Kind Inference for Datatypes: Technical Supplement

NINGNING XIE, The University of Hong Kong, China
RICHARD A. EISENBERG, Bryn Mawr College, USA and Tweag I/O
BRUNO C. D. S. OLIVEIRA, The University of Hong Kong, China

This technical supplement to *Kind Inference for Datatypes* serves to expand upon the text in the main paper. It contains detailed typing rules, proofs, and connections to the Glasgow Haskell Compiler (GHC). Sections in this document are meant to connect to sections in the main paper. There are many hyperlinks throughout, especially those highlighting the connections to GHC; you may wish to read on a computer instead of on paper.

A OTHER LANGUAGE EXTENSIONS

This section accompanies Section 8 of the main paper, including discussion about more related language extensions. These extensions affect kind inference, but not in a fundamental way.

A.1 Visible Dependent Quantification

Besides specified type variables for which users can optionally provide type arguments, Haskell also incorporates *visible dependent quantification (VDQ)*[1], e.g., **type** $T :: \forall (k :: \star) \rightarrow k \rightarrow \star$, with which users are forced to provide type arguments to T. That is, one would use T with, e.g., $T\ \star\ \mathit{Int}$ and $T\ (\star \rightarrow \star)\ \mathit{Maybe}$, never just $T\ \mathit{Int}$. Visible dependent quantification is Haskell's equivalent to routine dependent quantification in dependently typed languages.

To support VDQ, rule DT-TT needs to be extended, as VDQ brings variables into scope for later reference. For example, given

type $T :: \forall (k :: \star) \rightarrow k \rightarrow \star$
data $T\ k\ a = \mathit{MkT}$

We should get a context $k :: \star, a :: k$ when checking MkT.

VDQ opens an interesting design choice: should unannotated type variables be able to introduce VDQ? For example, in the definition of P below, we use f and a as the arguments to T. To make it type-check, we need to infer $P :: \forall (f :: \star) \rightarrow f \rightarrow \star$.

data $P\ f\ a = \mathit{MkP}\ (T\ f\ a)$

However, the tricky part with inferring the kind of P is that we cannot have a fixed initial form of the kind of P, i.e., $\widehat{\alpha} \rightarrow \widehat{\beta} \rightarrow \star$ or $\forall (f : \widehat{\alpha}) \rightarrow \widehat{\beta} \rightarrow \star$, when type-checking the **rec** group of P, until we type-check P's body. In order to avoid this challenge, we support GHC's current ruling on the matter: *dependent variables must be manifestly so.* That is, the initial kind of a datatype includes VDQ only for those variables that appear, lexically, in the kind of a variable; other type parameters are reflected in a datatype's initial kind with a regular (non-dependent) arrow. This guideline rejects P as an example of non-manifest dependency.

[1] In GHC 8.6, GHC infers kinds using VDQ, but users are not allowed to write VDQ explicitly. This has been rectified for the GHC 8.8 release, as described in this proposal.

Authors' addresses: Ningning Xie, The University of Hong Kong, Department of Computer Science, Hong Kong, China, nnxie@cs.hku.hk; Richard A. Eisenberg, Bryn Mawr College, Department of Computer Science, Bryn Mawr, PA, USA, Tweag I/O, rae@richarde.dev; Bruno C. d. S. Oliveira, The University of Hong Kong, Department of Computer Science, Hong Kong, China, bruno@cs.hku.hk.

A.2 Datatype Promotion

Haskellers can use datatypes as kinds and can write data constructors in types [Yorgey et al. 2012]. In the PolyKinds system, types and kinds are mixed (allowing datatypes to be used as kinds), but there is no facility to use a data constructor in a type.

To support such usage, the kinding judgment must now use the term context to fetch the type of data constructors. Moreover, dependency analysis needs to take dependencies on data constructors into account.

Definition A.1 (Dependency Analysis with Type-Level Data). *We extend Definition 6.1 with*

(iii) The definition of T_1 depends on the definition of T_2 if T_1 uses data constructors of T_2.

While the appearance of data constructors in types enriches the type language considerably, they do not pose a particular challenge for inference; the rest of our presentation would remain unaffected.

A.3 Partial Type Signatures

For quite some time, GHC has supported kind signatures on a subset of a datatype's parameters, much like the partial type signatures described by Winant et al. [2014]. For example, *App*, below, does not have a signature but still has a kind annotation for f.

$\textbf{data}\ \mathit{App}\ (f :: \star \to \star)\ a = A\ (f\ a)$

To deal with such a construct we first need to amend the syntax of a datatype declaration to support kind annotations for variables.

$$\text{datatype decl.} \quad \mathcal{T} \quad ::= \quad \textbf{data}\ T\ \phi = \overline{\mathcal{D}_j}^{\,j}$$

Kind annotations can also contain free variables, which need to be generalized in a similar way as signatures. For example, $\mathit{T2}$ has kind $\forall\{k :: \star\}.\ \forall(f :: k).\ \star$.

$\textbf{data}\ \mathit{T2}\ (f :: k) = \mathit{MkT2}$

Supporting these partial signatures adds complication to rule PGM-DT-TT (and its algorithmic counterpart) to bring the kind variables into scope. However, and critically, a partial signature will still go via rule PGM-DT-TT, never rule PGM-DT-TTS, used for full signatures only. This means that a partial type signature does *not* unlock polymorphic recursion: the datatype will considered monomorphic and ungeneralized within its own recursive group.

B TODAY'S GHC

This paper describes, in depth, how kind inference can work for datatype declarations. Here, we review how our work relates to GHC. To make the claims in this section concrete, the technical supplement contains references to specific stretches of code within GHC, illustrating the points made here.

B.1 Constraint-based type inference

Type inference in GHC is based on generating and solving constraints [Pottier and Rémy 2005; Vytiniotis et al. 2011], distinct from our approach here, where we unify on the fly. Despite this different architecture, our results carry over to the constraint-based style. Instead of using eager unification, we can imagine accumulating constraints in output contexts Θ, and then invoking a solver to extend the context with solutions. This approach is taken by Eisenberg [2016].

B.2 Contexts

A typing context is *not* maintained in much of GHC's inference algorithm. Instead, a variable's kind is stored in the data structure representing the variable. This is very convenient, as it means that looking up a variable's type or kind is a pure, fast operation. One downside is that it is, in theory, possible for the same variable to be annotated with different kinds at different occurrences. Yet experience has shown that this particular problem does not arise often.

Beyond just storing variables' kinds, the typing context in this paper also critically stores variables' ordering. Lacking contexts, GHC uses a different mechanism: *level numbers*, originally invented to implement untouchability [Vytiniotis et al. 2011, Section 5.1]. Every type variable in GHC is assigned a level number during inference. Type variables contain a structure that includes level numbers. Roughly, the level number of a type variable a corresponds to the number of type variables in scope before a. Accordingly, we can tell the relative order (in a hypothetical context, according to the systems in this paper) of two variables simply by comparing their level numbers. One of GHC's invariants is that a unification variable at level n is never unified with a type that mentions a variable with a level number $m > n$; this is much like the extra checks in the unification judgments in our paper.

The *local scopes* of this paper are also tracked by GHC. All the variables in the same local scope are assigned the same level number, and they are flagged as reorderable. After inference is complete, GHC does a topological sort to get the final order.

A final role that contexts play in our formalism is that they store solutions for unification variables; we apply contexts as a substitution. In GHC, unification variables store mutable cells that get filled in. It has a process called *zonking*,[2] which is exactly analogous to our use contexts as substitutions. Zonking a unification variable replaces the variable with its solution, if any.

B.3 Unification

The solver in GHC still has to carry out unification, much along the lines of the unification judgment we present here. This algorithm has to deal with the heterogeneous unification problems we consider, as well. Indeed, GHC's unification algorithm recurs into the kinds of a unification variable and the type it is unifying with, just as ours does. As implied by our focus on decidability of unification, there have been a number of bugs in GHC's implementation that led to loops in the type checker; the most recent is #16902.

GHC actually uses several unification algorithms internally. It has an eager unifier, much like the one we describe. When that unifier fails, it generates the constraint that is sent to the solver. (The eager unifier is meant solely to be an optimization.) There is also a unifier meant to work after type inference is complete; it checks for instance overlap, for example. All the unifiers recur into kinds:

- The eager unifier recurs into kinds.
- The unifier in the solver recurs into kinds.
- The pure unifier uses an invariant that the kinds are related before looking at the types. It must recur when decomposing applications.

In addition, GHC also has an overlap problem within unification, as exhibited in our paper by the overlap between rules A-U-KVARL and A-U-KVARR in Figure 3. Both the eager unifier and the constraint-solver unifier deal with this ambiguity by using heuristics to choose which variable might be more suitable for unification. This particular issue—which variable to unify when there is a choice—has been the subject of some amount of churn over the years.

[2] There are actually two variants of zonking in GHC: we can zonk during type-checking or at the end. The difference between the variants is chiefly what to do for an unfilled unification variable. The former leaves them alone, while the latter has to default them somehow; details are beyond our scope here.

