                    \mathsf{Term} \Downarrow (\mathsf{w} \to f) \; \Gamma \Downarrow = \mathsf{Term} \Downarrow f (\Sigma, \mathsf{proj}_1 \; \Gamma \Downarrow)
            Quine: Type (\varepsilon \triangleright \text{'Type}\varepsilon') \rightarrow \text{Type }\varepsilon
                                                                                                                                                                    \mathsf{Term} \Downarrow (g \circ f) \Gamma \Downarrow x = \mathsf{Term} \Downarrow g \Gamma \Downarrow (\mathsf{Term} \Downarrow f \Gamma \Downarrow x)
                                                                                                                                                                    \mathsf{Term} \Downarrow (f \mathsf{w}''''_a x) \Gamma \Downarrow = \mathsf{lift} (\mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ''_a \mathsf{lower} (\mathsf{Term} \Downarrow x \Gamma \Downarrow)
            \top : \forall \{\Gamma\} \rightarrow \mathsf{Type} \ \Gamma
            '\bot': orall \left\{\Gamma\right\} 
ightarrow \mathsf{Type} \ \Gamma
                                                                                                                                                               module inner ('X': Type \varepsilon) ('f': Term \{\varepsilon\} ('\square' '' \sqcap 'X' \sqcap '\rightarrow' 'X')) where
      data Term : \{\Gamma : \mathsf{Context}\} \to \mathsf{Type}\ \Gamma \to \mathsf{Set}\ \mathsf{where}
                                                                                                                                                                    'H': Type ε
            \lceil \ \rceil : \forall \{\Gamma\} \rightarrow \mathsf{Type} \ \epsilon \rightarrow \mathsf{Term} \ \{\Gamma\} \ \mathsf{`Type}\epsilon'
                                                                                                                                                                    'H' = Quine (W1 '\square' '' 'VAR_0' '\rightarrow' W 'X')
                   (\neg \forall AR_0) \neg t' : \forall \{T\} \rightarrow \mathsf{Term} \{ \varepsilon \triangleright (\neg ' ' ' \cap T \neg \} (W (\neg ' ' \cap \neg ' \neg ' \neg T \bullet H)) : \neg ((\neg ' ' \cap ' H' \neg ' \rightarrow ' X') \rightarrow ' H') ) 
            \texttt{`}\lambda \bullet \texttt{'} : \forall \ \{\Gamma \ A \ B\} \rightarrow \mathsf{Term} \ \{\Gamma \ \triangleright A\} \ (\mathsf{W} \ B) \rightarrow \mathsf{Term} \ \{\Gamma\} \ (A \ `\rightarrow `B) \qquad \texttt{`toH'} = \leftarrow \mathsf{SW1SV} \rightarrow \mathsf{W} \ \mathsf{quine} \leftarrow \mathsf{W} 
            \mathsf{'VAR}_0' : \forall \{\Gamma T\} \to \mathsf{Term} \{\Gamma \triangleright T\} (\mathsf{W} T)
               \overline{\mathsf{quine}} \to : \forall \{\phi\} \to \mathsf{Term} \{\epsilon\} (\mathsf{Quine} \ \phi \ ' \to ' \ \phi '' \ \Box \ \mathsf{Quine} \ \phi \ \urcorner)
                                                                                                                                                                    'fromH' = \rightarrowSW1SV\rightarrowW quine\rightarrow
            \mathsf{quine} \leftarrow : \forall \ \{\phi\} \rightarrow \mathsf{Term} \ \{\varepsilon\} \ (\phi \, " \, \square \, \mathsf{Quine} \ \phi \, " \, " \rightarrow " \, \mathsf{Quine} \ \phi)
                                                                                                                                                                    `\Box`H'\to\Box`X'':\Box(`\Box'``\ulcorner`H'\urcorner`\to'`\Box'``\ulcorner`X'\urcorner)
            \mathsf{'tt'}: \forall \ \{\Gamma\} \to \mathsf{Term} \ \{\Gamma\} \ \mathsf{'} \top
            \rightarrow SW1SV \rightarrow W : \forall \{ \widetilde{\Gamma} \widetilde{T}XAB \} \{x : \mathsf{Term} X \}
                                                                                                                                                                    \rightarrow \mathsf{Term} \{ \Gamma \} (T \xrightarrow{\cdot} ) (\mathsf{W1} A \xrightarrow{\cdot} \mathsf{VAR}_0 \xrightarrow{\cdot} ) \mathsf{W} B ) \xrightarrow{\cdot} x )
                  \rightarrow \mathsf{Term} \; \{\Gamma\} \; (T \; \stackrel{\cdot}{\rightarrow} \; \stackrel{\cdot}{A} \; ^{\cdot \prime} \; x \; \stackrel{\cdot}{\rightarrow} \; ^{\prime} B)
                                                                                                                                                                    'h': Term 'H'
            \leftarrow \mathsf{SW1SV} \rightarrow \mathsf{W} : \forall \ \{\Gamma \ TXA \ B\} \ \{x : \mathsf{Term} \ X\}
                                                                                                                                                                    \mathsf{'h'} = \mathsf{'toH'} \, \mathsf{''}_a \, (f' \, \mathsf{'o'} \, \mathsf{'} \square \mathsf{'H'} \rightarrow \square \mathsf{'X''})
                   \rightarrow \mathsf{Term} \; \{ \Gamma \} \; ((\mathsf{W} 1 \, A \, '' \, '\mathsf{VAR}_0 \, ' \, ' \to ' \, \mathsf{W} \, B) \, '' \, x \, ' \to ' \, T) \\ \rightarrow \mathsf{Term} \; \{ \Gamma \} \; ((A \, '' \, x \, ' \to ' \, B) \, ' \to ' \, T) 
                                                                                                                                                                    Löb : □ 'X'
                                                                                                                                                                    \mathsf{L\ddot{o}b} = \mathsf{`fromH'''}_a \, \mathsf{`h'''}_a \, \ulcorner \, \mathsf{`h'} \, \urcorner \mathsf{t}
            \mathsf{w}: \forall \; \{\Gamma \, A \, T\} \to \mathsf{Term} \; \{\Gamma\} \, A \to \mathsf{Term} \; \{\Gamma \triangleright T\} \; (\mathsf{W} \, A)
            \mathsf{w} \to \ : \forall \ \{\Gamma \ A \ B \ X\} \to \mathsf{Term} \ \{\Gamma\} \ (A \ `\to `B) \to \mathsf{Term} \ \{\Gamma \ \! \triangleright X\} \ (\mathbb{W} \ A \ `\to `\mathbb{W} \ B)
            \bot_\lrcorner: Type \epsilon \to \mathsf{Set} _
\square:\mathsf{Type}\;\epsilon\to\mathsf{Set}\;\_
                                                                                                                                                              \bot T \bot = \mathsf{Type} \Downarrow T \mathsf{tt}
\square = \text{Term } \{ \epsilon \}
                                                                                                                                                              \label{eq:def:gamma} \begin{array}{l} \text{`}\neg\text{'}\underline{\phantom{}} : \forall \; \{\Gamma\} \to \mathsf{Type} \; \Gamma \to \mathsf{Type} \; \Gamma \\ \text{`}\neg\text{'}\underline{\phantom{}} T = T \; \stackrel{}{\to} \text{'} \; \stackrel{}{\to} \text{'} \; \stackrel{}{\bot} \text{'} \end{array}
max-level : Level
max-level = |zero
                                                                                                                                                              |\ddot{\mathsf{o}}\mathsf{b}:\forall\ \{\, `X'\}\to \square\ (\,\dot{}\,\Box\,,\,\,\dot{}\,\,,\,\,\Box\,\,(\,\dot{}\,\,X'\,\,\neg\,\,\,\dot{}\,\,,\,\,\dot{}\,\,\,\dot{}\,\,\,\dot{}\,\,\,\dot{}\,\,\,)\to \ \ \ \ `X'\,\,\,\Box\,\,\,\dot{}\,\,\,
                                                                                                                                                              |\ddot{o}b f = \text{Term} \Downarrow (\text{L\"{o}b } f) \text{ tt}
      \mathsf{Context} \Downarrow : (\Gamma : \mathsf{Context}) \to \mathsf{Set} (|\mathsf{suc\ max-level})
      Context\psi \epsilon = \top
                                                                                                                                                               \neg \quad : \forall \ \{\ell\} \to \mathsf{Set} \ \ell \to \mathsf{Set} \ \ell
      \mathsf{Context} \Downarrow (\Gamma \triangleright T) = \Sigma \; (\mathsf{Context} \Downarrow \Gamma) \; (\mathsf{Type} \Downarrow \{\Gamma\} \; T)
                                                                                                                                                               \neg \{\ell\}\ T = T \rightarrow \bot \{\ell\}
      \mathsf{Type} \Downarrow (\mathsf{W} \ T) \ \Gamma \Downarrow = \mathsf{Type} \Downarrow T \left( \Sigma . \mathsf{proj}_1 \ \Gamma \Downarrow \right)
                                                                                                                                                              incompleteness = löb
      \mathsf{Type} \Downarrow (\mathsf{W}1\ T)\ \Gamma \Downarrow = \mathsf{Type} \Downarrow T\ ((\Sigma.\mathsf{proj}_1\ (\Sigma.\mathsf{proj}_1\ \Gamma \Downarrow))\ ,\ (\Sigma.\mathsf{proj}_2\ \Gamma \Downarrow))
      \mathsf{Type} \Downarrow (T ``x) \Gamma \Downarrow = \mathsf{Type} \Downarrow T (\Gamma \Downarrow , \mathsf{Term} \Downarrow x \Gamma \Downarrow)
                                                                                                                                                              soundness: \neg \Box '\bot '
                                                                                                                                                              soundness x = \text{Term} \psi x \text{ tt}
      Type\downarrow 'Type\epsilon' \Gamma \downarrow = Lifted (Type \epsilon)
      Type\Downarrow '\square' \Gamma \Downarrow = \text{Lifted (Term } \{\epsilon\} \text{ (lower } (\Sigma.\text{proj}_2 \Gamma \Downarrow)))
      \mathsf{Type} \Downarrow (A \hookrightarrow B) \Gamma \Downarrow = \mathsf{Type} \Downarrow A \Gamma \Downarrow \to \mathsf{Type} \Downarrow B \Gamma \Downarrow
                                                                                                                                                               non-emptyness : \Sigma (Type \varepsilon) (\lambda T \rightarrow \square T)
      Type\Downarrow '\top' \Gamma \Downarrow = \top
                                                                                                                                                              non-emptyness = \dot{\top}, 'tt'
      Type\Downarrow '\bot' \Gamma \Downarrow = \bot
      \mathsf{Type} \Downarrow (\mathsf{Quine} \ \phi) \ \Gamma \Downarrow = \mathsf{Type} \Downarrow \phi \ (\Gamma \Downarrow \ , \ (\mathsf{lift} \ (\mathsf{Quine} \ \phi)))
      \mathsf{Term} \Downarrow : \forall \{\Gamma : \mathsf{Context}\} \{T : \mathsf{Type}\ \Gamma\} \to \mathsf{Term}\ T \to (\Gamma \Downarrow : \mathsf{Cont} \mathbf{M} \Downarrow \Gamma \mathbf{Dignession} \Gamma \mathbf{Application}\ \mathbf{of}\ \mathbf{Quining}\ \mathbf{to}\ \mathbf{The}
      \mathsf{Term} \Downarrow \ulcorner x \urcorner \Gamma \Downarrow = \mathsf{lift} \ x
                                                                                                                                                                    Prisoner's Dilemma
      \mathsf{Term} \! \Downarrow \ulcorner x \urcorner \mathsf{t} \; \Gamma \! \Downarrow \; = \mathsf{lift} \; x
                                                                                                                                                               module prisoners-dilemma where
      \mathsf{Term} \dot{\Downarrow} \; (f^{\, {}^{\prime \prime}}{}_a \; x) \; \Gamma \Downarrow = \mathsf{Term} \! \Downarrow \! f \, \Gamma \! \Downarrow \; (\mathsf{Term} \! \Downarrow \! x \; \Gamma \! \Downarrow)
                                                                                                                                                               module lob where
      Term\Downarrow 'tt' \Gamma \Downarrow = tt
                                                                                                                                                                   infix| 2 _⊳_
infix| 3 _''_
      Term\Downarrow (quine\rightarrow \{\phi\}) \Gamma \Downarrow x = x
```

```
\begin{array}{c} \inf x \ 1 \ \_`\to '\_ \\ \inf x \ 1 \ \_``\to ''\_ \end{array}
                                                                                                                                                                                                                                                                                                                                                                                             - ''''' : \forall {\Gamma A} \rightarrow Term {\Gamma \triangleright A} ('Type' (\Gamma \triangleright A) '\rightarrow
                                                                                                                                                                                                                                                                                                                                                                                                 \_``\to ``\_: \forall \ \{\Gamma\} \to \mathsf{Term} \ \{\Gamma\} \ (\text{`Type'} \ \Gamma) \to \mathsf{Term} \ \{\Gamma\} \ (\text{`Type'} \ \Gamma)
                                                                                                                                                                                                                                                                                                                                                                                                 \begin{array}{c} \text{ww''} \rightarrow \text{'''} \\ \text{ww''} \rightarrow \text{'''} \\ \text{ww''} \times \text{'''} \\ \text{:} \forall \left\{ \Gamma A B \right\} \rightarrow \mathsf{Term} \left\{ \Gamma \triangleright A \triangleright B \right\} \text{ (W (W ('Term' '' \cap A'')'')} \\ \text{ww''} \times \text{'''} \\ \text{:} \forall \left\{ \Gamma A B \right\} \rightarrow \mathsf{Term} \left\{ \Gamma \triangleright A \triangleright B \right\} \text{ (W (W ('Term' '' \cap A'')'')} \\ \end{array} 
  \mathsf{infixr} \ 1 \ \_ww``` \to ```\_
infix| 3 _''a_
infix| 3 _w'''a_
infixr 2_'o'_
                                                                                                                                                                                                                                                                                                                                                                  \square: Type \epsilon \to \mathsf{Set}
 infixr 2_'x'
                                                                                                                                                                                                                                                                                                                                                                  \square = \text{Term } \{\epsilon\}
 infixr 2 _".x"
  infixr 2 _w"×",_
                                                                                                                                                                                                                                                                                                                                                                   \square : \forall \{\Gamma\} \rightarrow \mathsf{Type}\ \Gamma \rightarrow \mathsf{Type}\ \Gamma
                                                                                                                                                                                                                                                                                                                                                                   \square T = \text{`Term'} \cap T
  mutual
                                                                                                                                                                                                                                                                                                                                                                           \text{``}\times\text{''}\quad:\forall\;\{\Gamma\}\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\{\Gamma\}\;(\text{`Type'}\;\Gamma)\rightarrow\mathsf{Term}\;\Gamma\}
              data Context : Set where
                                                                                                                                                                                                                                                                                                                                                                  \overline{A} "×"\overline{B} = "×" " \overline{A} A " \overline{A} B
                            \epsilon: Context

hd \ dash \ : (\Gamma : \mathsf{Context}) 	o \mathsf{Type} \ \Gamma 	o \mathsf{Context}
                                                                                                                                                                                                                                                                                                                                                                  max-level: Level
               data Type : Context → Set where
                                                                                                                                                                                                                                                                                                                                                                  max-level = |zero
                           \mathsf{W}: \forall \{\Gamma A\} \to \mathsf{Type}\ \Gamma \to \mathsf{Type}\ (\Gamma \triangleright A)
                            \mathsf{W1} : \forall \{\Gamma A B\} \to \mathsf{Type} \ (\Gamma \triangleright B) \to \mathsf{Type} \ (\Gamma \triangleright A \triangleright (\mathsf{W} \ \{\Gamma = \Gamma\} \ \{A \cap \# A A \} B))
                                   : orall \left\{ \Gamma A 
ight\} 	o \mathsf{Type} \; (\Gamma 	riangle A) 	o \mathsf{Term} \; \left\{ \Gamma 
ight\} A 	o \mathsf{Type} \; \Gamma
                                                                                                                                                                                                                                                                                                                                                                               \mathsf{Context} \Downarrow : (\Gamma : \mathsf{Context}) \to \mathsf{Set} (|\mathsf{suc} \, \mathsf{max-level})
                            \overline{\,\,\,}Type' : \forall \ \Gamma \to \mathsf{Type} \ \Gamma
                                                                                                                                                                                                                                                                                                                                                                                \mathsf{Context} \! \! \downarrow \! \! \epsilon = \top
                             \text{`Term'}:\forall\ \{\Gamma\} \to \mathsf{Type}\ (\Gamma \rhd \text{`Type'}\ \Gamma)
                                                                                                                                                                                                                                                                                                                                                                                \mathsf{Context} \Downarrow (\Gamma \triangleright T) = \Sigma \; (\mathsf{Context} \Downarrow \Gamma) \; (\mathsf{Type} \Downarrow \{\Gamma\} \; T)
                            \mathsf{Type} \Downarrow : \{\Gamma : \mathsf{Context}\} \to \mathsf{Type} \; \Gamma \to \mathsf{Context} \Downarrow \Gamma \to \mathsf{Set} \; \mathsf{max}\text{-}\mathsf{|eve|}
                            \mathsf{Quine}: \forall \ \{\Gamma\} \to \mathsf{Type} \ (\Gamma \rhd \text{`Type'} \ \Gamma) \to \mathsf{Type} \ \Gamma
                                                                                                                                                                                                                                                                                                                                                                                \mathsf{Type} \Downarrow (\mathsf{W}\ T)\ \Gamma \Downarrow = \mathsf{Type} \Downarrow T\left(\Sigma.\mathsf{proj}_1\ \Gamma \Downarrow\right)
                             \text{`}\top\text{'}:\forall\ \{\Gamma\}\rightarrow\mathsf{Type}\ \Gamma
                                                                                                                                                                                                                                                                                                                                                                                \mathsf{Type} \Downarrow (\mathsf{W1}\ \mathit{T})\ \Gamma \Downarrow = \mathsf{Type} \Downarrow \mathit{T}\left(\left(\Sigma.\mathsf{proj}_1\ (\Sigma.\mathsf{proj}_1\ \Gamma \Downarrow)\right),\ (\Sigma.\mathsf{proj}_2\ \Gamma \Downarrow)\right)
                            '\bot': orall \left\{\Gamma\right\} 	o \mathsf{Type} \; \Gamma
                                                                                                                                                                                                                                                                                                                                                                                \mathsf{Type} \Downarrow (T ``x) \ \Gamma \Downarrow = \mathsf{Type} \Downarrow T (\Gamma \Downarrow \ , \ \mathsf{Term} \Downarrow x \ \Gamma \Downarrow)
                                                                                                                                                                                                                                                                                                                                                                                \mathsf{Type} \Downarrow (\mathsf{`Type'}\ \Gamma)\ \Gamma \Downarrow = \mathsf{Lifted}\ (\mathsf{Type}\ \Gamma)
               data \mathsf{Term}: \{\Gamma: \mathsf{Context}\} \to \mathsf{Type}\ \Gamma \to \mathsf{Set}\ \mathsf{where}
                                                                                                                                                                                                                                                                                                                                                                                \mathsf{Type} \Downarrow \mathsf{`Term'} \; \Gamma \Downarrow = \mathsf{Lifted} \; (\mathsf{Term} \; (\mathsf{lower} \; (\Sigma \, \mathsf{proj}_2 \; \Gamma \Downarrow)))
                                          _{\square} : orall \{\Gamma\} 	o \mathsf{Type}\ \Gamma 	o \mathsf{Term}\ \{\Gamma\}\ (\mathsf{`Type'}\ \Gamma)
                                                                                                                                                                                                                                                                                                                                                                                \mathsf{Type} \!\!\!\!\downarrow (A \ `\rightarrow `B) \ \Gamma \!\!\!\!\downarrow = \mathsf{Type} \!\!\!\downarrow \!\!\!\!\downarrow A \ \Gamma \!\!\!\!\downarrow \to \mathsf{Type} \!\!\!\downarrow \!\!\!\downarrow B \ \Gamma \!\!\!\!\downarrow
                           \begin{array}{c} \text{`} \vdash \forall \ \{\Gamma'\} \rightarrow \mathsf{Iype} \ \mathsf{I} \rightarrow \mathsf{Ierm} \ \{\Gamma\} \ T \rightarrow \mathsf{Term} \ \{\Gamma\} \ (\mathsf{`Term'} \ `` \vdash T \neg) \\ \text{``} \vdash \forall \ \{\Gamma \ T\} \rightarrow \mathsf{Term} \ \{\Gamma\} \ (\mathsf{`Term'} \ `` \vdash T \neg) \\ \text{``} \vdash \forall \ \{\Gamma \ T\} \rightarrow \mathsf{Term} \ \{\Gamma \ \vdash \mathsf{`Term'} \ `` \vdash T \neg\} \ (\mathsf{W} \ (\mathsf{`Term'} \ `` \vdash \mathsf{Type} \ \mathsf{``} \vdash \mathsf{``} \vdash \mathsf{Type} \ \mathsf{``} \vdash 
                                                                                                                                                                                                                                                                                                                                                                                \mathsf{Type} \Downarrow (A '\times 'B) \Gamma \Downarrow = \mathsf{Type} \Downarrow A \Gamma \Downarrow \times \mathsf{Type} \Downarrow B \Gamma \Downarrow
                            `\lambda \bullet' : \forall \ \{\Gamma \ A \ B\} \rightarrow \mathsf{Term} \ \{\Gamma \triangleright A\} \ (\mathsf{W} \ B) \rightarrow \mathsf{Term} \ \{\Gamma\} \ (A \ `\rightarrow' B)
                                                                                                                                                                                                                                                                                                                                                                              \mathsf{Type} \Downarrow (\mathsf{Quine} \ \phi) \ \Gamma \Downarrow = \mathsf{Type} \Downarrow \phi \ (\Gamma \Downarrow \ , \ (\mathsf{lift} \ (\mathsf{Quine} \ \phi)))
                             \mathsf{'VAR_0'} : \forall \ \{\Gamma \ \mathit{T}\} \to \mathsf{Term} \ \{\Gamma \rhd \mathit{T}\} \ (\mathsf{W} \ \mathit{T})
                           \mathsf{'tt'}:\forall\ \{\Gamma\}\to\mathsf{Term}\ \{\Gamma\}\ \mathsf{'}\top
                            \mathsf{SW} : \forall \ \{\Gamma \ X \ A\} \ \{a : \mathsf{Term} \ A\} \to \mathsf{Term} \ \{\Gamma\} \ (\mathsf{W} \ X \ ``\ a) \to \mathsf{Term} \ X
                                                                                                                                                                                                                                                                                                                                                                                \mathsf{Term} \Downarrow (f''_a x) \Gamma \Downarrow = \mathsf{Term} \Downarrow f \Gamma \Downarrow (\mathsf{Term} \Downarrow x \Gamma \Downarrow)
                           Term\psi 'tt' \Gamma \psi = tt
                                                                                                                                                                                                                                                                                                                                                                               \begin{array}{l} \operatorname{\mathsf{Term}} \Downarrow (\operatorname{\mathsf{quine}} \to \{\phi\}) \ \Gamma \Downarrow x = x \\ \operatorname{\mathsf{Term}} \Downarrow (\operatorname{\mathsf{quine}} \leftarrow \{\phi\}) \ \Gamma \Downarrow x = x \\ \operatorname{\mathsf{Term}} \Downarrow (`\lambda \bullet `f) \ \Gamma \Downarrow x = \operatorname{\mathsf{Term}} \Downarrow f (\Gamma \Downarrow \,, \, x) \\ \end{array} 
                            \leftarrowSW1SV\rightarrowW: \forall \{\Gamma TXAB\} \{x : \text{Term } X\}
                                         \mathsf{Term} \Downarrow \mathsf{VAR}_0' \Gamma \Downarrow = \Sigma.\mathsf{proj}_2 \Gamma \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                \mathsf{Term} \Downarrow (\mathsf{SW}\ t) = \mathsf{Term} \Downarrow t
                             \rightarrowSW1SV\rightarrowSW1SV\rightarrowW: \forall \{\Gamma TXAB\} \{x : \text{Term } X\}
                                                                                                                                                                                                                                                                                                                                                                                \mathsf{Term} \Downarrow (\leftarrow \mathsf{SW1SV} \rightarrow \mathsf{W} f) = \mathsf{Term} \Downarrow f
                                         \rightarrow \text{Term} \{\Gamma\} (T' \rightarrow (W1A'' VAR_0'' \rightarrow W1A'' VAR_0'' \rightarrow WB) rin () (\rightarrow SW1SV \rightarrow Wf) = \text{Term} (f') 
                                         \rightarrow \mathsf{Term} \ \{ \Gamma \} \ (T \ \rightarrow' \ A \ '' \ x \ \rightarrow' \ A \ '' \ x \ \rightarrow' \ B)
                                                                                                                                                                                                                                                                                                                                                                               \mathsf{Term} \Downarrow (\leftarrow \mathsf{SW1SV} \rightarrow \mathsf{SW1SV} \rightarrow \mathsf{W} f) = \mathsf{Term} \Downarrow f
                            \leftarrow \mathsf{SW1SV} \rightarrow \mathsf{SW1SV} \rightarrow \mathsf{W} : \forall \{\Gamma \ T \ X \ A \ B\} \{x : \mathsf{Term} \ X\}
                                                                                                                                                                                                                                                                                                                                                                               \mathsf{Term} \Downarrow (\rightarrow \mathsf{SW1SV} \rightarrow \mathsf{SW1SV} \rightarrow \mathsf{W} f) = \mathsf{Term} \Downarrow f
                                          \rightarrow \mathsf{Term} \ \{ \Gamma \} \ ((\mathsf{W}1\ A\ ''\ '\mathsf{VAR}_0'\ ' \rightarrow '\ \mathsf{W}1\ A\ ''\ '\mathsf{VAR}_0'\ ' \rightarrow '\ \mathsf{W}\ B)\ ''\ x\mathsf{Term} \ T)(\mathsf{w}\ x)\ \Gamma \Downarrow = \mathsf{Term} \ x\ (\Sigma.\mathsf{proj}_1\ \Gamma \Downarrow)
                                         \rightarrow \mathsf{Term} \left\{ \Gamma \right\} ((A " x ' \rightarrow ' A " x ' \rightarrow ' B) ' \rightarrow ' T)
                                                                                                                                                                                                                                                                                                                                                                               \mathsf{Term} \Downarrow (\mathsf{w} \rightarrow f) \; \Gamma \Downarrow = \mathsf{Term} \Downarrow f \; \Gamma \Downarrow
                           \mathsf{w}: \forall \; \{\Gamma\,A\,\mathit{T}\} \to \mathsf{Term}\; \{\Gamma\}\,A \to \mathsf{Term}\; \{\Gamma \,{dash}\,\mathit{T}\}\; (\mathsf{W}\,A)
                                                                                                                                                                                                                                                                                                                                                                               \mathsf{Term} \Downarrow (\rightarrow \mathsf{w} f) \; \Gamma \Downarrow = \mathsf{Term} \Downarrow f \; \Gamma \Downarrow
                            \begin{tabular}{l} $ `\circ' = : \forall \ \{\Gamma \ A \ B \ C\} \to \mathsf{Term} \ \{\Gamma\} \ (B \ \to' \ C) \to \mathsf{Term} \ \{\Gamma\} \ (A \ \to' \ B) \end{tabular} $ = \mathsf{Fin} \ (A) \ \Gamma \to \mathsf{F} \ (O) \ \Gamma \to \mathsf{Fin} \ (O) \ \Gamma
```