B.4 Promotion

The promotion operation, too, is present in GHC, though its form is quite different than what we have presented. Instead of promoting during unification, GHC simply refuses to solve a unification variable if any of the free variables of its supposed solution lives to the right of the variable in the context. Because GHC is working with constraints, it just leaves the unification problem as an unsolved constraint. If there remain unsolved constraints, GHC then promotes the variables it can: some cannot be promoted because they depend on locally bound quantified (not unification) type variables.

B.5 Complete user-supplied kinds

As of GHC 8.6, GHC does not support stand-alone kind signatures, as described in this paper. Instead, it supports *complete user-supplied kinds*, or CUSKs. A datatype has a CUSK when certain syntactic conditions are satisfied; GHC detects these conditions *before* doing any kind inference. These CUSKs are a poor substitute for proper kind signatures, as the syntactic cues are fragile and unexpected: users sometimes write a CUSK without meaning to, and also sometimes leave out a necessary part of a CUSK when they intend to specify the kind. Future versions of GHC will hopefully support kind signatures, though they will begin with the keyword **type** instead of **data**, as we have used in our paper.

Interestingly, it would be wrong to support CUSKs in a system without polymorphic kinds. Consider this example:

```
data S1 a = MkT1 S2
data S2 = MkS2 (S1 Maybe)
```

The types *S1* and *S2* form a group. We put *S2* (which has a CUSK) into the context with kind $\star$. When we check *S1*, we find no constraints on *a* (in the constraint-generation pass; see the general approach below). The kind of *S1* is then defaulted to $\star \to \star$. Checking *S2* fails. Instead, we wish to pretend that *S2* does not have a CUSK. This would mean that constraint-generation happens for all the constructors in both *S1* and *S2*, and *S1* would get its correct kind $(\star \to \star) \to \star$.

With kind-polymorphism, we have no problem because the kind of *T1* will be generalized to $\forall (k :: \star).\ k \to \star$.

This was reported as bug #16609.

B.6 Dependency analysis

The algorithm implemented in GHC for processing datatype declarations starts with dependency analysis, as ours does. The dependency analysis is less fine-grained than what we have proposed in this paper: signatures are ignored in the dependency analysis, and so datatypes with signatures are processed alongside all the others. This means that the kinds in the example below have more restrictive kinds in GHC than they do in our system:

```
cusk (S1 :: ∀k. k → ★)
data S1 a = MkS1 (S2 Int)
data S2 a = MkS2 (S3 Int)
data S3 a = MkS3 (S1 Int)
```

A naïve dependency analysis would put all three definitions in the same group. The kind for *S1* is given; it would indeed have that kind. The parameters of *S2* and *S3* would initially have an unknown kind, but when occurrences of *S2* and *S3* are processed (in the definitions of *S1* and *S2*, respectively), this unknown kind would become $\star$. Neither *S2* nor *S3* would be generalized.

There is a ticket to improve the dependency analysis: #9427.

B.7 Approach to kind-checking datatypes

GHC's approach is summarized in this comment. Overall kind-checking is orchestrated by this function.

After dependency analysis, so-called *initial kinds* are produced for all the datatypes in the group. These either come from a datatype's CUSK or from a simple analysis of the header of the datatype (without looking at constructors). This step corresponds to our algorithm's placing a binding for the datatype in the context, either with the kind signature or with a unification variable (rules A-PGM-DT-TTS and A-PGM-DT-TT).

If there is no CUSK, GHC then passes over all the datatype's constructors, collecting constraints on unification variables. After solving these constraints, GHC generalizes the datatype kind.

For all datatypes, now with generalized kinds, all data constructors are checked (again, for non-CUSK types). Because the kinds of the types are now generalized, this pass infers any invisible parameters to polykinded types. For non-CUSK types, this second pass using generalized kinds replaces the $T_i \mapsto T_i\ @\phi_i^c$ substitution in the context in the last premise to rule A-PGM-DT-TT. Performing a substitution—instead of re-generating and solving constraints—may be an opportunity for improvement in GHC.

B.8 Syntax for GADTs

Haskell's *syntax* for GADT declarations is very troublesome. Consider these examples:

```
data R a where
  MkR :: b → R b
data S a where
  MkS :: S b
data T a where
  MkT :: ∀(k :: ★) (b :: k). T b
```

In GHC's implementation of GADTs, any variables declared in the header (between **data** and **where**) *do not scope*. In all the examples above, the type variable a does not scope over the constructor declarations. This is why we have written the variable b in those types, to make it clear that b is distinct from a. We could have written a—it would still be a distinct a from that in the header—but it would be more confusing.

The question is: how do we determine the kind of the parameter to the datatype? One possibility is to look only in the header. In all cases above, we would infer no constraints and would give each type a kind of $\forall(k :: \star).\ k \to \star$. This is unfortunate, as it would make R a kind-indexed GADT: the MkR constructor would carry a proof that the kind of its type parameter is $\star$. This, in turn, wreaks havoc with type inference, as it is hard to infer the result type of a pattern-match against a GADT [Vytiniotis et al. 2011].

Furthermore, this approach might accept *more* programs than the user wants. Consider this definition:

```
data P a where
  MkP1 :: b → P b
  MkP2 :: f a → P f
```

Does the user want a kind-indexed GADT, noting that b and f have different kinds? Or would the user want this rejected? If we make the fully general kind $\forall k.\ k \to \star$ for P, this would be accepted, perhaps surprising users.

It thus seems we wish to look at the data constructors when inferring the kind of the datatype. The challenge in looking at data constructors is that their variables are *locally* bound. In MkR and

MkS, we implicitly quantify over b. In *MkR*, we discover that $b :: \star$, and thus that R must have kind $\star \to \star$. In *MkS*, we find no constraints on b's kind, and thus no constraints on S's argument's kind, and so we can generalize to get $S :: \forall (k :: \star).\ k \to \star$. Let us now examine *MkT*: it explicitly brings k and b into scope. Thus, the argument to T has *local* kind k. It would be impossible to unify the kind of T's argument—call it $\widehat{\alpha}$—with k, because k would be bound to the *right* of $\widehat{\alpha}$ in an inference context. Thus it seems we would reject T.

This result is also dissatisfying. In practice, GHC implements an ad-hoc algorithm, described in Section B.9.

Our conclusion here is that the design of GADTs in GHC/Haskell is flawed: the type variables mentioned in the header should indeed scope over the constructors. This would mean we could reject T: if the user wanted to explicitly make T polykinded, they could do so right in the header. We recognize that it would be hard to make this change today, but one result of this work is the interplay between scoping (order in the context) and unification; the current state of affairs will always require ad-hoc support.

B.9 Polymorphic recursion

One challenge in kind inference is in the handling of polymorphic recursion. Although non-CUSK types are indeed monomorphic during the constraint-generation pass, some limited form of polymorphic recursion can get through. This is because all type variables are represented by a special form of unification variable called a TyVarTv. TyVarTvs can unify only with other type variables. This design is motivated by the following examples:

```
data T1 (a :: k) b = MkT1 (T2 a b)
data T2 (c :: j) d = MkT2 (T1 c d)
data T3 a where
  MkT3 :: ∀(k :: ★) (b :: k). T3 b
```

We want to accept all of these definitions. The first two, *T1* and *T2*, form a mutually recursive group. Neither has a CUSK. However, the recursive occurrences are not polymorphically recursive: both recursive occurrences are at the *same* kind as the definition. Yet the first parameter to *T1* is declared to have kind k while the first parameter to *T2* is declared to have kind j. The solution: allow k to unify with j during the constraint-generation pass. We would *not* want to allow either k or j to unify with a non-variable, as that would seem to go against the user's wishes. But they must be allowed to unify with each other to accept this example.

With *T3* (identical to T from Section B.8), we have a different motivation. During inference, we will guess the kind of a; call it $\widehat{\alpha}$. When checking the *MkT3* constructor, we will need to unify $\widehat{\alpha}$ with the locally bound k. We cannot set $\widehat{\alpha} := k$, as that will fill $\widehat{\alpha}$ with a k, bound to $\widehat{\alpha}$'s *right* in the context. Instead, we must set $k := \widehat{\alpha}$. This is possible only if k is represented by a unification variable.

There are two known problems with this approach:

(1) It sometimes accepts polymorphic recursion, even without a CUSK. Here is an example:

```
data T4 a = ∀(k :: ★) (b :: k). MkT4 (T4 b)
```

The definition of *T4* is polymorphically recursive: the occurrence *T4 b* is specialized to a kind other than the kind of a. Yet this definition is accepted. The two kinds unify (as k becomes a unification variable, set to the guessed kind of a) during the constraint-generation pass. Then, *T4* is generalized to get the kind $\forall k.\ k \to \star$, at which point the last pass goes through without a hitch.

The reason this acceptance is troublesome is not that *T4* is somehow dangerous or unsafe. It is that we know that polymorphic recursion cannot be inferred [Henglein 1993], and yet GHC does it. Invariably, this must mean that GHC's algorithm will be hard to specify beyond its implementation.

This wrinkle is described on the GHC wiki.