```
'other-cooperates-with' : \forall \{\Gamma\} 	o \mathsf{Term} \{\Gamma 	riangleright `\Box' 'Bot' 
hd \mathsf{W} ('\Box' 'Bot')\} ('
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      'other-cooperates-with' \{\Gamma\} = 'eval-other'' 'o' w\rightarrow (w (w\rightarrow (w (\lambda \bullet' '\Gamma'VA
                 module inner ('X': Type \varepsilon) ('f': Term \{\varepsilon\} ('\square' 'X' '\rightarrow' 'X')) where where
                                  'H': Type ε
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        'eval-other': Term \{\Gamma \rhd '\Box ' \; | \; \mathsf{Bot}' \rhd \mathsf{W} \; (\; \Box' \; '\mathsf{Bot}')\} \; (\mathsf{W} \; (\; \Box' \; (\; \Box' \; '\mathsf{Bot}'))\} \; (\mathsf{W} \; (\; \Box' \; (\; \Box' \; '\mathsf{Bot}'))\} \; (\mathsf{W} \; (\; \Box' \; (\; \Box' \; '\mathsf{Bot}'))) \; (\; \Box' \; (\; \Box' \; '\mathsf{Bot}')) \; (\; \Box' \; (\; \Box' \; (\; \Box' \; '\mathsf{Bot}'))) \; (\; \Box' \; (\; \Box' \; (\; \Box' \; '\mathsf{Bot}'))) \; (\; \Box' \; (\; \Box' \; (\; \Box' \; ))) \; (\; \Box' \; (\; \Box' \; )) \; (\; \Box' \; )) \; (\; \Box' \; (\; \Box' \; )) \; (\; \Box' \; )) \; (\; \Box' \; (\; \Box' \; )) \; (\; \Box' \;
                                 'H' = Quine (W1 'Term' '' 'VAR_0' '\rightarrow' W 'X')
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        'eval-other' = w \rightarrow (w (w \rightarrow (w \text{ ''eval-bot'''}))) ''<sub>a</sub> 'VAR<sub>0</sub>
                                 \mathsf{'toH'}: \square ((\mathsf{'\square'} \mathsf{'H'} \mathsf{'} \to \mathsf{'} \mathsf{'} X') \mathsf{'} \to \mathsf{'} \mathsf{'H'})
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       'eval-other'' : Term (W (W ('\square' ('\square' 'Bot'))) '\rightarrow' W (W ('\square' ('Type'
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        'eval-other'' = ww \rightarrow (w \rightarrow (w (w \rightarrow (w'''a')))''a 'eval-other')
                                 \text{`toH'} = \leftarrow \text{SW1SV} \rightarrow \text{W quine} \leftarrow
                                 \mathsf{'fromH'}: \square (\mathsf{'H'} \; \mathsf{'} \to \mathsf{'} \; (\mathsf{'}\square\mathsf{'} \; \mathsf{'H'} \; \mathsf{'} \to \mathsf{'} \; \mathsf{'}X'))
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      \mathsf{`self'}: \forall \ \{\Gamma\} \to \mathsf{Term} \ \{\Gamma \rhd \mathsf{`\Box'} \mathsf{`Bot'} \rhd \mathsf{W} \ (\mathsf{`\Box'} \mathsf{`Bot'})\} \ (\mathsf{W} \ (\mathsf{W} \ (\mathsf{`\Box'} \mathsf{`Bot'})))
                                 'fromH' = \rightarrowSW1SV\rightarrowW quine\rightarrow
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      'self' = w 'VAR<sub>0</sub>
                                 \texttt{`other'}: \forall \left\{\Gamma\right\} \rightarrow \mathsf{Term} \left\{\Gamma \rhd `\Box' `\mathsf{Bot'} \rhd \mathsf{W} (`\Box' `\mathsf{Bot'})\right\} (\mathsf{W} (\mathsf{W} (`\Box' `\mathsf{Bot'})))
                                 '\Box'H' \rightarrow \Box'X'' = '\lambda \bullet' (w \ ''fromH' \ 't \ w''' \ '' \ 'VAR_0' \ w'''' \ '' \ ''VAR_0' \circ t'' \ '' \ '' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ ''' \ '
                                 'h': Term 'H'
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     \mathsf{make}\mathsf{-bot}: \forall \ \{\Gamma\} \to \mathsf{Term} \ \{\Gamma \, \triangleright \, `\Box' \, `\mathsf{Bot}' \, \triangleright \, \mathsf{W} \, (\, `\Box' \, \, `\mathsf{Bot}')\} \, \big(\mathsf{W} \, \big(\mathsf{W} \, \, \big(\, \mathsf{`Type'} \, \mathsf{``Bot'} \, \big)\} \, \big(\mathsf{W} \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big)\} \, \big(\mathsf{W} \, \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big)\} \, \big(\mathsf{W} \, \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big)\} \, \big(\mathsf{W} \, \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big)\} \, \big(\mathsf{W} \, \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big)\} \, \big(\mathsf{W} \, \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big)\} \, \big(\mathsf{W} \, \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big)\} \, \big(\mathsf{W} \, \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big)\} \, \big(\mathsf{W} \, \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big)\} \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Bot'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Type'} \, \big) \big) \, \big(\mathsf{``Type'} \, \mathsf{``Type'} \, \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Type'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Type'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \mathsf{``Type'} \, \big) \big) \, \big(\mathsf{W} \, \, \big(\, \mathsf{``Type'} \, \big) \, \big(\mathsf{``Type'} \, \big
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     make-bot t = \leftarrow SW1SV \rightarrow SW1SV \rightarrow W quine \leftarrow ''_a '\lambda \bullet' (\rightarrow w ('\lambda \bullet' t))
                                 \mathsf{'h'} = \mathsf{'toH'} \, \mathsf{''}_a \, (\mathsf{'}f' \, \mathsf{'o'} \, \mathsf{'}\Box \mathsf{'H'} \to \Box \mathsf{'X''})
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    \mathsf{ww}```\neg```\_:\forall\:\{\Gamma\:A\:B\}
                                 Löb : □ 'X'
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      \rightarrow \mathsf{Term} \; \{\Gamma \, \check{\triangleright} \, A \, \rhd \, \check{B\}} \; (\mathsf{W} \; (\mathsf{`} \Box \, ' \; (\mathsf{`Type'} \; \Gamma))))
                                 \mathsf{L\ddot{o}b} = \mathsf{`fromH'} \ ``_a \ \mathsf{`h'} \ ``_a \ \ulcorner \ \mathsf{`h'} \ \urcorner \mathsf{t}
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  \mathsf{L\ddot{o}b} : \forall \ \{X\} \to \mathsf{Term} \ \{\epsilon\} \ (`\Box' \ X \ `\to' \ X) \to \mathsf{Term} \ \{\epsilon\} \ X
                L\ddot{o}b \{X\}f = inner.L\ddot{o}b Xf
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      'DefectBot' : \Box 'Bot'
              \  \  \, \sqsubseteq \, \bot : \mathsf{Type} \ \epsilon \to \mathsf{Set} \ \_
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      'CooperateBot' : ☐ 'Bot'
              T = Type  Ttt
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        'FairBot' : ☐ 'Bot'
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        'PrudentBot' : □ 'Bot'
              \label{eq:def:gamma} \begin{array}{l} \text{`}\neg\text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}} \text{'}\underline{\phantom{}}}
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      \label{eq:DefectBot} \begin{split} \text{`DefectBot'} &= \mathsf{make\text{-}bot} \left( w \; (w \; \ulcorner \; \bot \; \urcorner) \right) \\ \text{`CooperateBot'} &= \mathsf{make\text{-}bot} \left( w \; (w \; \ulcorner \; \top \; \urcorner) \right) \end{split}
                      \overline{A} \text{ w''} \times \overline{B} = \overline{W} \rightarrow (\overline{W} \rightarrow (\overline{W} \rightarrow (\overline{W} \rightarrow \overline{W} \rightarrow \overline{W}) ) ) a A) a B
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        \varphi_0:\forall \ \{\Gamma\} \to \mathsf{Term} \ \{\Gamma \rhd `\Box' `\mathsf{Bot'} \rhd \mathsf{W} \ (`\Box' `\mathsf{Bot'})\} \ (\mathsf{W} \ (\mathsf{W} \ (`\Box' \ (\mathsf{Typ}))) \}
                \mathsf{l\ddot{o}b} : \forall \ \{ \ X' \} \rightarrow \square \ ( \ ^{\boldsymbol{\cdot}}\square \ ^{\boldsymbol{\cdot}} \ X' \ \rightarrow \ ^{\boldsymbol{\cdot}} \ X' ) \rightarrow \boldsymbol{\llcorner} \ ^{\boldsymbol{\cdot}} X' \ \boldsymbol{\lrcorner}
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        \phi_0 = 'other-cooperates-with' "a 'self'
                |\ddot{o}b f = \text{Term} \Downarrow (\text{L\"{o}b } f) \text{ tt}
                \neg\_: \forall \: \{\ell\} \to \mathsf{Set} \: \ell \to \mathsf{Set} \: \ell \\ \neg\_\: \{\ell\} \: T = T \to \bot \: \{\ell\}
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       other\text{-}defects\text{-}against\text{-}DefectBot: Term } \{\_ \rhd `\Box' `Bot' \rhd W \ (`\Box' `Bot other\text{-}defects\text{-}against\text{-}DefectBot} = ww```\neg'`' \ (`other\text{-}cooperates\text{-}with other)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        \neg\Box\bot:\forall\;\{\Gamma\,A\,B\}\to\mathsf{Term}\;\{\Gamma\triangleright A\triangleright B\}\;(\mathsf{W}\;(\mathsf{W}\;(\dot{}\;\Box\dot{}\;(\dot{}\;\mathsf{Type'}\;\Gamma))))\\\neg\Box\bot=\mathsf{w}\;(\mathsf{w}\;\ulcorner\,\dot{}\;\neg\dot{}\;(\dot{}\;\Box\dot{}\;\dot{}\;\dot{}\;\dot{}\;\dot{}\;)\;\neg\,\mathsf{t})
                incompleteness : \neg \Box ('\neg' ('\Box' '\bot'))
              incompleteness = löb
              soundness : \neg \Box '\bot '
                soundness x = \text{Term} \Downarrow x \text{ tt}
                                                                                                                                                                                                                                                                                                                                                                                                                                                         11. Encoding with Add-Quote Function
                                                                                                                                                                                                                                                                                                                                                                                                                                                         (appendix) - Discuss whiteboard phrasing of sentence with sigmas
                non-emptyness : \Sigma (Type \varepsilon) (\lambda T \rightarrow \Box T)
                                                                                                                                                                                                                                                                                                                                                                                                                                                            - It remains to show that we can construct - Discuss whiteboard
                non-emptyness = '\dot{\top}', 'tt'
                                                                                                                                                                                                                                                                                                                                                                                                                                                          phrasing of untyped sentence - Given: - X - \square = \text{Term} - f : \square 'X'
                                                                                                                                                                                                                                                                                                                                                                                                                                                          -> X - define y:X - Suppose we have a type H\cong Term\ ^{\lceil}\ H\to X
12. Removing add-quote and actually tying the
         cooperates-with : \Box 'Bot' \rightarrow \Box 'Bot' \rightarrow Type \epsilon
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      knot (future work 1)
b1 cooperates-with b2 = |\text{ower} (\text{Term} \Downarrow b1 \text{ tt (lift } b1) (\text{lift } b2))|
                                                                                                                                                                                                                                                                                                                                                                                                                                                          - Bibliography - Appendix - Temporary outline section to be moved
'eval-bot'' : \forall \{\Gamma\} \rightarrow \text{Term } \{\Gamma\} ('Bot' '\rightarrow' ('\Box' 'Bot' '\rightarrow' '\Box' '
```

 $\text{`eval-bot''} = \rightarrow \text{SW1SV} \rightarrow \text{SW1SV} \rightarrow \text{W quine} \rightarrow$ 

 $\text{``eval-bot'''}: \forall \{\Gamma\} \rightarrow \mathsf{Term} \{\Gamma\} \text{ (`$\square$' `Bot' `$\rightarrow$' `$\square$' ({- other -} {-}))}$ 

2016/2/26

We will say that a type-theoretic quine is a program that outputs its own (well-typed) abstract syntax tree. Generalizing this slightly, we can consider programs that output an arbitrary function of their own syntax trees. - TODO: Examples of double quotation, single quotation, etc. - Given any function φ from doubly-quoted syntactic types to singly-quoted syntactic types, and given an operator \\_\ which adds an extra level of quotation, we can define the type of a quine at  $\phi$  to be a (syntactic) type "Quine  $\phi$ " which is isomorphic to " $\phi$  ( $^{\Box}$ Quine  $\phi$   $^{\Box}$ ))". - What's wrong is that self-reference with truth is impossible. In a particular technical sense, it doesn't terminate. Solution: Provability - Quining / self-referential provability sentence and provability implies truth - Curry-Howard, quines, abstract syntax trees (This is an interpreter!)

## A. Encoding with Add-Quote Function

```
module well-typed-syntax where
```

```
substTyp2-substTyp1-substTyp-weakenTyp : \forall \{\Gamma A B C T\} \{a : Teals \}
module lob-by-repr where
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        infixl 2 _⊳_
                          infix| 3 _''_
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        \mathsf{weakenTyp2}\text{-}\mathsf{weakenTyp1}: \forall \ \{\Gamma \ A \ B \ C \ D\} \xrightarrow{} \mathsf{Term} \ \{\Gamma \ \triangleright A \ \triangleright \ \mathsf{W} \ B \ \triangleright \mathsf{W} \ B \ \mathsf{W} \ B \
                          infix| 3 _"1_
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        weakenTyp1-weakenTyp : \forall \{\Gamma A B C\} \rightarrow \mathsf{Term} \{\Gamma \triangleright A \triangleright \mathsf{W} B\} (W
                          infix| 3 _"2_
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        \mathsf{weakenTyp1\text{-}weakenTyp\text{-}inv}: \forall \ \{\Gamma \ A \ B \ C\} \to \mathsf{Term} \ \{\Gamma \ \triangleright A \ \triangleright \ \mathsf{W} \ B\}
                          infix| 3 _"3_
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        weakenTyp1-weakenTyp1-weakenTyp : orall \ \{\Gamma\ A\ B\ C\ T\} 
ightarrow \mathsf{Term}\ \{\Gamma
                          infix| 3 _"a_
                        infixr 1 _'→'
infixl 3 _''''_
                        infix| 3 _w ""
                          infixr 1 _''→'.
                          \inf x 1 w" \rightarrow"
                          mutual
                                                        data Context : Set where
                                                                                     \epsilon: Context
                                                                                         \mathsf{data}\ \mathsf{Typ}: \mathsf{Context} \to \mathsf{Set}\ \mathsf{where}
                                                                                          \begin{array}{l} - : \forall \ \{\Gamma\ A\} \rightarrow \ \mathsf{lyp}\ (\Gamma\ \triangleright A) \rightarrow \ \mathsf{lerm}\ \{\Gamma\}\ A \rightarrow \ \mathsf{lyp}\ \Gamma \\ - : 1 : \exists \ \forall \ \{\Gamma\ AB\} \rightarrow (C: \mathsf{Typ}\ (\Gamma \triangleright A \triangleright B)) \rightarrow (a: \mathsf{Term}\ \{\Gamma\}\ A) \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright B) \rightarrow (B: \mathsf{Typ}\ \Gamma) \\ - : 2 : \exists \ \forall \ \{\Gamma\ AB\ C\} \rightarrow (D: \mathsf{Typ}\ (\Gamma \triangleright A \triangleright B \triangleright C)) \rightarrow (a: \mathsf{Term}\ \{\Gamma\}\ A) \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright B) \\ - : 3 : \exists \ \forall \ \{\Gamma\ AB\ C\ D\} \rightarrow (E: \mathsf{Typ}\ (\Gamma \triangleright A \triangleright B \triangleright C \triangleright D)) \rightarrow (a: \mathsf{Term}\ \{\Gamma\}\ A) \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright B) \\ - : 3 : \exists \ \forall \ \{\Gamma\ AB\ C\ D\} \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A \triangleright B \triangleright C \triangleright D)) \rightarrow (a: \mathsf{Term}\ \{\Gamma\}\ A) \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright B) \\ + \mathsf{Term}\ \{\Gamma\}\ A\} \rightarrow \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ \Gamma \rightarrow \mathsf{Typ}\ (\Gamma\ \triangleright A) \\ + \mathsf{Typ}\ (\Gamma\ \triangleright A) 
                                                                                 \begin{array}{l} \mathbb{W} : \forall \ \{\Gamma A\} \to \mathsf{Typ} \ (\Gamma \rhd A) \\ \mathbb{W} 1 : \forall \ \{\Gamma AB\} \to \mathsf{Typ} \ (\Gamma \rhd B) \to \mathsf{Typ} \ (\Gamma \rhd A \rhd (\mathbb{W} \ \{\Gamma = \Gamma\} \ \{A = A\} \ B) \} \\ \mathbb{W} 2 : \forall \ \{\Gamma ABC\} \to \mathsf{Typ} \ (\Gamma \rhd B \rhd C) \to \mathsf{Typ} \ (\Gamma \rhd A \rhd \mathbb{W} \ B \rhd \mathbb{W} 1 \ C) \\ \mathbb{W} 2 : \forall \ \{\Gamma ABC\} \to \mathsf{Typ} \ (\Gamma \rhd B \rhd C) \to \mathsf{Typ} \ (\Gamma \rhd A \rhd \mathbb{W} \ B \rhd \mathbb{W} 1 \ C) \\ \mathbb{W} 3 : \forall \ \{\Gamma ABC\} \to \mathsf{Typ} \ (\Gamma \rhd B \rhd C) \to \mathsf{Typ} \ (\Gamma \rhd A) \to \mathsf{Typ} \ (
                                                                                     W : \forall \{\Gamma A\} \rightarrow \mathsf{Typ} \ \Gamma \rightarrow \mathsf{Typ} \ (\Gamma \triangleright A)
                                                                                     \underbrace{\ \ '\to' \ : \forall \left\{\Gamma\right\} \left(A: \mathsf{Typ}\ \Gamma\right) \to \mathsf{Typ}\ \left(\Gamma \rhd A\right) \to \mathsf{Typ}\ \Gamma}_{\ \ \Sigma': \ \forall \left\{\Gamma\right\} \left(T: \mathsf{Typ}\ \Gamma\right) \to \mathsf{Typ}\ \left(\Gamma \rhd T\right) \to \mathsf{Typ}\ \Gamma}
                                                                                     \label{eq:context} \begin{tabular}{ll} \begi
                                                                                     \mathsf{'Term'}: \forall \ \{\Gamma\} \to \mathsf{Typ} \ (\Gamma \rhd \mathsf{'Context'} \rhd \mathsf{'Typ'})
                                                                              data Term : \forall \{\Gamma\} \rightarrow \mathsf{Typ} \ \Gamma \rightarrow \mathsf{Set} \ \mathsf{where}
                                                                                                     \begin{array}{l} {\color{red} {\color{blue} \land \bullet'}} : \forall \; \{ \Gamma A \, B \} \rightarrow \mathsf{Term} \; \{ \Gamma = (\Gamma \triangleright A) \} \; B \rightarrow \mathsf{Term} \; \{ \Gamma \} \; (A \; \stackrel{\cdot}{\rightarrow} \; B) \\ {\color{blue} {\color{blue} \land \circ \circ}} : \forall \; \{ \Gamma A \, B \} \rightarrow (f \colon \mathsf{Term} \; \{ \Gamma \} \; (A \; \stackrel{\cdot}{\rightarrow} \; B)) \rightarrow (x \colon \mathsf{Term} \; \{ \Gamma \} \; A) \rightarrow \overset{\mathsf{lerm}}{\longrightarrow} \overset{\mathsf{lerm}}{\longleftarrow} \underbrace{\{ \Gamma \triangleright T' \} \; (\mathsf{W2} \; (\mathsf{W} \; T) \; "_1 \; a \; "_b)} \\ {\color{blue} {\color{blue} \land \circ \circ \circ}} : \forall \; \{ \Gamma T' A \; B \; T \} \; \underbrace{\{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' A \; B \; T \} \; \{ \Gamma T' 
                                                                                     \overline{\mathsf{'VAR_0'}}: \forall \{\Gamma \ T\} \to \mathsf{Term} \{\Gamma = \Gamma \triangleright T\} \ (\mathsf{W} \ T)
                                                                                                                  _¬c : orall \{\Gamma\} 
ightarrow Context 
ightarrow Term \{\Gamma\} 'Context'
                                                                                   \ulcorner \neg \exists \text{ T} : \forall \ \{\Gamma \ \Gamma'\} \rightarrow \mathsf{Typ} \ \Gamma' \rightarrow \mathsf{Term} \ \{\Gamma\} \ (\text{`Typ' ''} \ \ulcorner \ \Gamma' \ \lnot c)
```