(2) In rare cases, the constraint-generation pass will succeed, while the final pass—meant to be redundant—will fail. Here is an example:

$\textbf{data}\ \mathit{SameKind} :: k \to k \to \mathit{Type}$
$\textbf{data}\ \mathit{Bad}\ a\ \textbf{where}$
$\quad \mathit{MkBad} :: \forall k_1\ k_2\ (a :: k_1)\ (b :: k_2).\ \mathit{Bad}\ (\mathit{SameKind}\ a\ b)$

During the constraint-generation pass, the kinds k_1 and k_2 are allowed to unify, accepting the definition of *Bad*. During the final pass, however, k_1 and k_2 are proper quantified type variables, always distinct. Thus the *SameKind* a b type is ill-kinded and rejected.

The fact that this final pass can fail means that we cannot implement it via a simple substitution, as we do in rule A-PGM-DT-TT. One possible solution is our suggestion to change the scoping of type parameters to GADT-syntax datatype declarations. With that change, our second motivation above for TyVarTvs would disappear. GHC could then use TyVarTvs only for kind variables in the head of a datatype declaration, using proper quantified type variables in constructors. Of course, this change would break much code in the wild, and we do not truly expect it to ever be adopted.

This problem is documented in this comment.

B.10 The quantification check

Our quantification check (Section 7.2) also has a parallel in GHC, but GHC's solution to the problem differs from ours. Instead of rejecting programs that fail the quantification check, GHC accepts them, replacing the variables that would be (but cannot be) quantified with its constant $\mathit{Any} :: \forall k.\ k$. The *Any* type is uninhabited, but exists at all kinds. As such, it is an appropriate replacement for unquantifiable, unconstrained unification variables. Yet this decision in GHC has unfortunate consequences: the *Any* type can appear in error messages, and its introduction induces hard-to-understand type errors.

The GHC developers are questioning their approach to this problem. See this comment and this ticket.

Another design alternative is to generalize the variable to the leftmost position where it is still well-formed. Recall the example in Section 7.2:

$\textbf{data}\ \mathit{Proxy} :: \forall k.\ k \to \star$
$\textbf{data}\ \mathit{Relate} :: \forall a\ (b :: a).\ a \to \mathit{Proxy}\ b \to \star$
$\textbf{data}\ T :: \forall (a :: \star)\ (b :: a)\ (c :: a)\ d.\ \mathit{Relate}\ b\ d \to \star$

We have $d :: \widehat{\alpha}$, and $\widehat{\alpha} = \mathit{Proxy}\ \widehat{\beta}$, with $\widehat{\beta} :: a$. As there are no further constraints on $\widehat{\beta}$, the definition of T is rejected by the quantification check.

Instead of rejecting the program, or solving $\widehat{\beta}$ using *Any*, we can generalize over $\widehat{\beta}$ as a fresh variable f, which is put after a to make it well-kinded. Namely, we get

$\textbf{data}\ T :: \forall (a :: \star)\ \{f :: a\}\ (b :: a)\ (c :: a)\ (d :: \mathit{Proxy}\ f).\ \mathit{Relate}\ @a\ @f\ b\ d \to \star$

However, this ordering of the variables violates our declarative specification. Moreover, this type requires an inferred variable to be between specified variables. With higher-rank polymorphism, due to the fact that GHC does not support first-class type-level abstraction (i.e., Λ in types), this type cannot be instantiated to

$\forall (a :: \star)\ (b :: a)\ (c :: a)\ (d :: \mathit{Proxy}\ f).\ \mathit{Relate}\ @a\ @b\ b\ d \rightarrow \star$
or
$\forall (a :: \star)\ (b :: a)\ (c :: a)\ (d :: \mathit{Proxy}\ f).\ \mathit{Relate}\ @a\ @c\ b\ d \rightarrow \star$
which makes the generalization less useful.

B.11 ScopedSort

When GHC deals with a local scope—a set of variables that may be reordered—it does a topological sort on the variables at the end. However, not any topological sort will do: it must use one that preserves the left-to-right ordering of the variables as much as possible. This is because GHC considers these implicitly bound variables to be *specified*: they are available for visible type application. For example, recall the example from Section 2.2, modified slightly:

$\mathbf{data}\ Q\ (a :: (f\ b))\ (c :: k)\ (x :: f\ c)$

Inference will tell us that k must come before f and b, but the order of f and b is immaterial. Our approach here is to make f, b, and k *inferred* variables: users of Q will not be able to instantiate these parameters with visible type application. However, GHC takes a different view: because the user has written the names of f, b, and k, they will be *specified*. This choice means that the precise sorting algorithm GHC uses to fix the order of local scopes becomes part of the *specification* of the language. Indeed, GHC documents the precise algorithm in its manual. If we followed suit, the algorithm would have to appear in our declarative specification, which goes against the philosophy of a declarative system.

Some recent debate led to a conclusion that we would change the interpretation of the Q example from the main paper, meaning that its kind variables would indeed become *inferred*. However, the problem with ScopedSort still exists in type signatures, where type variables may be implicitly bound.

B.12 The "forall-or-nothing" rule

GHC implements the so-called *forall-or-nothing* rule, which states that either *all* variables are quantified by a user-written `forall`, or none are. These examples illustrate the effect:

$\mathit{ex1} :: a \rightarrow b \rightarrow a$
$\mathit{ex2} :: \forall a\ b.\ a \rightarrow b \rightarrow a$
$\mathit{ex3} :: \forall a.\ a \rightarrow b \rightarrow a$
$\mathit{ex4} :: (\forall a.\ a \rightarrow b \rightarrow a)$

The signatures for both *ex1* and *ex2* are accepted: *ex1* quantifies none, while *ex2* quantifies all. The signature for *ex3* is rejected, as GHC rejects a mixed economy. However, and perhaps surprisingly, *ex4* is accepted. The only difference between *ex3* and *ex4* is the seemingly-redundant parentheses. However, because the forall-or-nothing rule applies only at the top level of a signature, the rule is not in effect for the $\forall$in *ex4*.

This rule interacts with the main paper only in that our formalism (and some of our examples) does not respect it. This may be the cause of differing behavior between GHC and the examples we present.

C COMPLETE SET OF RULES

In this section we include the complete set of rules. Some of the rules are repeated from those in the paper.

C.1 Declarative Haskell98

$\boxed{\Sigma \vdash^{k} \sigma : \kappa}$ *(Kinding for Polymorphic Types)*

K-FORALL
$$\frac{\Sigma, a : \kappa \vdash^{k} \sigma : \star}{\Sigma \vdash^{k} \forall a : \kappa . \sigma : \star}$$

$\boxed{\Sigma \vdash \Psi}$ *(Well-formed Term Contexts)*

ECTX-EMPTY
$$\frac{}{\Sigma \vdash \bullet}$$

ECTX-DCON
$$\frac{\Sigma \vdash \Psi \qquad \Sigma \vdash^{k} \sigma : \star}{\Sigma \vdash \Psi, D : \sigma}$$

C.2 Algorithmic Haskell98

$\boxed{\Delta \Vdash^{k} \sigma : \kappa \dashv \Theta}$ *(Kinding for Polymorphic Types)*

A-K-FORALL
$$\frac{\Delta \Vdash^{kv} \kappa \qquad \Delta, a : \kappa \Vdash^{k} \sigma : \kappa_2 \dashv \Theta, a : \kappa \qquad [\Theta]\kappa_2 = \star}{\Delta \Vdash^{k} \forall a : \kappa . \sigma : \star \dashv \Theta}$$

$\boxed{\Delta \Vdash^{kc} \sigma \Leftarrow \kappa}$ *(Checking)*

A-KC-EQ
$$\frac{\Delta \Vdash^{k} \sigma : \kappa_1 \dashv \Delta \qquad [\Delta]\kappa_1 = [\Delta]\kappa_2}{\Delta \Vdash^{kc} \sigma \Leftarrow \kappa_2}$$

$\boxed{\Delta \Vdash^{kv} \kappa}$ *(Well-formed Kinds)*

A-KV-STAR
$$\frac{}{\Delta \Vdash^{kv} \star}$$

A-KV-ARROW
$$\frac{\Delta \Vdash^{kv} \kappa_1 \qquad \Delta \Vdash^{kv} \kappa_2}{\Delta \Vdash^{kv} \kappa_1 \rightarrow \kappa_2}$$

A-KV-KUVAR
$$\frac{\widehat{\alpha} \in \Delta}{\Delta \Vdash^{kv} \widehat{\alpha}}$$

$\boxed{\Delta \text{ ok}}$ *(Well-formed Type Contexts)*

A-TCTX-EMPTY
$$\frac{}{\bullet \text{ ok}}$$

A-TCTX-TVAR
$$\frac{\Delta \text{ ok} \qquad \Delta \Vdash^{kv} \kappa}{\Delta, a : \kappa \text{ ok}}$$

A-TCTX-TCON
$$\frac{\Delta \text{ ok} \qquad \Delta \Vdash^{kv} \kappa}{\Delta, T : \kappa \text{ ok}}$$

A-TCTX-KUVAR
$$\frac{\Delta \text{ ok}}{\Delta, \widehat{\alpha} \text{ ok}}$$

A-TCTX-KUVARSOLVED
$$\frac{\Delta \text{ ok} \qquad \Delta \Vdash^{kv} \kappa}{\Delta, \widehat{\alpha} = \kappa \text{ ok}}$$