```
\mathsf{subst}\,\mathsf{Typ1\text{-}weaken}\,\mathsf{Typ1}:\forall\;\{\Gamma\,A\,B\,C\}\;\{a:\mathsf{Term}\;\{\Gamma\}\,A\}\to\mathsf{Term}\;\{
                                                                                                                                                                                                                                                                                                                                                                                                                                                     weakenTyp1-substTyp-weakenTyp1-inv : \forall \{\Gamma A T" T' T\} \{a : \mathsf{Tern}\}

ightarrow Term \{\Gamma 
hd T" 
hd W (T" "a)\} (W1 (W (T" a)))
                                                                                                                                                                                                                                                                                                                                                                                                                                                                      \rightarrow \mathsf{Term} \left\{ \Gamma \triangleright T" \triangleright \mathsf{W} \left( T' " a \right) \right\} \left( \mathsf{W1} \left( \mathsf{W} \ T"_1 \ a \right) \right)
                                                                                                                                                                                                                                                                                                                                                                                                                                                     weakenTyp1-substTyp-weakenTyp1 : \forall {\Gamma A T" T} {a : Term {\Gamma
                                                                                                                                                                                                                                                                                                                                                                                                                                                                     weakenTyp-substTyp-substTyp-weakenTyp1 : \forall \{\Gamma \ T' \ B \ A\} \ \{b : \mathsf{Te}\}
                                                                                                                                                                                                                                                                                                                                                                                                                                                                     weakenTyp-substTyp-weakenTyp1-inv : \forall \ \{\Gamma \ T' \ B \ A\} \ \{b \ \}
                                                                                                                                                                                                                                                                                                                                                                                                                                                                     weaken Typ-subst Typ2-subst Typ1-subst Typ-weaken Typ1 : \forall {\Gamma A B
                                                                                                                                                                                                                                                                                                                                                                                                                                                                     substTyp1-substTyp-tProd : \forall \{\Gamma TT'ABab\} \rightarrow \mathsf{Term} (( '\rightarrow '
                                                                                                                                                                                                                                                                                                                                                                                                                                                     subst\,Typ2-subst\,Typ-weaken\,Typ1-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Typ-weaken\,Ty
                                                                                                                                                                                                                                                                                                                                                                                                                                                                    \rightarrow \mathsf{Term} \left\{ \Gamma \triangleright T' \right\} \left( \mathsf{W1} \ T'' \ a \right)
                                                                                                                                                                                                                                                                                                                                                                                                                                                   weakenTyp-weakenTyp1-weakenTyp : \forall \ \{\Gamma \ A \ B \ C \ D\} 	o \ \mathsf{Term} \ \{\Gamma \ \mathsf{D} \ A \ B \ C \ D\}
 \begin{array}{c} \text{Tr} : \forall \ \{\Gamma\ \Gamma'\} \ \{T: \ \mathsf{Typ}\ \Gamma'\} \to \mathsf{Term}\ \{\Gamma\}\ (\exists\ \mathsf{Yp}^{\Gamma} \cap \Gamma', \exists \mathsf{C}) \\ \text{Tr} : \forall \ \{\Gamma\ \Gamma'\} \ \{T: \ \mathsf{Typ}\ \Gamma'\} \to \mathsf{Term}\ T \to \mathsf{Term}\ \{\Gamma\}\ (\exists\ \mathsf{Term}^{\Gamma}, \exists\ \mathsf{Term}
 2016/2/26
```

 $\mathsf{weakenTyp\text{-}tProd\text{-}inv}: \forall \; \{\Gamma \, A \, B \, C\} \to \mathsf{Term} \; \{\Gamma = \Gamma \rhd C\} \; (\mathsf{W} \, A \ \text{`--} \mathsf{V}) \; (\mathsf{W} \, A \ \mathsf{V}) \; = \mathsf{V} \; (\mathsf{W} \, A \ \mathsf{V}) \; (\mathsf{W} \, A \$ 

 $\mathsf{weakenTyp\text{-}weakenTyp\text{-}tProd}: \forall \ \{\Gamma \ A \ B \ C \ D\} \to \mathsf{Term} \ \{\Gamma \ \triangleright \ C \ \triangleright \ D \ A \ B \ C \ D\} \to \mathsf{Term} \ \{\Gamma \ \triangleright \ C \ \triangleright \ D \ A \ B \ C \ D\}$ 

 $\mathsf{subst}\,\mathsf{Typ1-tProd}:\forall\;\{\Gamma\;T\;T\;A\;B\}\;\{a:\mathsf{Term}\;\{\Gamma\}\;T\}\to\mathsf{Term}\;\{\Gamma\rhd T\}$ 

 $\mathsf{weakenTyp1-tProd}: \forall \ \{\Gamma \ \textit{CDAB}\} \to \mathsf{Term} \ \{\Gamma \ \triangleright \ \textit{C} \ \triangleright \ \mathsf{W} \ \textit{D}\} \ (\mathsf{W1} \ (\mathsf{W1} \ ))$ 

 $\mathsf{subst}\,\mathsf{Typ}\,\mathsf{2-t}\,\mathsf{P}\,\mathsf{rod}:\forall\;\{\Gamma\;T\;T\;T\;A\;B\}\;\{a:\mathsf{Term}\;\{\Gamma\}\;T\}\to\mathsf{Term}\;\{\Gamma\}$ 

weakenTyp-substTyp2-substTyp1-substTyp-weakenTyp-inv :  $\forall \{\Gamma\}$ 

 $\rightarrow$  Term  $\{\Gamma \triangleright T"'\}$   $((W(A''_2 a''_1 b'' c))' \rightarrow (W1(B''_3 a''_2 b))$ 

 $\rightarrow \mathsf{Term} \{\Gamma \triangleright T'\} (\mathsf{W} (T''_1 a'' b))$  $\rightarrow \mathsf{Term} \{ \Gamma \triangleright T' \} (\mathsf{W} (\mathsf{W} \ T "_2 \ a "_1 \ b " \ c))$ 

 $\rightarrow \mathsf{Term} \{ \Gamma \} (\mathsf{W} \ T^{"}_{2} \ a^{"}_{1} \ b^{"} \ c)$ 

 $\rightarrow \mathsf{Term} \{\Gamma \triangleright T"'\} (\mathsf{W} ((A \rightarrow B) a a a a b c))$ 

 $\rightarrow \mathsf{Term} \{ \Gamma \} (T''_1 \ a '' \ b)$ 

7

```
\begin{array}{l} \to \mathsf{Term} \; \{\epsilon\} \; (\text{`Typ' `'`} \; \Gamma \rhd A \; \urcorner c) \\ \to \mathsf{Term} \; \{\epsilon\} \; (\text{`Term' `'}_1 \; \Gamma \; \Gamma \; \urcorner c \; `' \; \Gamma \; A \; \urcorner T) \end{array}
                                                                                                                                                                                                                                                                                                                        \{b: \mathsf{Term}\; \{\epsilon\}\; (\mathit{T}\; ^{\cdot} \rightarrow ^{\cdot}\; \mathsf{W}\; (^{\cdot}\mathsf{Typ'}\; ^{\cdot}\; ^{\cdot}\; \vdash \epsilon \mathrel{\vartriangleright} B \; ^{\neg} \mathsf{c}))\}
                                                                                                                                                                                                                                                                                                                        \{c: \mathsf{Term} \ \{\varepsilon\} \ (T' \to \mathsf{'W} \ (\mathsf{`Term'} \ \mathsf{''}_1 \ \mathsf{E} \ \mathsf{"C} \ \mathsf{''} \ \mathsf{''} \ B \ \mathsf{"T}))\}
    \{v: \mathsf{Term} \{\varepsilon\} T\} \rightarrow
                                                                                                                                                                                                                                                                                                              (Term {ε} ('Term'''<sub>1</sub> Γε ¬c
         \to \overline{\operatorname{\mathsf{Term}}} \left\{ \varepsilon \triangleright X \right\} \left( W \left( \operatorname{\mathsf{Typ}}^{\prime} \cap \Gamma \triangleright A \operatorname{\mathsf{Tc}} \right) \right)
                                                                                                                                                                                                                                                                                                                         '' ((SW ((('\lambda \bullet' (SW (w \rightarrow b ''_a 'VAR_0') w'''' SW (w \rightarrow c ''_a 'V
           \rightarrow \text{Term } \{ \varepsilon \triangleright X \} \ (W \ (\text{'Term' '}_1 \ \Gamma \ \Gamma \ C \ (\text{``} \ A \ T))
                                                                                                                                                                                                                                                                                                                                             "
\rightarrow" (SW (b "_a v) "" SW <math>(c "_a v))))
            \to \mathsf{Term} \; \big\{ \epsilon \rhd X \big\} \; \big( \mathsf{W} \; \big( \mathsf{`Typ'} \; \mathsf{``} \; \Gamma \; \mathsf{\urcornerc} ) \big)
                                                                                                                                                                                                                                                                                                     s \leftarrow \leftarrow : \forall \{TB\}
        ``\to ```\_: \forall \ \{\Gamma\}
                                                                                                                                                                                                                                                                                                                        \{b: \bar{\mathsf{Term}}\ \{\epsilon\}\ (T\ \to\ \mathsf{W}\ (\mathsf{`Typ'}\ `\mathsf{``}\ \vdash \epsilon \rhd B\ \lnot\mathsf{c}))\}
     \overline{\phantom{a}} 
ightarrow \mathsf{Term} \left\{ arepsilon 
ight\} \left( \mathsf{``Typ'} \; \mathsf{``} \; \Gamma 
ight)
                                                                                                                                                                                                                                                                                                                          \{c: \mathsf{Term} \ \{\varepsilon\} \ (T' \to \mathsf{W} \ (\mathsf{`Term'} \ ''_1 \vdash \varepsilon \vdash c \ '' \vdash B \vdash \mathsf{T}))\}
            \rightarrow \mathsf{Term} \left\{ \epsilon \right\} \left( \mathsf{'Typ'} \; \mathsf{''} \; \Gamma \right)
                                                                                                                                                                                                                                                                                                                          \{v: \mathsf{Term} \{\varepsilon\} T\} \to \emptyset
            \rightarrow \mathsf{Term} \ \{\epsilon\} \ (\mathsf{`Typ'} \ \mathsf{``}\ \Gamma)
                                                                                                                                                                                                                                                                                                               (\text{Term } \{\epsilon\} \ (\text{`Term'} \, \text{`'}_1 \, \lceil \epsilon \, \rceil c)
   \mathsf{w}'' \to "" : \forall \{X \Gamma\}
                                                                                                                                                                                                                                                                                                                           ((\mathsf{SW}(b^{"}_a v)^{"})^{"}\mathsf{SW}(c^{"}_a v))
           \to \mathsf{Term}\; \{\epsilon \rhd \mathit{X}\}\; (\mathsf{W}\; (\mathsf{`Typ'}\; \mathsf{``}\; \Gamma))
                                                                                                                                                                                                                                                                                                                                              "
\rightarrow" (SW ((('\lambda \bullet' (SW (w \rightarrow b "_a 'VAR_0') w" SW (w \rightarrow b "_a 'VAR_0') w"

ightarrow Term \{ \epsilon \triangleright \mathit{X} \} (W ('Typ' '' \Gamma))
            \rightarrow \text{Term } \{ \varepsilon \triangleright X \} (W ('Typ' '' \Gamma))
                                                                                                                                                                                                                                                                   module well-typed-syntax-helpers where
\mathsf{w} \to : \forall \{\Gamma \land B \land C\} \to \mathsf{Term} (A \hookrightarrow \mathsf{W} B) \to \mathsf{Term} \{\Gamma = \Gamma \triangleright C\} (\mathsf{W}_{\mathsf{pAn}} + \mathsf{well} \mathsf{W}_{\mathsf{pAn}} + \mathsf{well} \mathsf{W}_{\mathsf{pAn}}) 
 {- things that were postulates, but are no longer -}
 "\rightarrow""\rightarrow "": \forall \{T'\}
                                                                                                                                                                                                                                                                             infix| 3 _'''a_
infixr 1 _'→''.
            {b : Term {ε} ('Typ' '' Γε ¬c)}
            \{c: \mathsf{Term}\, \{\varepsilon \triangleright T'\}\ (\mathsf{W}\ (\mathsf{`Typ'}^\mathsf{```} \ulcorner \varepsilon \urcorner \mathsf{c}))\}
                                                                                                                                                                                                                                                                              infix| 3 _'t'_
            \{e : \mathsf{Term} \{\epsilon\} T'\}
                                                                                                                                                                                                                                                                              infix| 3 _'t'1_
             	o Term \{arepsilon\} ('Term' ''_1 \ulcorner arepsilon \urcorner c '' (SW ('\lambda•' (c w''	o''' w b) ''_a in)f)x| 3 _'t'_2_
                      \rightarrow 'W ('Term' ^{\prime\prime}_{1} ^{\prime} \epsilon ^{\prime}c '' (SW ('\lambda \bullet' c ''_{a} e) ''\rightarrow''' b)))
                                                                                                                                                                                                                                                                             infixr 2 _'o'_
\mathsf{w}``\to```\to```:\forall\ \{T'\}
            \{b : \mathsf{Term} \{\epsilon\} \ (\mathsf{`Typ'} `\mathsf{``} \ \lceil \epsilon \ \rceil \mathsf{c})\}
                                                                                                                                                                                                                                                                              \mathsf{WS} \forall : \forall \ \{\Gamma \ T \ T' \ A \ B\} \ \{a : \mathsf{Term} \ \{\Gamma = \Gamma\} \ T\} \to \mathsf{Term} \ \{\Gamma = \Gamma \triangleright T'\} \ (\mathsf{W} ) = \mathsf{WS} \forall : \forall \ \{\Gamma \ T' \ A \ B\} \ \{A : \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} \} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T' \ A \ B\} = \mathsf{Term} \ \{\Gamma \ T'
            \{c : \mathsf{Term} \{ \varepsilon \triangleright T' \} (\mathsf{W} (\mathsf{`Typ'``} \vdash \varepsilon \lnot c)) \}
                                                                                                                                                                                                                                                                             WS \forall = weakenTyp-substTyp-tProd
\vdash \leftarrow '\lnot: \forall \{HX\} \rightarrow
                                                                                                                                                                                                                                                                                       \text{ `t'}\_: \forall \ \{\Gamma \ A\} \ \{B: \mathsf{Typ} \ (\Gamma \triangleright A)\} \rightarrow (b: \mathsf{Term} \ \{\Gamma = \Gamma \triangleright A\} \ B) \rightarrow (a: \mathsf{Top}) 
           \begin{array}{l} \mathsf{Term} \; \{ \widetilde{\epsilon} \} \; ( \, {}^{'}\!\mathsf{Term}^{'} \, {}^{''}_{1} \, {}^{\Gamma} \, \epsilon \, {}^{\Gamma} c \, {}^{"} \, ( \, {}^{\Gamma}\!\!H \, {}^{\sqcap}\!\mathsf{T} \, {}^{"} \! \to \! {}^{"}\!\!\! \mathsf{T} \, ( \, {}^{\Gamma}\!\!\! \mathsf{Term}^{'} \, {}^{"}_{1} \, {}^{\Gamma} \, \epsilon \, {}^{\Gamma} c \, {}^{"} \, {}^{\Gamma} \, H \, {}^{"} \! \to \! {}^{"}\!\!\! \mathsf{T} \, ( \, {}^{\Gamma}\!\!\!\! \mathsf{Term}^{'} \, {}^{"}_{1} \, {}^{\Gamma} \, \epsilon \, {}^{\Gamma} c \, {}^{"} \, {}^{\Gamma} \, H \, {}^{"} \! \to \! {}^{"}\!\!\!\! \mathsf{T} \, ( \, {}^{\Gamma}\!\!\!\!\! \mathsf{Term}^{'} \, {}^{"}_{1} \, {}^{\Gamma} \, \epsilon \, {}^{\Gamma} c \, {}^{"} \, {}^{"} \, H \, {}^{"} \!\!\!\! \to \! {}^{"}\!\!\!\! \mathsf{T} \, ( \, {}^{\Gamma}\!\!\!\!\!\!\! \mathsf{Term}^{'} \, {}^{"}_{1} \, {}^{\Gamma} \, \epsilon \, {}^{\Gamma} c \, {}^{"} \, {}^{"} \, H \, {}^{"} \!\!\!\!\! \to \! {}^{"}\!\!\!\!\!\!\!\! \mathsf{T} \, ( \, {}^{\Gamma}\!\!\!\!\!\!\!\!\!\!\!\!\!\!\mathsf{Term}^{'} \, {}^{"}_{1} \, {}^{"} \, {}^{\Gamma} \, h \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, {}^{"} \, 
                                                                                                                                                                                                                                                                              b 't' a = \lambda \bullet b' b''_a a
 \ulcorner \rightarrow `\urcorner : \forall \{HX\} \rightarrow
                                                                                                                                                                                                                                                                              \mathsf{substTyp\text{-}tProd}: \forall \ \{\Gamma \ TA \ B\} \ \{a: \mathsf{Term} \ \{\Gamma\} \ T\} \to
           Term \{\epsilon\} ('Term' ''<sub>1</sub> \lceil \epsilon \rceilc '' \lceil H \rightarrow WX \rceilT '\rightarrow W ('Term' ''<sub>1</sub> \lceil \epsilon \rceilc '' (\lceil H \rceilT ''\rightarrow W ('Term')))
                                                                                                                                                                                                                                                                             \begin{array}{l} \operatorname{\mathsf{Term}} \left\{ \Gamma \right\} \left( \left( A \stackrel{\cdot}{\rightarrow} \stackrel{\cdot}{\rightarrow} B \right) \stackrel{\cdot \cdot}{a} \right) \\ \rightarrow \operatorname{\mathsf{Term}} \left\{ \Gamma \right\} \left( \stackrel{\cdot}{\rightarrow} \stackrel{\cdot}{-} \left\{ \Gamma = \Gamma \right\} \left( A \stackrel{\cdot \cdot}{a} a \right) \left( B \stackrel{\cdot \cdot}{\rightarrow} a \right) \right) \\ \operatorname{\mathsf{substTyp-tProd}} \left\{ \Gamma \right\} \left\{ T \right\} \left\{ A \right\} \left\{ B \right\} \left\{ a \right\} x = \operatorname{\mathsf{SW}} \left( \left( \operatorname{\mathsf{WSV}} \right) \left( \operatorname{\mathsf{w}} x \right) \right) \stackrel{\mathsf{'t'}}{a} \right) \end{array}
 \text{``fcomp-nd''}: \forall \ \{A\ B\ C\} \rightarrow
          Term \{\varepsilon\} ('Term' "1 \Gamma \varepsilon \Gamma \varepsilon "\varepsilon " (A "\rightarrow"' C)