$\boxed{\Delta \Vdash^{ectx} \Gamma}$ *(Well-formed Term Contexts)*

A-ECTX-EMPTY
$$\frac{}{\Delta \Vdash^{ectx} \bullet}$$

A-ECTX-DCON
$$\frac{\Delta \Vdash^{ectx} \Gamma \qquad \Delta \Vdash^{kc} \sigma \Leftarrow \star}{\Delta \Vdash^{ectx} \Gamma, D : \sigma}$$

$\boxed{\Delta \twoheadrightarrow \Omega}$ *(Defaulting)*

A-CTXDE-EMPTY
$$\frac{}{\bullet \twoheadrightarrow \bullet}$$

A-CTXDE-TVAR
$$\frac{\Delta \twoheadrightarrow \Omega}{\Delta, a : \kappa \twoheadrightarrow \Omega, a : \kappa}$$

A-CTXDE-TCON
$$\frac{\Delta \twoheadrightarrow \Omega}{\Delta, T : \kappa \twoheadrightarrow \Omega, T : \kappa}$$

A-CTXDE-KUVARSOLVED
$$\frac{\Delta \twoheadrightarrow \Omega}{\Delta, \widehat{\alpha} = \kappa \twoheadrightarrow \Omega, \widehat{\alpha} = \kappa}$$

A-CTXDE-SOLVE
$$\frac{\Delta \twoheadrightarrow \Omega}{\Delta, \widehat{\alpha} \twoheadrightarrow \Omega, \widehat{\alpha} = \star}$$

C.3 Context Application in Haskell98

$[\Delta]\kappa$ applies Δ as a substitution to κ.

$$\begin{array}{lcl}
[\Delta]\star & = & \star \\
[\Delta]\kappa_1 \to \kappa_2 & = & [\Delta]\kappa_1 \to [\Delta]\kappa_2 \\
[\Delta[\widehat{\alpha}]]\widehat{\alpha} & = & \widehat{\alpha} \\
[\Delta[\widehat{\alpha} = \kappa]]\widehat{\alpha} & = & [\Delta[\widehat{\alpha} = \kappa]]\kappa
\end{array}$$

$[\Delta]\Gamma$ applies Δ as a substitution to Γ.

$$\begin{array}{lcl}
[\Delta]\bullet & = & \bullet \\
[\Delta](\Gamma, D : \sigma) & = & [\Delta]\Gamma, D : [\Delta]\sigma
\end{array}$$

$[\Omega]\Delta$ applies Ω as a substitution to Δ.

$$\begin{array}{lcll}
[\bullet]\bullet & = & \bullet & \\
[\Omega, a : \kappa](\Delta, a : \kappa) & = & [\Omega]\Delta, a : [\Omega]\kappa & \\
[\Omega, T : \kappa](\Delta, T : \kappa) & = & [\Omega]\Delta, T : [\Omega]\kappa & \\
[\Omega, \widehat{\alpha} = \kappa](\Delta, \widehat{\alpha}) & = & [\Omega]\Delta & \\
[\Omega, \widehat{\alpha} = \kappa](\Delta, \widehat{\alpha} = \kappa') & = & [\Omega]\Delta & \text{if } [\Omega]\kappa = [\Omega]\kappa' \\
[\Omega, \widehat{\alpha} = \kappa]\Delta & = & [\Omega]\Delta & \text{if } \widehat{\alpha} \notin \Delta
\end{array}$$

C.4 Context Extension in Haskell98

$\boxed{\Delta \longrightarrow \Theta}$ *(Context Extension)*

$$\textsc{a-ctxe-empty}\ \frac{}{\bullet \longrightarrow \bullet}
\qquad
\textsc{a-ctxe-tvar}\ \frac{\Delta \longrightarrow \Theta}{\Delta, a : \kappa \longrightarrow \Theta, a : \kappa}
\qquad
\textsc{a-ctxe-tcon}\ \frac{\Delta \longrightarrow \Theta}{\Delta, T : \kappa \longrightarrow \Theta, T : \kappa}
\qquad
\textsc{a-ctxe-kuvar}\ \frac{\Delta \longrightarrow \Theta}{\Delta, \widehat{\alpha} \longrightarrow \Theta, \widehat{\alpha}}$$

$$\textsc{a-ctxe-kuvarSolved}\ \frac{\Delta \longrightarrow \Theta \qquad [\Theta]\kappa_1 = [\Theta]\kappa_2}{\Delta, \widehat{\alpha} = \kappa_1 \longrightarrow \Theta, \widehat{\alpha} = \kappa_2}
\qquad
\textsc{a-ctxe-solve}\ \frac{\Delta \longrightarrow \Theta \qquad \Theta \Vdash^{\mathsf{kv}} \kappa}{\Delta, \widehat{\alpha} \longrightarrow \Theta, \widehat{\alpha} = \kappa}
\qquad
\textsc{a-ctxe-add}\ \frac{\Delta \longrightarrow \Theta}{\Delta \longrightarrow \Theta, \widehat{\alpha}}
\qquad
\textsc{a-ctxe-addSolved}\ \frac{\Delta \longrightarrow \Theta \qquad \Theta \Vdash^{\mathsf{kv}} \kappa}{\Delta \longrightarrow \Theta, \widehat{\alpha} = \kappa}$$

C.5 Declarative PolyKinds

$\boxed{\rceil \sigma \lceil}$ *(Kind results in $\star$)*

$$\textsc{sr-star}\ \frac{}{\rceil \star \lceil}
\qquad
\textsc{sr-arrow}\ \frac{\rceil \kappa_2 \lceil}{\rceil \kappa_1 \to \kappa_2 \lceil}
\qquad
\textsc{sr-forall}\ \frac{\rceil \sigma \lceil}{\rceil \forall \phi.\ \sigma \lceil}$$

$\boxed{\Sigma \vdash^{\mathsf{inst}} \mu_1 : \eta \sqsubseteq \omega \rightsquigarrow \mu_2}$ *(Instantiation)*

$$\textsc{inst-refl}\ \frac{}{\Sigma \vdash^{\mathsf{inst}} \mu : \omega \sqsubseteq \omega \rightsquigarrow \mu}
\qquad
\textsc{inst-forall}\ \frac{\Sigma \vdash^{\mathsf{ela}} \rho : \omega_1 \qquad \Sigma \vdash^{\mathsf{inst}} \mu_1\ @\rho : \eta[a \mapsto \rho] \sqsubseteq \omega_2 \rightsquigarrow \mu_2}{\Sigma \vdash^{\mathsf{inst}} \mu_1 : \forall a : \omega_1 . \eta \sqsubseteq \omega_2 \rightsquigarrow \mu_2}$$

$$\textsc{inst-forall-infer}\ \frac{\Sigma \vdash^{\mathsf{ela}} \rho : \omega_1 \qquad \Sigma \vdash^{\mathsf{inst}} \mu_1\ @\rho : \eta[a \mapsto \rho] \sqsubseteq \omega_2 \rightsquigarrow \mu_2}{\Sigma \vdash^{\mathsf{inst}} \mu_1 : \forall \{a : \omega_1\}. \eta \sqsubseteq \omega_2 \rightsquigarrow \mu_2}$$

$\boxed{\Sigma \vdash^{\mathsf{kc}} \sigma \Leftarrow \omega \rightsquigarrow \mu}$ *(Kind Checking)*

$$\textsc{kc-sub}\ \frac{\Sigma \vdash^{\mathsf{k}} \sigma : \eta \rightsquigarrow \mu_1 \qquad \Sigma \vdash^{\mathsf{inst}} \mu_1 : \eta \sqsubseteq \omega \rightsquigarrow \mu_2}{\Sigma \vdash^{\mathsf{kc}} \sigma \Leftarrow \omega \rightsquigarrow \mu_2}$$

$\boxed{\Sigma \vdash^{\mathsf{k}} \sigma : \eta \rightsquigarrow \mu}$ *(Kinding)*

KTT-STAR

$$\frac{}{\Sigma \vdash^{\mathsf{k}} \star : \star \rightsquigarrow \star}$$

KTT-NAT

$$\frac{}{\Sigma \vdash^{\mathsf{k}} \mathsf{Int} : \star \rightsquigarrow \mathsf{Int}}$$

KTT-VAR

$$\frac{(a : \omega) \in \Sigma}{\Sigma \vdash^{\mathsf{k}} a : \omega \rightsquigarrow a}$$

KTT-ARROW

$$\frac{}{\Sigma \vdash^{\mathsf{k}} \rightarrow : \star \rightarrow \star \rightarrow \star \rightsquigarrow \rightarrow}$$

KTT-TCON

$$\frac{(T : \eta) \in \Sigma}{\Sigma \vdash^{\mathsf{k}} T : \eta \rightsquigarrow T}$$

KTT-APP

$$\frac{\Sigma \vdash^{\mathsf{k}} \tau_1 : \eta_1 \rightsquigarrow \rho_1 \qquad \Sigma \vdash^{\mathsf{inst}} \rho_1 : \eta_1 \sqsubseteq (\omega_1 \rightarrow \omega_2) \rightsquigarrow \rho_2 \qquad \Sigma \vdash^{\mathsf{kc}} \tau_2 \Leftarrow \omega_1 \rightsquigarrow \rho_3}{\Sigma \vdash^{\mathsf{k}} \tau_1\, \tau_2 : \omega_2 \rightsquigarrow \rho_2\, \rho_3}$$

KTT-KAPP

$$\frac{\Sigma \vdash^{\mathsf{k}} \kappa_1 : \forall a : \omega.\eta \rightsquigarrow \rho_1 \qquad \Sigma \vdash^{\mathsf{kc}} \kappa_2 \Leftarrow \omega \rightsquigarrow \rho_2}{\Sigma \vdash^{\mathsf{k}} \kappa_1\ @\kappa_2 : \eta[a \mapsto \rho_2] \rightsquigarrow \rho_1\ @\rho_2}$$

KTT-KAPP-INFER

$$\frac{\begin{array}{c}\Sigma \vdash^{\mathsf{k}} \kappa_1 : \forall\{\overline{a_i : \omega_i}^{\,i}\}.\forall a : \omega.\eta \rightsquigarrow \rho_1' \\ \overline{\Sigma \vdash^{\mathsf{ela}} \rho_i : \omega_i[\overline{a_i \mapsto \rho_i}^{\,i}]}^{\,i} \qquad \Sigma \vdash^{\mathsf{kc}} \kappa_2 \Leftarrow \omega[\overline{a_i \mapsto \rho_i}^{\,i}] \rightsquigarrow \rho_2'\end{array}}{\Sigma \vdash^{\mathsf{k}} \kappa_1\ @\kappa_2 : \eta[\overline{a_i \mapsto \rho_i}^{\,i}][a \mapsto \rho_2] \rightsquigarrow \rho_1'\ @\overline{\rho_i}^{\,i}\ @\rho_2'}$$

KTT-FORALL

$$\frac{\Sigma \vdash^{\mathsf{kc}} \kappa \Leftarrow \star \rightsquigarrow \omega \qquad \Sigma, a : \omega \vdash^{\mathsf{kc}} \sigma \Leftarrow \star \rightsquigarrow \mu}{\Sigma \vdash^{\mathsf{k}} \forall a : \kappa.\sigma : \star \rightsquigarrow \forall a : \omega.\mu}$$

KTT-FORALLI

$$\frac{\Sigma \vdash^{\mathsf{ela}} \omega : \star \qquad \Sigma, a : \omega \vdash^{\mathsf{kc}} \sigma \Leftarrow \star \rightsquigarrow \mu}{\Sigma \vdash^{\mathsf{k}} \forall a.\sigma : \star \rightsquigarrow \forall a : \omega.\mu}$$

$\boxed{\Sigma \vdash^{\mathsf{ela}} \mu : \eta}$ *(Elaborated Kinding)*

ELA-STAR

$$\frac{}{\Sigma \vdash^{\mathsf{ela}} \star : \star}$$

ELA-NAT

$$\frac{}{\Sigma \vdash^{\mathsf{ela}} \mathsf{Int} : \star}$$

ELA-VAR

$$\frac{(a : \omega) \in \Sigma}{\Sigma \vdash^{\mathsf{ela}} a : \omega}$$

ELA-TCON

$$\frac{(T : \eta) \in \Sigma}{\Sigma \vdash^{\mathsf{ela}} T : \eta}$$

ELA-ARROW

$$\frac{}{\Sigma \vdash^{\mathsf{ela}} \rightarrow : \star \rightarrow \star \rightarrow \star}$$

ELA-APP

$$\frac{\Sigma \vdash^{\mathsf{ela}} \rho_1 : \omega_1 \rightarrow \omega_2 \qquad \Sigma \vdash^{\mathsf{ela}} \rho_2 : \omega_1}{\Sigma \vdash^{\mathsf{ela}} \rho_1\, \rho_2 : \omega_2}$$

ELA-KAPP

$$\frac{\Sigma \vdash^{\mathsf{ela}} \rho_1 : \forall a : \omega.\eta \qquad \Sigma \vdash^{\mathsf{ela}} \rho_2 : \omega}{\Sigma \vdash^{\mathsf{ela}} \rho_1\ @\rho_2 : \eta[a \mapsto \rho_2]}$$

ELA-KAPP-INFER

$$\frac{\Sigma \vdash^{\mathsf{ela}} \rho_1 : \forall\{a : \omega\}.\eta \qquad \Sigma \vdash^{\mathsf{ela}} \rho_2 : \omega}{\Sigma \vdash^{\mathsf{ela}} \rho_1\ @\rho_2 : \eta[a \mapsto \rho_2]}$$

ELA-FORALL

$$\frac{\Sigma \vdash^{\mathsf{ela}} \omega : \star \qquad \Sigma, a : \omega \vdash^{\mathsf{ela}} \mu : \star}{\Sigma \vdash^{\mathsf{ela}} \forall a : \omega.\mu : \star}$$

ELA-FORALL-INFER

$$\frac{\Sigma \vdash^{\mathsf{ela}} \omega : \star \qquad \Sigma, a : \omega \vdash^{\mathsf{ela}} \mu : \star}{\Sigma \vdash^{\mathsf{ela}} \forall\{a : \omega\}.\mu : \star}$$

$\boxed{\Sigma\ \mathsf{ok}}$ *(Well-formed Type Contexts)*

TCTX-EMPTY

$$\frac{}{\bullet\ \mathsf{ok}}$$

TCTX-TVAR-TT

$$\frac{\Sigma\ \mathsf{ok} \qquad \Sigma \vdash^{\mathsf{ela}} \rho : \star}{\Sigma, a : \rho\ \mathsf{ok}}$$

TCTX-TCON-TT

$$\frac{\Sigma\ \mathsf{ok} \qquad \Sigma \vdash^{\mathsf{ela}} \eta : \star}{\Sigma, T : \eta\ \mathsf{ok}}$$

$\boxed{\Sigma \vdash \Psi}$ *(Well-formed Term Contexts)*

ECTX-EMPTY

$$\frac{}{\Sigma \vdash \bullet}$$

ECTX-DCON-TT

$$\frac{\Sigma \vdash \Psi \qquad \Sigma \vdash^{\mathsf{ela}} \mu : \star}{\Sigma \vdash \Psi, D : \mu}$$

C.6 Algorithmic PolyKinds

$\boxed{\Delta \Vdash^{\mathsf{inst}} \mu_1 : \eta \sqsubseteq \omega \rightsquigarrow \mu_2 \dashv \Theta}$ *(Instantiation)*

A-INST-REFL

$$\frac{\Delta \Vdash^{\mathsf{u}} \omega_1 \approx \omega_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{inst}} \mu : \omega_1 \sqsubseteq \omega_2 \rightsquigarrow \mu \dashv \Theta}$$

A-INST-FORALL

$$\frac{\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{inst}} \mu_1 \mathbin{@}\widehat{\alpha} : \eta[a \mapsto \widehat{\alpha}] \sqsubseteq \omega_2 \rightsquigarrow \mu_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{inst}} \mu_1 : \forall a : \omega_1 . \eta \sqsubseteq \omega_2 \rightsquigarrow \mu_2 \dashv \Theta}$$

A-INST-FORALL-INFER

$$\frac{\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{inst}} \mu_1 \mathbin{@}\widehat{\alpha} : \eta[a \mapsto \widehat{\alpha}] \sqsubseteq \omega_2 \rightsquigarrow \mu_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{inst}} \mu_1 : \forall \{a : \omega_1\} . \eta \sqsubseteq \omega_2 \rightsquigarrow \mu_2 \dashv \Theta}$$

$\boxed{\Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \eta \rightsquigarrow \mu \dashv \Theta}$ *(Kind Checking)*

A-KC-SUB

$$\frac{\Delta \Vdash^{\mathsf{k}} \sigma : \eta \rightsquigarrow \mu_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{inst}} \mu_1 : [\Delta_1]\eta \sqsubseteq [\Delta_1]\omega \rightsquigarrow \mu_2 \dashv \Delta_2}{\Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \omega \rightsquigarrow \mu_2 \dashv \Delta_2}$$

$\boxed{\Delta \Vdash^{\mathsf{k}} \sigma : \eta \rightsquigarrow \mu \dashv \Theta}$ *(Kinding)*

A-KTT-STAR

$$\frac{}{\Delta \Vdash^{\mathsf{k}} \star : \star \rightsquigarrow \star \dashv \Delta}$$

A-KTT-KUVAR

$$\frac{(\widehat{\alpha} : \eta) \in \Delta}{\Delta \Vdash^{\mathsf{k}} \widehat{\alpha} : \eta \rightsquigarrow \widehat{\alpha} \dashv \Delta}$$

A-KTT-NAT

$$\frac{}{\Delta \Vdash^{\mathsf{k}} \mathsf{Int} : \star \rightsquigarrow \mathsf{Int} \dashv \Delta}$$

A-KTT-VAR

$$\frac{(a : \omega) \in \Delta}{\Delta \Vdash^{\mathsf{k}} a : \omega \rightsquigarrow a \dashv \Delta}$$

A-KTT-TCON

$$\frac{(T : \eta) \in \Delta}{\Delta \Vdash^{\mathsf{k}} T : \eta \rightsquigarrow T \dashv \Delta}$$