'\rightarrow' \forall ('Term' "1 \Gamma \varepsilon \Gamma \varepsilon "(C "\rightarrow"" B)

'\rightarrow' \forall ('Term' "1 \Gamma \varepsilon \Gamma \varepsilon "(C "\rightarrow"" B))))
                                                                                                                                                                                                                                                                             S \forall = substTyp-tProd
 \ulcorner ` \urcorner : orall \left\{ \mathit{B} \, A 
ight\} \left\{ \mathit{b} : \mathsf{Term} \left\{ arepsilon 
ight\} \right\} 
ightarrow
                                                                                                                                                                                                                                                                              \lambda' \bullet' : \forall \{\Gamma A B\} \to \mathsf{Term} \{\Gamma \triangleright A\} (\mathsf{W} B) -> \mathsf{Term} (A \to B)
            \begin{array}{c} \mathsf{Term} \left\{ \epsilon \right\} \left( \mathsf{`Term'} \; ``_1 \vdash \epsilon \; \urcorner \mathsf{c} \; `` \\ \left( \ulcorner A \; `' b \; \urcorner \mathsf{T} \; `` \to \urcorner \mathsf{T} \; `` \sqcap b \; \urcorner \mathsf{t} ) \right) \end{array} 
                                                                                                                                                                                                                                                                              \lambda' \bullet' f = \lambda \bullet' f
 \ulcorner ``\urcorner" : \forall \ \{\textit{B}\ \textit{A}\} \ \{\textit{b} : \mathsf{Term} \ \{\epsilon\} \ \textit{B}\} \rightarrow
                                                                                                                                                                                                                                                                              \mathsf{SW1V} : \forall \ \{\Gamma \ A \ T\} \to \mathsf{Term} \ \{\Gamma \ \triangleright A\} \ (\mathsf{W1} \ T \ `` \ `\mathsf{VAR}_0 \ ') \to \mathsf{Term} \ \{\Gamma \ \triangleright A\} \ ?
           Term \{\varepsilon\} ('Term' ''<sub>1</sub> \lceil \varepsilon \rceilc ''
(\lceil A \rceil T ''' \lceil b \rceil t '' \rightarrow '' \lceil A '' b \rceil T))
                                                                                                                                                                                                                                                                              SW1V = substTyp-weakenTyp1-VAR_0
  'cast-refl' : \forall \{T : \mathsf{Typ} \ (\varepsilon \triangleright `\Sigma' `\mathsf{Context'}' \mathsf{Typ}')\} \rightarrow
                                                                                                                                                                                                                                                                              Term \{\varepsilon\} ('Term'''<sub>1</sub> \lceil \varepsilon \rceilc''
                                                                                                                                                                                                                                                                             S_1 \forall = \text{substTyp1-tProd}
                      ((\lceil T \rceil) \text{ 'exist } T \rceil \lceil \varepsilon \triangleright \text{ '} \Sigma \text{ 'Context' 'Typ' } \rceil \text{c} \lceil T \rceil T \rceil T)
                                                                                                                                                                                                                                                                              \mathsf{un}\, `\lambda \bullet ' : \forall \ \{\Gamma \, A \, B\} \to \mathsf{Term} \ (A \ `\to ' B) \to \mathsf{Term} \ \{\Gamma \triangleright A\} \ B
                                 (\mathsf{SW}\ (\text{`cast'}\ ''_a\ \text{`exist}\ \mathsf{T'}\ \ulcorner \ \epsilon \rhd \ `\Sigma'\ \text{`Context'}\ \mathsf{`Typ'}\ \urcorner \mathsf{c}\ \ulcorner \ T\ \urcorner \mathsf{Tu})\mathsf{n}\ `\lambda \bullet'\ f = \mathsf{SW1V}\ (\mathsf{weaken}\ \mathsf{Typ-tProd}\ (\mathsf{w}\ f)\ ''_a\ \mathsf{`VAR}_0')
                                           '''' SW ('quote-sigma' ''_a 'existT' \lceil \varepsilon \triangleright '\Sigma' 'Context' 'Typ' \lceil c \lceil T \rceil T))))
  \text{`cast-refl''}: \forall \ \{T\colon \mathsf{Typ}\ (\epsilon \,{\triangleright}\,\, `\Sigma' \,\, \mathsf{`Context'}\,\, \mathsf{`Typ'})\} \rightarrow
                                                                                                                                                                                                                                                                             weakenProd : \forall \{\Gamma A B C\} \rightarrow
            Term \{\epsilon\} ('Term' ''<sub>1</sub> \lceil \epsilon \rceilc''
                                                                                                                                                                                                                                                                                         Term \{\Gamma\} (A \rightarrow B)
                       ((SW ('cast' ''_a 'existT' \lceil \varepsilon \triangleright '\Sigma' 'Context' 'Typ' \rceil c \lceil T \rceil T))
                                                                                                                                                                                                                                                                                       \rightarrow \mathsf{Term} \{ \Gamma = \Gamma \triangleright C \} (\mathsf{W} A ' \rightarrow \mathsf{W} 1 B)
                                           \text{SW ('quote-sigma' ''}_a \text{ 'existT' } \vdash \epsilon \vartriangleright \text{ '} \Sigma' \text{ 'Context' '} \text{ 'TypekenP} \text{ id} \text{ T})) } \{A\} \{B\} \{C\} \ x = \text{weakenTyp-tProd (w } x) \} 
                                                                                                                                                                                                                                                                              w\forall = weakenProd
                                 (\ulcorner T \text{ `` `existT'}  \ulcorner \epsilon \rhd `\Sigma' \text{ `Context' `Typ'}  \urcorner c  \ulcorner T  \urcorner T  \urcorner T)))
 \forall s \rightarrow \rightarrow : \forall \{TB\}
                                                                                                                                                                                                                                                                             \mathsf{w1}: \forall \ \{\Gamma \ A \ B \ C\} \to \mathsf{Term} \ \{\Gamma = \Gamma \triangleright B\} \ C \to \mathsf{Term} \ \{\Gamma = \Gamma \triangleright A \triangleright \mathsf{W} \ \{\Gamma = \Gamma \triangleright A \triangleright \mathsf{W} \}
```

```
w1 x = un'\lambda \bullet' (weakenTyp-tProd (w ('\lambda \bullet' x)))
                                                                                                                                                                                                                                                                                                                                                                                                                            weakenProd-nd-Prod-nd : \forall \{\Gamma A B C D\} \rightarrow
                                                                                                                                                                                                                                                                                                                                                                                                                                             Term (A \hookrightarrow B \hookrightarrow C)
              \text{'t'}_1 \quad : \forall \ \{\Gamma \ A \ B \ C\} \rightarrow (c : \mathsf{Term} \ \{\Gamma = \Gamma \rhd A \rhd B\} \ C) \rightarrow (a : \mathsf{Term} \ \{\Gamma\} \ A) \ \ \mathsf{Ter} \ \mathsf{Ther} \ \mathsf{Th
\overline{f}'t'<sub>1</sub>\overline{x} = un'\lambda \bullet' (S\forall ('\lambda \bullet' ('\lambda \bullet' f) ''_a x))
                                                                                                                                                                                                                                                                                                                                                                                                                             \text{weakenProd-nd-Prod-nd} \ \{\Gamma\} \ \{A\} \ \{B\} \ \{C\} \ \{D\} \ x = \text{weakenTyp-tProd-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-Prod-nd-P
            \text{ ``t'}_{2} : \forall \left\{ \Gamma A \ B \ C \ D \right\} \rightarrow \left( c : \mathsf{Term} \left\{ \Gamma = \Gamma \triangleright A \triangleright B \triangleright C \right\} D \right) \rightarrow \left( a : \mathsf{Twrm} \left\{ \Gamma \right\} A \right) \text{ each eller like the like of definition } a \triangleright C \text{ ''}_{1} \ a \right\} \left( D \text{ ''}_{2} \ a \right) 
 f't'_2 \overline{x} = un'\lambda \bullet' (S_1 \forall (un'\lambda \bullet' (S \forall ('\lambda \bullet' ('\lambda \bullet' ('\lambda \bullet' f)) ''_a x))))
                                                                                                                                                                                                                                                                                                                                                                                                                              \mathsf{S}_1\mathsf{W}1: \forall \ \{\Gamma\ A\ B\ C\}\ \{a: \mathsf{Term}\ \{\Gamma\}\ A\} \to \mathsf{Term}\ \{\Gamma\ \triangleright \ \mathsf{W}\ B\ ``\ a\}\ (\mathsf{W}1\ C\ ``
  S_{10}W' = substTyp1-substTyp-weakenTyp-inv
  \mathsf{S}_{10}\mathsf{W}:\forall \left\{\Gamma \ C \ TA\right\} \left\{a: \mathsf{Term} \left\{\Gamma\right\} \ C\right\} \left\{b: \mathsf{Term} \left\{\Gamma\right\} \left(T'' \ a\right)\right\} \to \mathsf{Term} \\ \mathsf{V}[\mathsf{S}_{1}] \mathsf{W}: \forall \left\{\Gamma \ C \ TA\right\} \left\{a: \mathsf{Term} \left\{\Gamma\right\} \ C\right\} \left\{b: \mathsf{Term} \left\{\Gamma\right\} \left(T'' \ a\right)\right\} \to \mathsf{Term} \\ \mathsf{V}[\mathsf{S}_{1}] \mathsf{W}: \forall \left\{\Gamma \ C \ TA\right\} \left\{a: \mathsf{Term} \left\{\Gamma\right\} \ C\right\} \left\{b: \mathsf{Term} \left\{\Gamma\right\} \left(T'' \ a\right)\right\} \to \mathsf{Term} \\ \mathsf{V}[\mathsf{S}_{1}] \mathsf{W}: \forall \left\{\Gamma \ C \ TA\right\} \left\{a: \mathsf{Term} \left\{\Gamma\right\} \ C\right\} \left\{b: \mathsf{Term} \left\{\Gamma\right\} \left(T'' \ a\right)\right\} \to \mathsf{Term} \\ \mathsf{V}[\mathsf{S}_{1}] \mathsf{W}: \mathsf{
                                                                                                                                                                                                                                                                                                                                                                                                                                             \rightarrow \text{Term} \{\Gamma \triangleright T" \triangleright W (T'' a)\} (W1 (W (T'' a)))
  S_{10}W = substTyp1-substTyp-weakenTyp
                                                                                                                                                                                                                                                                                                                                                                                                                                             \rightarrow \text{Term } \{\Gamma \triangleright T" \triangleright W (T'' a)\} (W1 (W T''_1 a))
  \operatorname{\mathsf{subst}\mathsf{Typ}1-\mathsf{subst}\mathsf{Typ-weaken}\mathsf{Typ-weaken}\mathsf{Typ}} : \forall \{\Gamma TA\} \{B : \operatorname{\mathsf{Typ}}(\Gamma \bowtie \mathsf{W}1)\}_1 W' = \operatorname{\mathsf{weaken}\mathsf{Typ}1-\mathsf{subst}\mathsf{Typ-weaken}\mathsf{Typ}1-\mathsf{inv}}
                 \rightarrow \{a : \mathsf{Term} \{\Gamma\} A\}

ightarrow \{b : \mathsf{Term}\ \{\Gamma\}\ (B\ ``\ a)\}
                 \rightarrow \text{Term } \{\Gamma\} \ (\text{W } (\text{W } T) \text{ "}_1 \ a \text{ "} b)
                                                                                                                                                                                                                                                                                                                                                                                                                              substTyp-weakenTyp1-inv: \forall \{\Gamma A T' T\}
                  \rightarrow \mathsf{Term} \ \{\Gamma\} \ T
                                                                                                                                                                                                                                                                                                                                                                                                                                             \{a: \mathsf{Term}\ \{\Gamma\}\, A\} \to
                                                                                                                                                                                                                                                                                                                                                                                                                                             Term \{\Gamma = (\Gamma \triangleright T' \cap a)\}\ (W(T' a))
  subst Typ1-subst Typ-weaken Typ-weaken Typ x = SW(S_{10}Wx)
                                                                                                                                                                                                                                                                                                                                                                                                                                             \rightarrow \text{Term } \{\Gamma = (\Gamma \triangleright T', i', a)\} (W T', a)
                                                                                                                                                                                                                                                                                                                                                                                                                              \mathsf{substTyp\text{-}weakenTyp1\text{-}inv}\ \{\mathsf{a}=a\}\ x=\mathsf{S}_1\mathsf{W1}\ (\mathsf{W1S}_1\mathsf{W'}\ (\mathsf{w1}\ x)\ \mathsf{`t'}_1\ a)
  S_{10}WW = substTyp1-substTyp-weakenTyp-weakenTyp
                                                                                                                                                                                                                                                                                                                                                                                                                              S_1W' = substTyp-weakenTyp1-inv
  \mathsf{S}_{210}\mathsf{W}:\forall\left\{\Gamma\,A\,B\,C\,T\right\}\left\{a:\mathsf{Term}\left\{\Gamma\right\}A\right\}\left\{b:\mathsf{Term}\left\{\Gamma\right\}\left(B^{\,\,\prime\prime}\,a\right)\right\}\left\{c:\mathsf{Term}\left\{\Gamma\right\}\left(C^{\,\,\prime\prime}\,\,a^{\,\,\prime\prime}\,\,b\right)\right\}
                                                                                                                                                                                                                                                                                                                                                                                                                                \_ 'o' \_ : \forall \{\Gamma A B C\}