A-KTT-ARROW

$$\frac{}{\Delta \Vdash^{\mathsf{k}} \rightarrow : \star \rightarrow \star \rightarrow \star \rightsquigarrow \rightarrow \dashv \Delta}$$

A-KTT-FORALL

$$\frac{\Delta \Vdash^{\mathsf{kc}} \kappa \Leftarrow \star \rightsquigarrow \omega \dashv \Delta_1 \qquad \Delta_1, a : \omega \Vdash^{\mathsf{kc}} \sigma \Leftarrow \star \rightsquigarrow \mu \dashv \Delta_2, a : \omega, \Delta_3 \qquad \Delta_3 \hookrightarrow a}{\Delta \Vdash^{\mathsf{k}} \forall a : \kappa . \sigma : \star \rightsquigarrow \forall a : \omega . [\Delta_3]\mu \dashv \Delta_2, \mathsf{unsolved}(\Delta_3)}$$

A-KTT-APP

$$\frac{\Delta \Vdash^{\mathsf{k}} \tau_1 : \eta_1 \rightsquigarrow \rho_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{kapp}} (\rho_1 : [\Delta_1]\eta_1) \bullet \tau_2 : \omega \rightsquigarrow \rho \dashv \Theta}{\Delta \Vdash^{\mathsf{k}} \tau_1\, \tau_2 : \omega \rightsquigarrow \rho \dashv \Theta}$$

A-KTT-FORALLI

$$\frac{\Delta, \widehat{\alpha} : \star, a : \widehat{\alpha} \Vdash^{\mathsf{kc}} \sigma \Leftarrow \star \rightsquigarrow \mu \dashv \Delta_2, a : \widehat{\alpha}, \Delta_3 \qquad \Delta_3 \hookrightarrow a}{\Delta \Vdash^{\mathsf{k}} \forall a . \sigma : \star \rightsquigarrow \forall a : \widehat{\alpha} . [\Delta_3]\mu \dashv \Delta_2, \mathsf{unsolved}(\Delta_3)}$$

A-KTT-KAPP

$$\frac{\Delta \Vdash^{\mathsf{k}} \tau_1 : \eta \rightsquigarrow \rho_1 \dashv \Delta_1 \qquad [\Delta_1]\eta = \forall a : \omega . \eta_2 \qquad \Delta_1 \Vdash^{\mathsf{kc}} \tau_2 \Leftarrow \omega \rightsquigarrow \rho_2 \dashv \Delta_2}{\Delta \Vdash^{\mathsf{k}} \tau_1 \mathbin{@}\tau_2 : \eta_2[a \mapsto \rho_2] \rightsquigarrow \rho_1 \mathbin{@}\rho_2 \dashv \Delta_2}$$

A-KTT-KAPP-INFER

$$\frac{\begin{array}{c} \Delta \Vdash^{\mathsf{k}} \tau_1 : \eta \rightsquigarrow \rho_1 \dashv \Delta_1 \\ [\Delta_1]\eta = \forall \{\overline{a_i : \omega_i}^{\,i}\} . \forall a : \omega . \eta_2 \qquad \Delta_1, \overline{\widehat{\alpha}_i : \omega_i[\overline{a_i \mapsto \widehat{\alpha}_i}^{\,i}]}^{\,i} \Vdash^{\mathsf{kc}} \tau_2 \Leftarrow \omega[\overline{a_i \mapsto \widehat{\alpha}_i}^{\,i}] \rightsquigarrow \rho_2 \dashv \Delta_2 \end{array}}{\Delta \Vdash^{\mathsf{k}} \tau_1 \mathbin{@}\tau_2 : \eta_2[\overline{a_i \mapsto \widehat{\alpha}_i}^{\,i}][a \mapsto \rho_2] \rightsquigarrow \rho_1 \mathbin{@}\overline{\widehat{\alpha}_i}^{\,i} \mathbin{@}\rho_2 \dashv \Delta_2}$$

$\boxed{\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \eta) \bullet \tau : \omega \rightsquigarrow \rho_2 \dashv \Theta}$ *(Application Kinding)*

A-KAPP-TT-ARROW

$$\frac{\Delta \Vdash^{\mathsf{kc}} \tau \Leftarrow \omega_1 \rightsquigarrow \rho_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \omega_1 \rightarrow \omega_2) \bullet \tau : \omega_2 \rightsquigarrow \rho_1\, \rho_2 \dashv \Theta}$$

A-KAPP-TT-FORALL

$$\frac{\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{kapp}} (\rho_1 \mathbin{@}\widehat{\alpha} : \eta[a \mapsto \widehat{\alpha}]) \bullet \tau : \omega \rightsquigarrow \rho \dashv \Theta}{\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \forall a : \omega_1 . \eta) \bullet \tau : \omega \rightsquigarrow \rho \dashv \Theta}$$

A-KAPP-TT-FORALL-INFER

$$\frac{\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{kapp}} (\rho_1 \mathbin{@}\widehat{\alpha} : \eta[a \mapsto \widehat{\alpha}]) \bullet \tau : \omega \leadsto \rho \dashv \Theta}{\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \forall\{a : \omega_1\}.\eta) \bullet \tau : \omega \leadsto \rho \dashv \Theta}$$

A-KAPP-TT-KUVAR

$$\frac{\Delta_1, \widehat{\alpha}_1 : \star, \widehat{\alpha}_2 : \star, \widehat{\alpha} : \omega = (\widehat{\alpha}_1 \to \widehat{\alpha}_2), \Delta_2 \Vdash^{\mathsf{kc}} \tau \Leftarrow \widehat{\alpha}_1 \leadsto \rho_2 \dashv \Theta}{\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \Vdash^{\mathsf{kapp}} (\rho_1 : \widehat{\alpha}) \bullet \tau : \widehat{\alpha}_2 \leadsto \rho_1\,\rho_2 \dashv \Theta}$$

$\boxed{\Delta \Vdash^{\mathsf{ela}} \mu : \eta}$ *(Elaborated Kinding)*

A-ELA-STAR

$$\frac{}{\Delta \Vdash^{\mathsf{ela}} \star : \star}$$

A-ELA-KUVAR

$$\frac{(\widehat{\alpha} : \omega) \in \Delta}{\Delta \Vdash^{\mathsf{ela}} \widehat{\alpha} : [\Delta]\omega}$$

A-ELA-NAT

$$\frac{}{\Delta \Vdash^{\mathsf{ela}} \mathsf{Int} : \star}$$

A-ELA-VAR

$$\frac{(a : \omega) \in \Delta}{\Delta \Vdash^{\mathsf{ela}} a : [\Delta]\omega}$$

A-ELA-TCON

$$\frac{(T : \eta) \in \Delta}{\Delta \Vdash^{\mathsf{ela}} T : [\Delta]\eta}$$

A-ELA-ARROW

$$\frac{}{\Delta \Vdash^{\mathsf{ela}} \to : \star \to \star \to \star}$$

A-ELA-FORALL

$$\frac{\Delta \Vdash^{\mathsf{ela}} \omega : \star \qquad \Delta, a : \omega \Vdash^{\mathsf{ela}} \mu : \star}{\Delta \Vdash^{\mathsf{ela}} \forall a : \omega.\mu : \star}$$

A-ELA-FORALL-INFER

$$\frac{\Delta \Vdash^{\mathsf{ela}} \omega : \star \qquad \Delta, a : \omega \Vdash^{\mathsf{ela}} \mu : \star}{\Delta \Vdash^{\mathsf{ela}} \forall\{a : \omega\}.\mu : \star}$$

A-ELA-APP

$$\frac{\Delta \Vdash^{\mathsf{ela}} \rho_1 : \omega_1 \to \omega_2 \qquad \Delta \Vdash^{\mathsf{ela}} \rho_2 : \omega_1}{\Delta \Vdash^{\mathsf{ela}} \rho_1\,\rho_2 : \omega_2}$$

A-ELA-KAPP

$$\frac{\Delta \Vdash^{\mathsf{ela}} \rho_1 : \forall a : \omega.\eta \qquad \Delta \Vdash^{\mathsf{ela}} \rho_2 : \omega}{\Delta \Vdash^{\mathsf{ela}} \rho_1 \mathbin{@}\rho_2 : \eta[a \mapsto [\Delta]\rho_2]}$$

A-ELA-KAPP-INFER

$$\frac{\Delta \Vdash^{\mathsf{ela}} \rho_1 : \forall\{a : \omega\}.\eta \qquad \Delta \Vdash^{\mathsf{ela}} \rho_2 : \omega}{\Delta \Vdash^{\mathsf{ela}} \rho_1 \mathbin{@}\rho_2 : \eta[a \mapsto [\Delta]\rho_2]}$$

$\boxed{\Delta\ \mathsf{ok}}$ *(Well-formed Type Contexts)*

A-TCTX-EMPTY

$$\frac{}{\bullet\ \mathsf{ok}}$$

A-TCTX-TVAR-TT

$$\frac{\Delta\ \mathsf{ok} \qquad \Delta \Vdash^{\mathsf{ela}} \omega : \star}{\Delta, a : \omega\ \mathsf{ok}}$$

A-TCTX-TCON-TT

$$\frac{\Delta\ \mathsf{ok} \qquad \Delta \Vdash^{\mathsf{ela}} \eta : \star}{\Delta, T : \eta\ \mathsf{ok}}$$