ightarrow Term \{\Gamma\} (W T ^{\prime\prime}{}_{2} a ^{\prime\prime}{}_{1} b ^{\prime\prime} c)
                                                                                                                                                                                                                                                                                                                                                                                                                                            \rightarrow \mathsf{Term} \{\Gamma\} (A \xrightarrow{\epsilon} B)
                 \rightarrow \mathsf{Term} \{\Gamma\} (T "_1 a " b)
                                                                                                                                                                                                                                                                                                                                                                                                                                            \rightarrow \operatorname{\underline{\mathsf{Term}}} \{\Gamma\} (B' \rightarrow C')
  S_{210}W = substTyp2-substTyp1-substTyp-weakenTyp
                                                                                                                                                                                                                                                                                                                                                                                                                                             \rightarrow \mathsf{Term} \{\Gamma\} (A' \rightarrow C')
  \mathsf{substTyp2-substTyp1-substTyp-weakenTyp-weakenTyp}: \forall \left\{ \Gamma \textit{ABCT} \right\} \\ \circ' f = \lambda \bullet' (\mathsf{w} \rightarrow f''' \ a \ (\mathsf{w} \rightarrow g \ ''' \ a' \ \mathsf{VAR}_0'))
                  \{a: \mathsf{Term} \{\Gamma\} A\}
                  \begin{cases} b : \mathsf{Term} \ \{\Gamma\} \ (B \ '' \ a) \} \\ \{c : \mathsf{Term} \ \{\Gamma\} \ (C \ ''_1 \ a \ '' \ b) \} \to \\ \mathsf{Term} \ \{\Gamma\} \ (W \ (W \ T) \ ''_2 \ a \ ''_1 \ b \ '' \ c) \end{cases} 
                                                                                                                                                                                                                                                                                                                                                                                                                              \mathsf{WS}_{00}\mathsf{W1}: \forall \{\Gamma \ T' \ B \ A\} \ \{b: \mathsf{Term} \ \{\Gamma\} \ B\} \ \{a: \mathsf{Term} \ \{\Gamma \rhd B\} \ (\mathsf{W} \ A)\} \ \{T' \ B \ A\} \ \{T' \ B 
                                                                                                                                                                                                                                                                                                                                                                                                                                            \rightarrow \mathsf{Term} \{\Gamma\} (T " a)
  substTyp2-substTyp1-substTyp-weakenTyp_x = S_{10}W (S_2W_0W_0W_1 = weakenTyp-substTyp-substTyp-weakenTyp1
 \mathsf{S}_{210}\mathsf{WW} = \mathsf{substTyp2\text{-}substTyp1\text{-}substTyp-weakenTyp-weakenTyp-weakenTyp} \quad \mathsf{WS}_{00}\mathsf{W1'} : \forall \left\{\Gamma \ T' \ B \ A\right\} \ \left\{b : \mathsf{Term} \ \left\{\Gamma\right\} \ B\right\} \left\{a : \mathsf{Term} \ \left\{\Gamma \rhd B\right\} \ (\mathsf{W} \ A)\right\} \left\{T \ \mathsf{W} \ \left\{T'' \ \mathsf{W} \ \left\{T''' \ \mathsf{SW} \ \left\{a''' \ \mathsf{W} \ \mathsf{W} \right\}\right\}\right\} \left\{T \ \mathsf{W} \ \mathsf{W} \ \mathsf{W} \right\} \left\{T \ \mathsf{W} \ \mathsf{W} \right\} \left\{T \ \mathsf{W} \ \mathsf{W} \right\} \left
  W1W = weakenTyp1-weakenTyp
                                                                                                                                                                                                                                                                                                                                                                                                                            WS_{00}W1' = weakenTyp-substTyp-substTyp-weakenTyp1-inv
  W1W1W = weakenTyp1-weakenTyp1-weakenTyp
                                                                                                                                                                                                                                                                                                                                                                                                                                             \{b : \mathsf{Term} \{\Gamma\} B\}
                                                                                                                                                                                                                                                                                                                                                                                                                                               \{a: \mathsf{Term}\ \{\Gamma \triangleright B\}\ (\mathsf{W}\ A)\}
  weaken Typ-t Prod-nd : \forall \{\Gamma A B C\} \rightarrow
                                                                                                                                                                                                                                                                                                                                                                                                                                              \{T: \mathsf{Typ}\,(\Gamma \triangleright A)\}
                 Term \{\Gamma = \Gamma \triangleright C\} (W (A \rightarrow B))
                                                                                                                                                                                                                                                                                                                                                                                                                                              \{X\} \rightarrow
                  \rightarrow \operatorname{\mathsf{Term}} \{\Gamma = \Gamma \triangleright \hat{C}\} \ (\stackrel{\circ}{\mathsf{W}} A \stackrel{\circ}{\rightarrow} \stackrel{\circ}{\mathsf{W}} B)
                                                                                                                                                                                                                                                                                                                                                                                                                                             Term \{\Gamma\} (T'') (SW(a't'b)) \rightarrow X
  weaken Typ-t Prod-nd x = \lambda' \bullet' (W1W (SW1V (weaken Typ-t Prod (w (weaken Typ-t Prod (weaken Typ-t Pro
                                                                                                                                                                                                                                                                                                                                                                                                                            substTyp-substTyp-weakenTyp1-inv-arr x = \lambda \bullet' (w \rightarrow x'''_a WS_{00}W1')
  \mathsf{weakenProd}\mathsf{-nd}:\forall\;\{\Gamma\,A\,B\,C\}\to
                 Term (A \rightarrow B)
                                                                                                                                                                                                                                                                                                                                                                                                                              S_{00}W1' \rightarrow = substTyp-substTyp-weakenTyp1-inv-arr
                  \rightarrow \mathsf{Term} \ \{\Gamma = \Gamma \triangleright C\} \ (\mathsf{W} \ A \ ' \rightarrow `` \ \mathsf{W} \ B)
  \mathsf{weakenProd-nd} \ \{\Gamma\} \ \{A\} \ \{B\} \ \{C\} \ x = \mathsf{weakenTyp-tProd-nd} \ (\mathsf{w} \ x)
                                                                                                                                                                                                                                                                                                                                                                                                                              \mathsf{subst}\,\mathsf{Typ}\text{-}\mathsf{subst}\,\mathsf{Typ}\text{-}\mathsf{weaken}\,\mathsf{Typ}\mathsf{1}\text{-}\mathsf{arr}\text{-}\mathsf{inv}: \forall\;\{\Gamma\,B\,A\}
                                                                                                                                                                                                                                                                                                                                                                                                                                             \{b: \mathsf{Term}\ \{\Gamma\}\ B\}
                                                                                                                                                                                                                                                                                                                                                                                                                                             \{a: \mathsf{Term} \ \{\Gamma \rhd B\} \ (\mathsf{W} \ A)\}
                                                                                                                                                                                                                                                                                                                                                                                                                                             \{T: \mathsf{Typ} (\Gamma \triangleright A)\}
                                                                                                                                                                                                                                                                                                                                                                                                                                             \{X\} \rightarrow
                                                                                                                                                                                                                                                                                                                                                                                                                                             Term \{\Gamma\} (X' \rightarrow '' T'' (SW (a 't' b)))
 weakenTyp-tProd-nd-tProd-nd: \forall \{\Gamma A B C D\} \rightarrow
                                                                                                                                                                                                                                                                                                                                                                                                                                             \rightarrow \text{Term } \{ \widehat{\Gamma} \} (X' \rightarrow " \widehat{W} 1 T'' a " b)
                  Term \{\Gamma = \Gamma \triangleright D\} (W (A \rightarrow B \rightarrow C))
                   \rightarrow \mathsf{Term} \ \{\Gamma = \Gamma \triangleright D\} \ (\overset{\cdot}{\mathsf{W}} A \overset{\cdot}{\rightarrow} \overset{\cdot}{\mathrel{'}} \overset{\cdot}{\mathsf{W}} B \overset{\cdot}{\rightarrow} \overset{\cdot}{\mathrel{'}} \overset{\cdot}{\mathsf{W}} C)
                                                                                                                                                                                                                                                                                                                                                                                                                            substTyp-substTyp-weakenTyp1-arr-inv x = \lambda \bullet' (WS_{00}W1' (un'\lambda \bullet' x))
  weakenTyp-tProd-nd-tProd-nd x = \lambda \bullet (weakenTyp-tProd-inv (\lambda \bullet) (W1W1W (SW1V (w\forall (weakenTyp-tProd (weakenTyp-weakenTyp-tProd))
                                                                                                                                                                                                                                                                                                                                                                                                                              S_{00}W1' \leftarrow = substTyp-substTyp-weakenTyp1-arr-inv
```

```
module well-typed-quoted-syntax where
                                                                                                                                                                                                          open well-typed-syntax
                                                                                                                                                                                                          open well-typed-syntax-helpers public
      substTyp-substTyp-weakenTyp1: \forall \{\Gamma B A\}
              \{b : \mathsf{Term} \{\Gamma\} B\}
                                                                                                                                                                                                          open well-typed-quoted-syntax-defs public
                                                                                                                                                                                                          open well-typed-syntax-context-helpers public
              \{a: \mathsf{Term} \{\Gamma \triangleright B\} \ (\mathsf{W} \ A)\}
              \{T \colon \mathsf{Typ}\; (\Gamma \triangleright A)\} \to
                                                                                                                                                                                                           open well-typed-syntax-eq-dec public
              Term \{\Gamma\} (W1 T'' a'' b)
               \rightarrow \mathsf{Term} \{ \widehat{\Gamma} \} (T \text{ "} (\mathsf{SW} (a \text{ 't'} b)))
                                                                                                                                                                                                          infixr 2 _"o"_
      substTyp-substTyp-weakenTyp1 x = (SW (WS_{00}W1 (w x) 't' x))
      S_{00}W1 = substTyp-substTyp-weakenTyp1
                                                                                                                                                                                                           quote-sigma : (\Gamma v : \Sigma \text{ Context Typ}) \rightarrow \text{Term } \{\epsilon\} \ (`\Sigma' `\text{Context' `Typ'})
                                                                                                                                                                                                           \mathsf{quote}\text{-}\mathsf{sigma}\;(\Gamma\;,\,v)=\,\mathsf{`exist}\mathsf{T'}\,\,\ulcorner\,\Gamma\,\,\urcorner\mathsf{c}\,\,\ulcorner\,v\,\,\urcorner\mathsf{T}
      \mathsf{SW1W} : \forall \{\Gamma \ T\} \{A : \mathsf{Typ} \ \Gamma\} \{B : \mathsf{Typ} \ \Gamma\}
              \rightarrow \{a : \mathsf{Term} \{\Gamma = \Gamma \triangleright T\} \ (\mathsf{W} \{\Gamma = \Gamma\} \{\mathsf{A} = T\} B)\}
                                                                                                                                                                                                               "o"_: ∀ {A B C}
                                                                                                                                                                                                                A \to \overline{\square} (`\square'`` (C ``\to ``` B))
              \rightarrow \mathsf{Term} \{ \Gamma = \Gamma \triangleright T \} (\mathsf{W1} (\mathsf{W} A) " a)
                                                                                                                                                                                                                  \rightarrow \Box (\Box'\Box' \Box' (A \Box \rightarrow \Box'))
              \rightarrow \mathsf{Term} \{ \Gamma = \Gamma \triangleright T \} (\mathsf{W} A)
                                                                                                                                                                                                                  \rightarrow \Box (\Box' \Box' \Box' (A \cdots \rightarrow \Box' B))
       SW1W = substTyp-weakenTyp1-weakenTyp
                                                                                                                                                                                                           g "o" f = ("fcomp-nd" "" a f" a g)
      S_{200}W1WW: \forall {\GammaA} {T: Typ (\GammaDA)} {T' CB} {a: Term {\Gamma}A} {\alpha: Term {\Gamma
              \{c : \mathsf{Term} \ \{\Gamma = (\Gamma \triangleright T')\} \ (\mathsf{W} \ (C'' \ a))\}
                                                                                                                                                                                                                  Term \{\Gamma = (\varepsilon \triangleright '\Box' '' qH0)\}
                                                                                                                                                                                                                        (\mathsf{W}(\mathsf{'}\Box\mathsf{'}\,\mathsf{'})^\mathsf{r}\,\mathsf{'}\Box\mathsf{'}\,\mathsf{'})^\mathsf{r}\,\mathsf{'}\mathsf{q}H0\,\mathsf{'}\overset{\mathsf{r}}{\to}\mathsf{'})^\mathsf{r}\,\mathsf{q}X\,\mathsf{T}))
              \rightarrow Term \{\Gamma = (\Gamma \triangleright T')\} (W1 (W (W T) ^{\prime\prime}_{2} a ^{\prime\prime} b) ^{\prime\prime} c)
              \rightarrow \mathsf{Term} \{ \Gamma = (\Gamma \triangleright T') \} (\mathsf{W} (T'' a))
                                                                                                                                                                                                                  \rightarrow \mathsf{Term} \{ \Gamma = (\varepsilon \triangleright '\Box' '' qH0) \}
      S_{200}W1WW = \text{substTyp2-substTyp-substTyp-weakenTyp1-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp-weakenTyp
                                                                                                                                                                                                                                 (`\Box'``(\ulcorner`\Box'``qH0\urcorner\top``\rightarrow```\ulcorner qX\urcorner\top)))
                                                                                                                                                                                                           Conv0 \{qH0\} \{qX\} x = w \rightarrow \neg \gamma' '''_a x
      S_{10}W2W : \forall \{\Gamma T' A B T\} \{a : \text{Term } \{\Gamma \triangleright T'\} (W A)\} \{b : \text{Term } \{\Gamma \triangleright T'\} (W1 B'' a)\}
              module well-typed-syntax-pre-interpreter where
                                                                                                                                                                                                          open well-typed-syntax
      S_{10}W2W = substTyp1-substTyp-weakenTyp2-weakenTyp
                                                                                                                                                                                                           open well-typed-syntax-helpers
module well-typed-syntax-context-helpers where
      open well-typed-syntax
                                                                                                                                                                                                           max-level : Level
      open well-typed-syntax-helpers
                                                                                                                                                                                                           max-level = |suc |zero
      \begin{array}{l} \square\_ : \mathsf{Typ}\: \epsilon \to \mathsf{Set} \\ \square\_\: T = \mathsf{Term}\: \{\Gamma = \epsilon\}\: T \end{array}
                                                                                                                                                                                                           module inner
                                                                                                                                                                                                                  (context-pick-if': \forall \ \ell \ (P: Context \rightarrow Set \ \ell)
module well-typed-quoted-syntax-defs where
                                                                                                                                                                                                                          (\Gamma : \mathsf{Context})
                                                                                                                                                                                                                          (dummy : P(\epsilon \triangleright `\Sigma' `Context' `Typ'))
      open well-typed-syntax
      open well-typed-syntax-helpers
                                                                                                                                                                                                                          (val : P \Gamma) \rightarrow
                                                                                                                                                                                                           P(\varepsilon \triangleright '\Sigma' 'Context' 'Typ'))
      open well-typed-syntax-context-helpers
                                                                                                                                                                                                                  (context-pick-if-refl' : \forall \ \ell \ P \ dummy \ val \rightarrow
                                                                                                                                                                                                                         context-pick-if' \ell P (\epsilon \triangleright `\Sigma' `Context' `Typ') <math>dummy \ val \equiv val)
      ε': Term {\Gamma = ε} 'Context'
      `\epsilon' = \ulcorner \, \epsilon \, \, \urcorner c
                                                                                                                                                                                                                  where
      `\Box\textrm{'}:\mathsf{Typ}\;(\epsilon\rhd\textrm{`Typ'}\;\textrm{`'}\;\epsilon\textrm{'})
                                                                                                                                                                                                                  context-pick-if: \forall \{\ell\} \{P : \mathsf{Context} \to \mathsf{Set} \ \ell\}
      '\square' = 'Term' "_1 '\varepsilon'
                                                                                                                                                                                                                          \{\Gamma:\mathsf{Context}\}
                                                                                                                                                                                                                          (dummy : P (\varepsilon \triangleright '\Sigma' 'Context' 'Typ'))
                                                                                                                                                                                                                          (val : P \Gamma) \rightarrow
module well-typed-syntax-eq-dec where
                                                                                                                                                                                                          P(\varepsilon \triangleright '\Sigma' 'Context' 'Typ')
      open well-typed-syntax
                                                                                                                                                                                                                  context-pick-if \{P = P\} dummy val = context-pick-if' P dummy val
                                                                                                                                                                                                                  context-pick-if-refl : \forall \{\ell \ P \ dummy \ val\} \rightarrow
      context-pick-if: \forall \{\ell\} \{P : \mathsf{Context} \to \mathsf{Set} \ \ell\}
                                                                                                                                                                                                                          context-pick-if \{\ell\} \{P\} \{\varepsilon \triangleright `\Sigma' `Context' `Typ'\} dummy <math>val \equiv val
              \{\Gamma:\mathsf{Context}\}
              (dummy : P(\varepsilon \triangleright '\Sigma' 'Context' 'Typ'))
                                                                                                                                                                                                                  context-pick-if-refl \{P = P\} = context-pick-if-refl' P _ _ 
              (val : P \Gamma) \rightarrow
      P (\varepsilon \triangleright '\Sigma' 'Context' 'Typ')
                                                                                                                                                                                                                  private
      context-pick-if \{P = P\} \{\varepsilon \triangleright `\Sigma' `Context' `Typ'\} dummy val = val
                                                                                                                                                                                                                          dummy: Typ \epsilon
      context-pick-if \{P = P\} \{\Gamma\} dummy \ val = dummy
                                                                                                                                                                                                                          dummy = 'Context'
      context-pick-if-refl : \forall \{\ell \ P \ dummy \ val\} \rightarrow
                                                                                                                                                                                                                  cast-helper: \forall \{X T A\} \{x : \mathsf{Term} X\} \to A \equiv T \to \mathsf{Term} \{\varepsilon\} (T " x \to T)
              context-pick-if \{\ell\} \{P\} \{\varepsilon \triangleright `\Sigma' `Context' `Typ'\} dummy <math>val \equiv val
                                                                                                                                                                                                                  cast-helper refl = (\lambda \bullet) 'VAR<sub>0</sub>'
      context-pick-if-refl \{P = P\} = refl
                                                                                                                                                                                                                  cast'-proof : \forall \{T\} \rightarrow \mathsf{Term} \{\epsilon\}  (context-pick-if \{P = \mathsf{Typ}\} (W dumr
                                                                                                                                                                                                                          '\rightarrow'' T'' 'exist T' \lceil \varepsilon \rhd '\Sigma'  'Context' 'Typ' \rceil c \lceil T \rceil T)
```

10 2016/2/26

```
\mathsf{cast'-proof}\ \{T\} = \mathsf{cast-helper}\ \{`\Sigma'\ `\mathsf{Context'}\ `\mathsf{Typ'}\}
                                                                                                                                                                                                                                                                                                                                                                                                                                         Term\Downarrow (weaken Typ-subst Typ2-subst Typ1-subst Typ-t Prod t) \Gamma \Downarrow T
               \{\text{context-pick-if } \{P = \mathsf{Typ}\} \{\varepsilon \triangleright `\Sigma' `\mathsf{Context'} `\mathsf{Typ'}\} (\mathsf{W} \mathsf{dummy}) T\} \mathsf{Term} \ \mathsf{W} \ \mathsf{weakenTyp2-weakenTyp1} t) \Gamma \ \mathsf{T} = \mathsf{Term} \ \mathsf{U} \ \mathsf{T} \ \mathsf{T
               \{T\} (sym (context-pick-if-refl \{P = Typ\} \{dummy = W dummy\}))
                                                                                                                                                                                                                                                                                                                                                                                                                                         \mathsf{Term} \Downarrow (\mathsf{weakenTyp1-weakenTyp}\ t)\ \Gamma \Downarrow = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                                                                         \mathsf{Term} \Downarrow (\mathsf{weakenTyp1\text{-}weakenTyp\text{-}inv}\ t)\ \Gamma \Downarrow = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow
\mathsf{cast}\text{-}\mathsf{proof}: \forall \ \{T\} \to \mathsf{Term} \ \{\epsilon\} \ (T'' \ \text{`existT'} \ \ulcorner \ \epsilon \, \vartriangleright \ \text{`$\Sigma'$ 'Context' 'Typ'} \ \urcorner \mathsf{c} \ \mathsf{Term} \ (\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}\text{-}\mathsf{weakenTyp1}
                 '→'' context-pick-if \{P = \mathsf{Typ}\} (\mathsf{W} dummy) T'' 'exist \mathsf{T}' \vdash \varepsilon \rhd '\Sigma' 'Context \mathsf{U} (\mathsf{Typ} b sto \mathsf{Typ}) \mathsf{M} en \mathsf{Typ}1 t) \mathsf{T} \Downarrow = \mathsf{Term} \Downarrow t \mathsf{T} \Downarrow
cast-proof \{T\} = cast-helper \{'\Sigma' 'Context' 'Typ'\} \{T\}
                                                                                                                                                                                                                                                                                                                                                                                                                                          Term\downarrow (weakenTyp1-substTyp-weakenTyp1-inv t) \Gamma \Downarrow = \text{Term} \Downarrow t \Gamma
                \{\mathsf{context\text{-}pick\text{-}if}\ \{\mathsf{P} = \mathsf{Typ}\}\ \{\epsilon \, \triangleright \, `\Sigma' \, `\mathsf{Context'}\,\, `\mathsf{Typ'}\} \, (\mathsf{W}\,\, \mathsf{dummy}) \, \mathit{T}\} \,\, \mathsf{Term} \, \downarrow \, (\mathsf{weaken}\, \mathsf{Typ}1\text{-}\mathsf{subst}\, \mathsf{Typ\text{-}weaken}\, \mathsf{Typ}1 \, \mathit{t}) \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathit{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathit{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathit{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathsf{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathsf{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathsf{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathsf{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathsf{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathsf{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathsf{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathsf{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathsf{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathsf{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \, \downarrow \, \mathsf{t} \,\, \Gamma \, \downarrow = \mathsf{Term} \,\, \downarrow \,\, \mathsf{t} \,\, 
               (context-pick-if-refl \{P = Typ\} \{dummy = W dummy\})
                                                                                                                                                                                                                                                                                                                                                                                                                                         Term\downarrow (weaken Typ-subst Typ-subst Typ-weaken Typ1 t) \Gamma \Downarrow = \text{Term}
                                                                                                                                                                                                                                                                                                                                                                                                                                         Term\Downarrow (weaken Typ-subst Typ-subst Typ-weaken Typ1-inv t) \Gamma \Downarrow = \mathsf{T}
'idfun' : \forall \{T\} \rightarrow \mathsf{Term} \{\epsilon\} (T' \rightarrow "T)
                                                                                                                                                                                                                                                                                                                                                                                                                                         \mathsf{Term} \Downarrow (\mathsf{subst} \mathsf{Typ\text{-}weaken} \mathsf{Typ1\text{-}weaken} \mathsf{Typ} t) \Gamma \Downarrow = \mathsf{Term} \Downarrow t \Gamma \Downarrow
'idfun' = '\lambda \bullet' 'VAR_0'
                                                                                                                                                                                                                                                                                                                                                                                                                                         Term\psi (subst Typ3-subst Typ2-subst Typ1-subst Typ-weaken Typ t) I
                                                                                                                                                                                                                                                                                                                                                                                                                                         Term\downarrow (weaken Typ-subst Typ2-subst Typ1-subst Typ-weaken Typ1 t
mutual
                                                                                                                                                                                                                                                                                                                                                                                                                                         \mathsf{Term} \Downarrow (\mathsf{substTyp1-substTyp-tProd}\ t)\ \Gamma \Downarrow T \Downarrow = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow T \Downarrow
               \mathsf{Context} \Downarrow : (\Gamma : \mathsf{Context}) \to \mathsf{Set} (|\mathsf{suc\ max-level})
                                                                                                                                                                                                                                                                                                                                                                                                                                         Term

↓ (subst Typ2-subst Typ-subst Typ-weaken Typ1-weaken Typ-w
               \mathsf{Typ} \Downarrow : \{\Gamma : \mathsf{Context}\} \to \mathsf{Typ} \; \Gamma \to \mathsf{Context} \Downarrow \Gamma \to \mathsf{Set} \; \mathsf{max}\text{-level}
                                                                                                                                                                                                                                                                                                                                                                                                                                         Term\downarrow (substTyp1-substTyp-weakenTyp2-weakenTyp t) \Gamma \Downarrow = \text{Ter}
                                                                                                                                                                                                                                                                                                                                                                                                                                         \mathsf{Term} \Downarrow (\mathsf{weakenTyp\text{-}weakenTyp1\text{-}weakenTyp}\ t)\ \Gamma \Downarrow = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                                                                         \mathsf{Term} \Downarrow (\mathsf{beta-under-subst}\ t)\ \Gamma \Downarrow = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow
               Context\psi \varepsilon = \top
               \mathsf{Context} \Downarrow (\Gamma \triangleright T) = \Sigma \; (\mathsf{Context} \Downarrow \Gamma) \; (\lambda \; \Gamma' \to \mathsf{Typ} \Downarrow T \; \Gamma')
                                                                                                                                                                                                                                                                                                                                                                                                                                         Term\Downarrow 'proj<sub>1</sub>'' \Gamma \Downarrow (x, p) = x
                                                                                                                                                                                                                                                                                                                                                                                                                                         Term\psi 'proj<sub>2</sub>'' (\Gamma \psi, (x, p)) = p
                                                                                                                                                                                                                                                                                                                                                                                                                                         \mathsf{Term} \Downarrow (\text{`existT'} \ x \ p) \ \Gamma \Downarrow = \mathsf{Term} \Downarrow x \ \Gamma \Downarrow \ , \ \mathsf{Term} \Downarrow p \ \Gamma \Downarrow
               \mathsf{Typ} \Downarrow (T_1 " x) \Gamma \Downarrow = \mathsf{Typ} \Downarrow T_1 (\Gamma \Downarrow , \mathsf{Term} \Downarrow x \Gamma \Downarrow)
                                                                                                                                                                                                                                                                                                                                                                                                                                         \mathsf{Term} \Downarrow (f'''' x) \Gamma \Downarrow = \mathsf{lift} (\mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) "\mathsf{lower} (\mathsf{Term} \Downarrow x \Gamma \Downarrow) 
               \mathsf{Typ} \Downarrow (T_2 "_1 a) (\Gamma \Downarrow , A \Downarrow) = \mathsf{Typ} \Downarrow T_2 ((\Gamma \Downarrow , \mathsf{Term} \Downarrow a \Gamma \Downarrow) , A \Downarrow)
               \mathsf{Typ} \Downarrow (T_3 \ ''_3 \ a) \ (((\Gamma \Downarrow \ , A \Downarrow) \ , B \Downarrow) \ , C \Downarrow) = \mathsf{Typ} \Downarrow T_3 \ ((((\Gamma \Downarrow \ , \mathsf{Term} \Downarrow \ a) \ \mathsf{Tle}) \ \mathsf{mall} \ \mathsf{lift} \ (\mathsf{lower} \ (\mathsf{Term} \Downarrow \ f \ \Gamma \Downarrow) \ ' \to '' \ \mathsf{lower} \ (\mathsf{Term} \Downarrow \ f \ \mathsf{lift}) \ \mathsf{lower} \ (\mathsf{Term} \Downarrow \ f \ \mathsf{lift}) \ \mathsf{lower} \ \mathsf
               \mathsf{Typ} \Downarrow (\mathsf{W}\ T_1)\ (\Gamma \Downarrow \ ,\ \_) = \mathsf{Typ} \Downarrow T_1\ \Gamma \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                                                                         \mathsf{Term} \Downarrow (f \mathsf{w}'' \to ''' x) \Gamma \Downarrow = \mathsf{lift} (\mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} \Downarrow f \Gamma \Downarrow) ' \to '' \mathsf{lower} (\mathsf{Term} ) ' \to '' \mathsf{lower} (\mathsf{lower} ) ' \to '' \mathsf{lower
                                                                                                                                                                                                                                                                                                                                                                                                                                         \mathsf{Typ} \Downarrow (\mathsf{W1}\ T_2)\ ((\Gamma \Downarrow \mathsf{,}\ A \Downarrow)\ \mathsf{,}\ B \Downarrow) = \mathsf{Typ} \Downarrow T_2\ (\Gamma \Downarrow \mathsf{,}\ B \Downarrow)
               \mathsf{Typ} \Downarrow (\mathsf{W2}\ T_3)\ (((\Gamma \Downarrow \ , A \Downarrow)\ , B \Downarrow)\ , C \Downarrow) = \mathsf{Typ} \Downarrow T_3\ ((\Gamma \Downarrow \ , B \Downarrow)\ , C \Downarrow)
                                                                                                                                                                                                                                                                                                                                                                                                                                         \mathsf{Term} \Downarrow "\to"" \to \mathsf{w}" \to "" \; \Gamma \Downarrow \; T \Downarrow = T \Downarrow
               \mathsf{Typ} \Downarrow (T \ \hookrightarrow' \ T_1) \ \Gamma \Downarrow = (T \Downarrow : \mathsf{Typ} \Downarrow T \ \Gamma \Downarrow) \to \mathsf{Typ} \Downarrow T_1 \ (\Gamma \Downarrow \ , T \Downarrow)
               Typ\downarrow 'Context' \Gamma \downarrow = Lifted Context
                                                                                                                                                                                                                                                                                                                                                                                                                                         Term\Downarrow 'tApp-nd' \Gamma \Downarrow f \Downarrow x \Downarrow = \text{lift (SW (lower } f \Downarrow \text{`'}_a \text{ lower } x \Downarrow))}
               \begin{array}{l} \mathsf{Typ} \Downarrow \mathsf{`Typ'} \; (\Gamma \Downarrow \; , \; T \Downarrow) = \mathsf{Lifted} \; (\mathsf{Typ} \; (\mathsf{lower} \; T \Downarrow)) \\ \mathsf{Typ} \Downarrow \mathsf{`Term'} \; (\Gamma \Downarrow \; , \; T \Downarrow \; , \; t \Downarrow) = \mathsf{Lifted} \; (\mathsf{Term} \; (\mathsf{lower} \; t \Downarrow)) \end{array}
                                                                                                                                                                                                                                                                                                                                                                                                                                         \mathsf{Term} \Downarrow \ulcorner \leftarrow \urcorner \Gamma \Downarrow T \Downarrow = T \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                                                                         \mathsf{Term} \Downarrow \ulcorner \to \urcorner \Gamma \Downarrow T \Downarrow = T \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                                                                       \mathsf{Typ} \Downarrow (`\Sigma' \ T \ T_1) \ \Gamma \Downarrow = \Sigma \ (\mathsf{Typ} \Downarrow T \ \Gamma \Downarrow) \ (\lambda \ T \Downarrow \to \mathsf{Typ} \Downarrow T_1 \ (\Gamma \Downarrow \ , \ T \Downarrow))
               \mathsf{Term} \Downarrow : \forall \ \{\Gamma : \mathsf{Context}\} \ \{T : \mathsf{Typ} \ \Gamma\} \to \mathsf{Term} \ T \to (\Gamma \Downarrow : \mathsf{Context} \Downarrow \Gamma) \\ \mathsf{Term}  \Downarrow (\Gamma T \Gamma \Downarrow B) \ \{A\} \ \{b\}) \ \Gamma \Downarrow = \mathsf{lift} \ (`\lambda \bullet' \ \{\epsilon\} \ (`\mathsf{VAR}_0` \ \{\epsilon\} \ \{\_\mathsf{Context}\}) \\ \mathsf{Term}  \Downarrow (\Gamma T \Gamma \Downarrow B) \ \{A\} \ \{b\}) \ \Gamma \Downarrow = \mathsf{lift} \ (`\lambda \bullet' \ \{\epsilon\} \ (`\mathsf{VAR}_0` \ \{\epsilon\} \ \{\_\mathsf{Context}\}) \\ \mathsf{Term}  \Downarrow (\Gamma T \Gamma \Downarrow B) \ \{A\} \ \{b\}) \ \Gamma \Downarrow = \mathsf{lift} \ (`\lambda \bullet' \ \{\epsilon\} \ (\mathsf{Context}) \ \mathsf{Term} 
                                                                                                                                                                                                                                                                                                                                                                                                                                       \mathsf{Term} \Downarrow (\mathsf{w}\ t)\ (\Gamma \Downarrow , A \Downarrow) = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow
               \mathsf{Term} \Downarrow (`\lambda \bullet `t) \ \Gamma \Downarrow T \Downarrow = \mathsf{Term} \Downarrow t \ (\Gamma \Downarrow \ , T \Downarrow)
               \mathsf{Term} \Downarrow (t \, ``_a \, t_1) \, \Gamma \Downarrow = \mathsf{Term} \Downarrow t \, \Gamma \Downarrow (\mathsf{Term} \Downarrow t_1 \, \Gamma \Downarrow)
               Term\downarrow 'VAR<sub>0</sub>' (\Gamma \downarrow , A \downarrow) = A \downarrow
                \begin{array}{l} \mathsf{Term} \Downarrow (\ulcorner \Gamma \urcorner \mathsf{c}) \ \Gamma \Downarrow = \mathsf{lift} \ \Gamma \\ \mathsf{Term} \Downarrow (\ulcorner T \urcorner \mathsf{T}) \ \Gamma \Downarrow = \mathsf{lift} \ T \\ \end{array} 
                                                                                                                                                                                                                                                                                                                                                                                            module well-typed-syntax-interpreter where
                                                                                                                                                                                                                                                                                                                                                                                                          open well-typed-syntax
               \mathsf{Term} \Downarrow (\ulcorner t \urcorner \mathsf{t}) \Upsilon \Downarrow = \mathsf{lift} \ t
                                                                                                                                                                                                                                                                                                                                                                                                          open well-typed-syntax-eq-dec
               \mathsf{Term} \Downarrow \mathsf{'quote\text{-}term'} \ \Gamma \Downarrow (\mathsf{lift} \ T \Downarrow) = \mathsf{lift} \ \ulcorner \ T \Downarrow \ \urcorner \mathsf{t}
               \mathsf{Term} \Downarrow (\mathsf{`quote-sigma'} \{\Gamma_0\} \{\Gamma_1\}) \ \Gamma \Downarrow (\mathsf{lift} \ \Gamma \ , \mathsf{lift} \ T) = \mathsf{lift} \ (\mathsf{`exist} \overrightarrow{\mathsf{m}} \mathsf{a} \not \{ I \widehat{\mathsf{q}} \not \} \mathsf{e} \Gamma \ I \mathsf{Le} \overline{\mathsf{v}} \mathsf{e} \Gamma \ T \ T)
               Term\Downarrow 'cast' \Gamma \Downarrow T \Downarrow = \text{lift (context-pick-if}
                                                                                                                                                                                                                                                                                                                                                                                                          max\text{-level} = well\text{-typed-syntax-pre-interpreter}. max\text{-level}
                               \{P = Typ\}
                               \{ | ower(\Sigma.proj_1 T \Downarrow) \}
                                                                                                                                                                                                                                                                                                                                                                                                          \mathsf{Context} \Downarrow : (\Gamma : \mathsf{Context}) \to \mathsf{Set} (|\mathsf{suc\ max-level})
                               (W dummy)
                                                                                                                                                                                                                                                                                                                                                                                                          \mathsf{Context} \Downarrow = \mathsf{well}\text{-}\mathsf{typed}\text{-}\mathsf{syntax}\text{-}\mathsf{pre}\text{-}\mathsf{inter}\mathsf{preter}.\mathsf{inner}.\mathsf{Context} \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                                                         (\lambda \ \ell \ P \ \Gamma' \ dummy \ val \rightarrow context-pick-if \{P = P\} \ dummy \ val)
                              (|\mathsf{lower}(\Sigma,\mathsf{proj}_2 T \Downarrow)))
                                                                                                                                                                                                                                                                                                                                                                                                                         (\lambda \ \ell \ P \ dummy \ val \rightarrow \text{context-pick-if-refl} \ \{P = P\} \ \{dummy\})
               \mathsf{Term} \Downarrow (\mathsf{SW}\ t)\ \Gamma \Downarrow = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow
               \mathsf{Term} \Downarrow (\mathsf{weakenTyp\text{-}substTyp\text{-}tProd}\ t)\ \Gamma \Downarrow T \Downarrow = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow T \Downarrow
               \mathsf{Term} \Downarrow (\mathsf{substTyp\text{-}weakenTyp1\text{-}VAR}_0 \ t) \ \Gamma \Downarrow = \mathsf{Term} \Downarrow t \ \Gamma \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                                          \mathsf{Typ} \Downarrow : \{\Gamma : \mathsf{Context}\} \to \mathsf{Typ} \; \Gamma \to \mathsf{Context} \Downarrow \Gamma \to \mathsf{Set} \; \mathsf{max}\text{-level}
               \mathsf{Term} \Downarrow (\mathsf{weakenTyp-tProd}\ t)\ \Gamma \Downarrow T \Downarrow = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow T \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                                          \mathsf{Typ} \Downarrow = \mathsf{well} - \mathsf{typed} - \mathsf{syntax} - \mathsf{pre} - \mathsf{interpreter}. \mathsf{inner}. \mathsf{Typ} \Downarrow
               \mathsf{Term} \Downarrow (\mathsf{weakenTyp-tProd-inv}\ t)\ \Gamma \Downarrow T \Downarrow = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow T \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                                                         (\lambda \ \ell \ P \ \Gamma' \ dummy \ val \rightarrow \mathsf{context-pick-if} \ \{\mathsf{P} = P\} \ dummy \ val)
               \mathsf{Term} \Downarrow (\mathsf{weakenTyp\text{-}weakenTyp\text{-}tProd}\ t)\ \Gamma \Downarrow T \Downarrow = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow T \Downarrow \ (\lambda\ \ell\ P\ dummy\ val \to \mathsf{context\text{-}pick\text{-}if\text{-}refl}\ \{\mathsf{P} = P\}\ \{dummy\})
               \mathsf{Term} \Downarrow (\mathsf{subst} \mathsf{Typ} 1 - \mathsf{tProd} \ t) \ \Gamma \Downarrow \ T \Downarrow = \mathsf{Term} \Downarrow t \ \Gamma \Downarrow \ T \Downarrow
               \mathsf{Term} \Downarrow (\mathsf{weakenTyp1-tProd}\ t)\ \Gamma \Downarrow T \Downarrow = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow T \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                                          \mathsf{Term} \Downarrow : \forall \{\Gamma : \mathsf{Context}\} \{T : \mathsf{Typ} \; \Gamma\} \to \mathsf{Term} \; T \to (\Gamma \Downarrow : \mathsf{Context} \Downarrow \Gamma) = \mathsf{Term} \; T \to \mathsf{Term} \;
               \mathsf{Term} \Downarrow (\mathsf{substTyp2-tProd}\ t)\ \Gamma \Downarrow T \Downarrow = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow T \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                                          Term \Downarrow = well-typed-syntax-pre-interpreter.inner.Term \Downarrow
               \mathsf{Term} \Downarrow (\mathsf{subst} \, \mathsf{Typ1} - \mathsf{subst} \, \mathsf{Typ-weaken} \, \mathsf{Typ-inv} \, t) \, \Gamma \Downarrow = \mathsf{Term} \Downarrow t \, \Gamma \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                                                        (\lambda \ \ell \ P \ \Gamma' \ dummy \ val \rightarrow context-pick-if \ \{P = P\} \ dummy \ val)
               \mathsf{Term} \Downarrow (\mathsf{substTyp1-substTyp-weakenTyp}\ t)\ \Gamma \Downarrow = \mathsf{Term} \Downarrow t\ \Gamma \Downarrow
                                                                                                                                                                                                                                                                                                                                                                                                                         (\lambda \ \ell \ P \ dummy \ val \rightarrow \text{context-pick-if-refl} \ \{P = P\} \ \{dummy\})
               Term\Downarrow (weaken Typ-weaken Typ-subst Typ1-subst Typ-weaken Typ t) \Gamma \Downarrow = \text{Term} \Downarrow t \Gamma \Downarrow
               Term \downarrow (substTyp2-substTyp1-substTyp-weakenTyp t) \Gamma \downarrow \downarrow = Term \downarrow en Twell-typed-syntax
```