A-TCTX-KUVAR-TT

$$\frac{\Delta\ \mathsf{ok} \qquad \Delta \Vdash^{\mathsf{ela}} \omega : \star}{\Delta, \widehat{\alpha} : \omega\ \mathsf{ok}}$$

A-TCTX-KUVARSOLVED-TT

$$\frac{\Delta\ \mathsf{ok} \qquad \Delta \Vdash^{\mathsf{ela}} \omega_2 : [\Delta]\omega_1}{\Delta, \widehat{\alpha} : \omega_1 = \omega_2\ \mathsf{ok}}$$

A-TCTX-LO

$$\frac{\Delta, \Delta^{\mathsf{lo}}\ \mathsf{ok}}{\Delta, \{\Delta^{\mathsf{lo}}\}\ \mathsf{ok}}$$

$\boxed{\Delta \Vdash^{\mathsf{ectx}} \Gamma}$ *(Well-formed Term Contexts)*

A-ECTX-EMPTY

$$\frac{}{\Delta \Vdash^{\mathsf{ectx}} \bullet}$$

A-ECTX-DCON-TT

$$\frac{\Delta \Vdash^{\mathsf{ectx}} \Gamma \qquad \Delta \Vdash^{\mathsf{ela}} \mu : \star}{\Delta \Vdash^{\mathsf{ectx}} \Gamma, D : \mu}$$

$\boxed{\Delta \Vdash^{\mathsf{u}} \omega_1 \approx \omega_2 \dashv \Theta}$ *(Unification)*

A-U-REFL-TT

$$\frac{}{\Delta \Vdash^{\mathsf{u}} \omega \approx \omega \dashv \Delta}$$

A-U-APP

$$\frac{\Delta \Vdash^{\mathsf{u}} \rho_1 \approx \rho_3 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{u}} [\Delta_1]\rho_2 \approx [\Delta_1]\rho_4 \dashv \Theta}{\Delta \Vdash^{\mathsf{u}} \rho_1\,\rho_2 \approx \rho_3\,\rho_4 \dashv \Theta}$$

A-U-KAPP

$$\frac{\Delta \Vdash^{\mathsf{u}} \rho_1 \approx \rho_3 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{u}} [\Delta_1]\rho_2 \approx [\Delta_1]\rho_4 \dashv \Theta}{\Delta \Vdash^{\mathsf{u}} \rho_1 \mathbin{@}\rho_2 \approx \rho_3 \mathbin{@}\rho_4 \dashv \Theta}$$

A-U-KVARL-TT

$$\frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta_1, \widehat{\alpha} : \omega_1, \Theta_2 \qquad \Theta_1 \Vdash^{\mathsf{ela}} \rho_2 : \omega_2 \qquad \Theta_1 \Vdash^{\mathsf{u}} [\Theta_1]\omega_1 \approx \omega_2 \dashv \Theta_3}{\Delta \Vdash^{\mathsf{u}} \widehat{\alpha} \approx \rho_1 \dashv \Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2}$$

A-U-KVARR-TT

$$\frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta_1, \widehat{\alpha} : \omega_1, \Theta_2 \qquad \Theta_1 \Vdash^{\mathsf{ela}} \rho_2 : \omega_2 \qquad \Theta_1 \Vdash^{\mathsf{u}} [\Theta_1]\omega_1 \approx \omega_2 \dashv \Theta_3}{\Delta \Vdash^{\mathsf{u}} \rho_1 \approx \widehat{\alpha} \dashv \Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2}$$

A-U-KVARL-LO-TT

$$\frac{\begin{array}{cc}\Delta_1, \Delta_2 \mathrel{+\!\!+}^{\mathsf{mv}} \widehat{\alpha} : \omega_1 \leadsto \Theta & \Delta[\{\Theta\}] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4 \\ \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{ela}} \rho_2 : \omega_2 & \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{u}} [\Theta_1, \Theta_2]\omega_1 \approx \omega_2 \dashv \Theta_5, \{\Theta_6\}\end{array}}{\Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}] \Vdash^{\mathsf{u}} \widehat{\alpha} \approx \rho_1 \dashv \Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4}$$

A-U-KVARR-LO-TT

$$\frac{\begin{array}{cc}\Delta_1, \Delta_2 \mathrel{+\!\!+}^{\mathsf{mv}} \widehat{\alpha} : \omega_1 \leadsto \Theta & \Delta[\{\Theta\}] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4 \\ \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{ela}} \rho_2 : \omega_2 & \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{u}} [\Theta_1, \Theta_2]\omega_1 \approx \omega_2 \dashv \Theta_5, \{\Theta_6\}\end{array}}{\Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}] \Vdash^{\mathsf{u}} \rho_1 \approx \widehat{\alpha} \dashv \Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4}$$

$\boxed{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \omega_2 \dashv \Theta}$ *(Promotion)*

A-PR-STAR

$$\frac{}{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \star \rightsquigarrow \star \dashv \Delta}$$

A-PR-ARR

$$\frac{}{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rightarrow \rightsquigarrow \rightarrow \dashv \Delta}$$

A-PR-TCON

$$\frac{}{\Delta[T][\widehat{\alpha}] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} T \rightsquigarrow T \dashv \Delta[T][\widehat{\alpha}]}$$

A-PR-NAT

$$\frac{}{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \mathsf{Int} \rightsquigarrow \mathsf{Int} \dashv \Delta}$$

A-PR-APP

$$\frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \rho_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} [\Delta_1]\omega_2 \rightsquigarrow \rho_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1\, \omega_2 \rightsquigarrow T \dashv \Theta}$$

A-PR-KAPP

$$\frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \rho_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} [\Delta_1]\omega_2 \rightsquigarrow \rho_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1\, @\omega_2 \rightsquigarrow T \dashv \Theta}$$

A-PR-TVAR

$$\frac{}{\Delta[a][\widehat{\alpha}] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} a \rightsquigarrow a \dashv \Delta[a][\widehat{\alpha}]}$$

A-PR-KUVARL

$$\frac{}{\Delta[\widehat{\beta}][\widehat{\alpha}] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \widehat{\beta} \rightsquigarrow \widehat{\beta} \dashv \Delta[\widehat{\beta}][\widehat{\alpha}]}$$

A-PR-KUVARR-TT

$$\frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} [\Delta]\rho \rightsquigarrow \rho_1 \dashv \Theta[\widehat{\alpha}][\widehat{\beta} : \rho]}{\Delta[\widehat{\alpha}][\widehat{\beta} : \rho] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \widehat{\beta} \rightsquigarrow \widehat{\beta}_1 \dashv \Theta[\widehat{\beta}_1 : \rho_1, \widehat{\alpha}][\widehat{\beta} : \rho = \widehat{\beta}_1]}$$

$\boxed{\Delta_1 \mathrel{+\!\!+}^{\mathsf{mv}} \Delta_2 \leadsto \Theta}$ *(Moving)*

A-MV-EMPTY

$$\frac{}{\bullet \mathrel{+\!\!+}^{\mathsf{mv}} \Delta \leadsto \Delta}$$

A-MV-KUVAR

$$\frac{\mathsf{vars}(\omega) \sharp \mathsf{dom}(\Delta_2) \qquad \Delta_1 \mathrel{+\!\!+}^{\mathsf{mv}} \Delta_2 \leadsto \Theta}{\widehat{\alpha} : \omega, \Delta_1 \mathrel{+\!\!+}^{\mathsf{mv}} \Delta_2 \leadsto \widehat{\alpha} : \omega, \Theta}$$

A-MV-KUVARM

$$\frac{\neg(\mathsf{vars}(\omega) \sharp \mathsf{dom}(\Delta_2)) \qquad \Delta_1 \mathrel{+\!\!+}^{\mathsf{mv}} \Delta_2, \widehat{\alpha} : \omega \leadsto \Theta}{\widehat{\alpha} : \omega, \Delta_1 \mathrel{+\!\!+}^{\mathsf{mv}} \Delta \leadsto \Theta}$$

A-MV-TVAR

$$\frac{\mathsf{vars}(\omega) \sharp \mathsf{dom}(\Delta_2) \qquad \Delta_1 \mathrel{+\!\!+}^{\mathsf{mv}} \Delta_2 \leadsto \Theta}{a : \omega, \Delta_1 \mathrel{+\!\!+}^{\mathsf{mv}} \Delta_2 \leadsto a : \omega, \Theta}$$

A-MV-TVARM

$$\frac{\neg(\mathsf{vars}(\omega) \sharp \mathsf{dom}(\Delta_2)) \qquad \Delta_1 \mathrel{+\!\!+}^{\mathsf{mv}} \Delta_2, a : \omega \leadsto \Theta}{a : \omega, \Delta_1 \mathrel{+\!\!+}^{\mathsf{mv}} \Delta_2 \leadsto \Theta}$$

C.7 Context Application in PolyKinds

$[\Delta]\eta$ applies Δ as a substitution to η.