```
'H': Τур ε
    open well-typed-syntax-interpreter
                                                                                                                                                     'H' = '□' '' 'H0'
     Contexts↓: Context↓ ε
     Context\epsilon \Downarrow = tt
                                                                                                                                                     H0': Typ ε
                                                                                                                                                     H0' = 'H' \rightarrow '' X'
    Typεψ: Typ ε → Set max-level
    Typ\epsilon \Downarrow T = \text{Typ} \Downarrow T \text{ Context} \epsilon \Downarrow
                                                                                                                                                     H': Set
                                                                                                                                                     H' = Term \{ \Gamma = \epsilon \} H0'
     \mathsf{Term}\,\epsilon \Downarrow : \{T : \mathsf{Typ}\,\,\epsilon\} \to \mathsf{Term}\,\,T \to \mathsf{Typ}\,\epsilon \Downarrow T
                                                                                                                                                     \text{`H0''}: \square \; (\text{`Typ'} \; \text{`'} \; \ulcorner \; \epsilon \; \lnot c)
    \mathsf{Term} \varepsilon \Downarrow t = \mathsf{Term} \Downarrow t \; \mathsf{Context} \varepsilon \Downarrow
                                                                                                                                                     'H0" = □ H0' ¬T
     \mathsf{Typ}\hspace{.01in}\epsilon\hspace{-.01in}\triangleright\!\!\!\downarrow: \forall \; \{A\} 	o \mathsf{Typ} \; (\epsilon \triangleright A) 	o \mathsf{Typ}\hspace{-.01in}\epsilon\!\!\!\downarrow\!\!\downarrow A 	o \mathsf{Set} \; \mathsf{max-level}
                                                                                                                                                    'H'' : Typ \epsilon
     'H'' = '□' '' 'H0''
     \mathsf{Term} \mathsf{E} \triangleright \Downarrow : \forall \{A\} \rightarrow \{T : \mathsf{Typ}\ (\mathsf{E} \triangleright A)\} \rightarrow \mathsf{Term}\ T \rightarrow (x : \mathsf{Type} \Downarrow A) \rightarrow \mathsf{Type} \triangleright \Downarrow Tx
                                                                                                                                                     toH-helper-helper : \forall \{k\} \rightarrow h2 \equiv k
     Term\varepsilon \triangleright \Downarrow t x = \text{Term} \Downarrow t \text{ (Context} \varepsilon \Downarrow , x)
                                                                                                                                                          \rightarrow \Box (h2 '' quote-sigma h '\rightarrow'' '\Box' '' \Box' h2 '' quote-sigma h '\rightarrow'' 'X
                                                                                                                                                          \rightarrow \Box (k '' quote-sigma h '\rightarrow'' '\Box' '' \lnot k '' quote-sigma h '\rightarrow'' 'X' \lnot
module löb where
                                                                                                                                                     toH-helper-helper p x= transport (\lambda k 
ightarrow \square (k " quote-sigma h '
ightarrow" "
    open well-typed-syntax
    open well-typed-quoted-syntax
                                                                                                                                                     toH-helper: \square (cast h " quote-sigma h '\rightarrow" 'H')
    open well-typed-syntax-interpreter-full
                                                                                                                                                     toH-helper = toH-helper-helper
      \text{module inner } ( \text{$X'$} : \mathsf{Typ} \ \epsilon) \ ( \text{$f'$} : \mathsf{Term} \ \{ \Gamma = \epsilon \, \triangleright \, ( \text{$'\Box'$} \, \text{```} \, \text{`$X'$} \, \text{$\urcorner$} T) \} \ (\mathsf{W} \, \text{$X'$})) \ w \\  \text{$\mathsf{ke}$} \text{$\mathsf{e}$} \text{$\mathsf{context}$-$\mathsf{pick}$-$\mathsf{if}} \ \{ \mathsf{P} = \mathsf{Typ} \} \ \{ \epsilon \, \triangleright \, \text{`$\Sigma'$} \, \text{`$\mathsf{Context'}$} \, \text{`$\mathsf{Typ'}$} \} \ (\mathsf{W} \, \mathsf{duminor}) 
         X : Set
                                                                                                                                                          (sym (context-pick-if-refl {P = Typ} {W dummy} {h2}))
         X = \mathsf{Typ}\epsilon \psi 'X'
                                                                                                                                                          \mathsf{f}'': (x:\mathsf{Type} \Downarrow ('\square' \,\, '' \,\, \ulcorner \,\, X' \,\, \urcorner \top)) \to \mathsf{Type} \bowtie \{ \,\, ('\square' \,\, '' \,\, \ulcorner \,\, X' \,\, \urcorner \top \} \,\, (\mathsf{W} \,\,\, 'X') \,\, x' \mathsf{toH}': \,\, \square \,\, ('\mathsf{H}'' \,\,\, '\to '' \,\,\, \mathsf{H}')
                                                                                                                                                     \mathsf{'toH'} = \ulcorner \overset{}{\rightarrow} \mathsf{'} \urcorner \mathsf{'o'} \mathsf{''fcomp-nd''} \mathsf{'''}_a \left( \ulcorner \overset{}{\rightarrow} \mathsf{'} \urcorner \mathsf{'''}_a \ulcorner \mathsf{toH-helper} \urcorner \mathsf{t} \right) \mathsf{'o'} \ulcorner \leftarrow
         f'' = \text{Term} \epsilon \triangleright \Downarrow f'
                                                                                                                                                     toH:H'\to H
         dummy: Typ \epsilon
                                                                                                                                                     toH h' = toH-helper 'o' h'
         dummy = 'Context'
         \mathsf{cast} : (\Gamma \nu : \Sigma \; \mathsf{Context} \; \mathsf{Typ}) \to \mathsf{Typ} \; (\epsilon \, \triangleright \, `\Sigma' \; `\mathsf{Context'} \; `\mathsf{Typ'})
                                                                                                                                                     from H-helper-helper : \forall \{k\} \rightarrow \mathsf{h2} \equiv k

ightarrow \square ('\square' " \sqcap h2 '' quote-sigma h '
ightarrow" 'X' \urcornerT '
ightarrow" h2 '' quote-sigm
         cast (\Gamma, \nu) = \text{context-pick-if } \{P = \text{Typ}\} \{\Gamma\} \text{ (W dummy) } \nu
                                                                                                                                                          \rightarrow \Box ('\Box' " \lnot k " quote-sigma h '\rightarrow" 'X' \lnot T '\rightarrow" k " quote-sigma
                                                                                                                                                     from H-helper-helper p x = transport (\lambda k \rightarrow \Box ('\Box' '' \sqcap k '' quote-sign
         Hf: (h: \Sigma \mathsf{Context} \mathsf{Typ}) \to \mathsf{Typ} \; \epsilon
         Hf h = (cast h " quote-sigma h " \rightarrow " "X")
                                                                                                                                                     from H-helper: \square ('H' '\rightarrow'' cast h'' quote-sigma h)
         \mathsf{qh}:\mathsf{Term}\;\{\Gamma=(\varepsilon\,\triangleright\,`\Sigma'\;\mathsf{`Context'}\;\mathsf{`Typ'})\}\;(\mathsf{W}\;(\mathsf{`Typ'}\;\mathsf{`'}\;\mathsf{`\varepsilon'}))
                                                                                                                                                     fromH-helper = fromH-helper-helper
         qh = f' w''' x
                                                                                                                                                          \{k = context-pick-if \{P = Typ\} \{\varepsilon \triangleright `\Sigma' `Context' `Typ'\} (W dumi
                                                                                                                                                          (sym (context-pick-if-refl {P = Typ} {W dummy} {h2})) (S<sub>00</sub>W1'\leftarrow (\vdash\rightarrow'\urcorner'o' ''fcomp-nd'' ''' _a (\vdash\rightarrow'\urcorner''' _a \vdash' \lorA\bullet' 'VAR_0' \urcornert
              where
                   f' = w \rightarrow `cast` '"_a 'VAR_0'
                                                                                                                                                     \mathsf{'from}\mathsf{H'}: \square \; (\mathsf{'H'}\; \mathsf{'}{\rightarrow}\mathsf{''}\; \mathsf{'H''})
                   x: Term (W ('Term' ''_1 \vdash \epsilon \lnot c '' \vdash '\Sigma' 'Context' 'Typ' \lnot T))
                                                                                                                                                     \text{`from} \mathsf{H'} = \ulcorner \to \text{'} \urcorner \text{ `o'} \text{ ``fcomp-nd''} \text{ ``'}_a \left( \ulcorner \to \text{'} \urcorner \text{ `''}_a \ \ulcorner \text{ from} \mathsf{H-helper} \ \urcorner \mathsf{t} \right) \text{ `o'}
                   x = (w \rightarrow `quote-sigma' ```a `VAR_0')
                                                                                                                                                     from H: H \rightarrow H'
         h2 : Typ (ε ▷ 'Σ' 'Context' 'Typ')
                                                                                                                                                     from H h' = \text{from H-helper 'o' } h'
         \mathsf{h2} = (\mathsf{W1} \ `\Box' \ `` \ (\mathsf{qh} \ \mathsf{w}`' {\rightarrow} ``' \ \mathsf{w} \ \ulcorner \ `X' \ \urcorner \mathsf{T}))
                                                                                                                                                    \mathsf{lob}: \square `X`
                                                                                                                                                    lob = from H h' "'_a   h'  t
         h: \Sigma Context Typ
         h = ((\epsilon \triangleright `\Sigma' `Context' `Typ'), h2)
                                                                                                                                                              H0: Typ\ \epsilon
                                                                                                                                                               f' = Conv0 \{ 'H0' \} \{ 'X' \} (SW1W (w \forall 'fromH' ''_a 'VAR_0'))
         H0 = Hfh
                                                                                                                                                              x : Term \{ \varepsilon \triangleright '\square' '' 'H0' \} (W ('\square' '' \sqcap 'H' \sqcap T)) \}
         H : Set
                                                                                                                                                              x = w \rightarrow 'quote-term' '''_a 'VAR_0'
         H = Term \{\Gamma = \epsilon\} H0
                                                                                                                                                               h': H
                                                                                                                                                               \mathsf{h}' = \mathsf{toH} \; ( `\lambda \bullet ' \; (\mathsf{w} \rightarrow ( `\lambda \bullet ' \; `f') \; ```_a \; (\mathsf{w} \rightarrow \rightarrow `\mathsf{tApp-nd'} \; ```_a \; \mathsf{f}' \; ```_a \; \mathsf{x} )
         'H0': □ ('Typ''' Γε ¬c)
         'H0' = □ H0 ¬T
                                                                                                                                               \mathsf{lob}: \{ \mathsf{'}X\mathsf{'}: \mathsf{Typ}\,\, \epsilon \} \to \square \, ((\mathsf{'}\square\mathsf{'}\,\,\mathsf{''}\,\,\ulcorner\,\, \mathsf{'}X\mathsf{'}\,\,\urcorner\mathsf{T})\,\,\mathsf{'}\!\to\!\mathsf{''}\,\,\, \mathsf{'}X\mathsf{'}) \to \square \,\,\mathsf{'}X\mathsf{'}
```

$$lob \{ 'X' \} 'f' = inner.lob 'X' (un'\lambda \bullet' 'f')$$

This is the text of the appendix, if you need one.

# Acknowledgments

Acknowledgments, if needed.

## References

P. Q. Smith, and X. Y. Jones. ...reference text...

13 2016/2/26