$$\begin{array}{lcl}
[\Delta]\star & = & \star \\
[\Delta]\mathsf{Int} & = & \mathsf{Int} \\
[\Delta]a & = & a \\
[\Delta]T & = & T \\
[\Delta] \rightarrow & = & \rightarrow \\
[\Delta]\forall a : \omega.\eta & = & \forall a : [\Delta]\omega.[\Delta]\eta \\
[\Delta](\rho_1\, \rho_2) & = & ([\Delta]\rho_1)\,([\Delta]\rho_2) \\
[\Delta](\rho_1\, @\rho_2) & = & ([\Delta]\rho_1)\, @([\Delta]\rho_2) \\
[\Delta[\widehat{\alpha}]]\widehat{\alpha} & = & \widehat{\alpha} \\
[\Delta[\widehat{\alpha} : \omega = \rho]]\widehat{\alpha} & = & [\Delta[\widehat{\alpha} : \omega = \rho]]\rho
\end{array}$$

$$
\begin{array}{lcl}
\multicolumn{3}{l}{[\Delta]\Gamma \text{ applies } \Delta \text{ as a substitution to } \Gamma.} \\
[\Omega]\bullet & = & \bullet \\
[\Omega](\Gamma, D : \mu) & = & [\Omega]\Gamma, D : [\Omega]\mu
\end{array}
$$

$$
\begin{array}{lcll}
\multicolumn{4}{c}{[\Omega]\Delta \text{ applies } \Omega \text{ as a substitution to } \Delta.} \\
[\Omega]\bullet & = & \bullet & \\
[\Omega, a : \omega](\Delta, a : \omega) & = & [\Omega]\Delta, a : [\Omega]\omega & \\
[\Omega, T : \omega](\Delta, T : \omega) & = & [\Omega]\Delta, T : [\Omega]\omega & \\
[\Omega, \widehat{\alpha} : \omega = \rho](\Delta, \widehat{\alpha} : \omega) & = & [\Omega]\Delta & \\
[\Omega, \widehat{\alpha} : \omega = \rho_1](\Delta, \widehat{\alpha} : \omega = \rho_2) & = & [\Omega]\Delta & \text{if } [\Omega]\rho_1 = [\Omega]\rho_2 \\
[\Omega, \widehat{\alpha} : \omega = \rho]\Delta & = & [\Omega]\Delta & \text{if } \widehat{\alpha} \notin \Delta \\
[\Omega, \blacktriangleright_D](\Delta, \blacktriangleright_D) & = & [\Omega]\Delta & \\
[\Omega, \{\Omega_1\}](\Delta, \{\Delta_1\}) & = & [\Omega, \Omega_1](\Delta, \Delta') & \\
 & & \text{where } \Delta' = \mathsf{topo}\,(\Delta_1) &
\end{array}
$$

C.8 Context Extension in PolyKinds

$\boxed{\Delta \longrightarrow \Theta}$ *(Context Extension)*

$$
\textsc{a-ctxe-empty}\ \frac{}{\bullet \longrightarrow \bullet}
\qquad
\textsc{a-ctxe-tvar-tt}\ \frac{\Delta \longrightarrow \Theta}{\Delta, a : \omega \longrightarrow \Theta, a : \omega}
\qquad
\textsc{a-ctxe-tcon-tt}\ \frac{\Delta \longrightarrow \Theta}{\Delta, T : \eta \longrightarrow \Theta, T : \eta}
\qquad
\textsc{a-ctxe-kuvar-tt}\ \frac{\Delta \longrightarrow \Theta}{\Delta, \widehat{\alpha} : \omega \longrightarrow \Theta, \widehat{\alpha} : \omega}
$$

$$
\textsc{a-ctxe-kuvarSolved-tt}\ \frac{\Delta \longrightarrow \Theta \qquad [\Theta]\rho_1 = [\Theta]\rho_2}{\Delta, \widehat{\alpha} : \omega = \rho_1 \longrightarrow \Theta, \widehat{\alpha} : \omega = \rho_2}
\qquad
\textsc{a-ctxe-solve-tt}\ \frac{\Delta \longrightarrow \Theta \qquad \Theta \Vdash^{\mathsf{ela}} \rho : [\Theta]\omega}{\Delta, \widehat{\alpha} : \omega \longrightarrow \Theta, \widehat{\alpha} : \omega = \rho}
\qquad
\textsc{a-ctxe-add-tt}\ \frac{\Delta \longrightarrow \Theta \qquad \Theta \Vdash^{\mathsf{ela}} \omega : \star}{\Delta \longrightarrow \Theta, \widehat{\alpha} : \omega}
$$

$$
\textsc{a-ctxe-addSolved-tt}\ \frac{\Delta \longrightarrow \Theta \qquad \Theta \Vdash^{\mathsf{ela}} \rho : [\Theta]\omega}{\Delta \longrightarrow \Theta, \widehat{\alpha} : \omega = \rho}
\qquad
\textsc{a-ctxe-marker}\ \frac{\Delta \longrightarrow \Theta}{\Delta, \blacktriangleright_D \longrightarrow \Theta, \blacktriangleright_D}
\qquad
\textsc{a-ctxe-lo}\ \frac{\Delta \longrightarrow \Theta \qquad \Delta, \mathsf{topo}\,(\Delta_1) \longrightarrow \Theta, \Theta_1}{\Delta, \{\Delta_1\} \longrightarrow \Theta, \{\Theta_1\}}
$$

D PROOF FOR HASKELL98

D.1 List of Lemmas

D.1.1 Well-formedness of Declarative Type System.

Lemma D.1 (Well-formedness of Declarative Typing Data Constructor Declarations). *If* $\Sigma \vdash^{\mathsf{dc}}_{\tau_1} \mathcal{D} \rightsquigarrow \tau_2$*, then* $\Sigma \vdash^{\mathsf{k}} \tau_2 : \star$*.*

Lemma D.2 (Well-formedness of Declarative Typing Datatype Declaration). *If* $\Sigma \vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow \Psi$*, then* $\Sigma \vdash \Psi$*.*

D.1.2 Well-formedness of Algorithmic Type System.

Lemma D.3 (Well-formedness of Promotion). *If* $\Delta_1, \widehat{\alpha}, \Delta_2$ ok*, and* $\Delta_1, \widehat{\alpha}, \Delta_2 \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa_1 \rightsquigarrow \kappa_2 \dashv \Theta$*, then* $\Theta = \Theta_1, \widehat{\alpha}, \Theta_2$*, and* $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta$*, and* $\Theta_1 \Vdash^{\mathsf{kv}} \kappa_2$*, and* Θ ok*.*

Lemma D.4 (Well-formedness of Kind Unification). *If* Δ ok*, and* $\Delta \Vdash^{\mathsf{u}} \kappa_1 \approx \kappa_2 \dashv \Theta$*, then* $\Delta \longrightarrow \Theta$*, and* Θ ok*.*

Lemma D.5 (Well-formedness of Application Kinding). *If* Δ ok*, and* $\Delta \Vdash^{\mathsf{kapp}} \kappa_1 \bullet \kappa_2 : \kappa \dashv \Theta$*, then* $\Delta \longrightarrow \Theta$*, and* Θ ok*. Moreover, if* $\Delta \Vdash^{\mathsf{kv}} \kappa_1$*, then we have* $\Theta \Vdash^{\mathsf{kv}} \kappa$*.*

Lemma D.6 (Well-formedness of Kinding). *If* Δ ok, *and* $\Delta \Vdash^{k} \sigma : \kappa \dashv \Theta$, *then* $\Delta \longrightarrow \Theta$, *and* Θ ok, *and* $\Theta \Vdash^{kv} \kappa$ *and* $\Theta \Vdash^{kc} \sigma \Leftarrow \kappa$.

Lemma D.7 (Well-formedness of Typing Data Constructor Declarations). *If* Δ ok, *and* $\Delta \Vdash^{dc}_{\tau'} \mathcal{D} \rightsquigarrow \tau \dashv \Theta$, *then* $\Delta \longrightarrow \Theta$, *and* Θ ok. *and* $\Theta \Vdash^{kc} \tau \Leftarrow \star$.

Lemma D.8 (Well-formedness of Typing Datatype Declaration). *If* Δ ok, *and* $\Delta \Vdash^{dt} \mathcal{T} \rightsquigarrow \Gamma \dashv \Theta$, *then* $\Delta \longrightarrow \Theta$, *and* Θ ok, *and* $\Theta \Vdash^{ectx} \Gamma$.

D.1.3 Properties of Context Extension.

Lemma D.9 (Declaration Preservation). *If* $\Delta \longrightarrow \Theta$, *if a type constructor or a type variable or a kind unification variable is declared in* Δ, *then it is declared in* Θ.

Lemma D.10 (Extension Weakening). *Given* $\Delta \longrightarrow \Theta$,

- *if* $\Delta \Vdash^{kv} \kappa$, *then* $\Theta \Vdash^{kv} \kappa$;
- *if* $\Delta \Vdash^{kc} \sigma \Leftarrow \kappa$, *then* $\Theta \Vdash^{kc} \sigma \Leftarrow \kappa$.

Definition D.11 (Contextual Size).

$$\begin{array}{lcl}
\mid \Delta \vdash \star \mid & = & 1 \\
\mid \Delta \vdash \kappa_1 \to \kappa_2 \mid & = & 1+ \mid \Delta \vdash \kappa_1 \mid + \mid \Delta \vdash \kappa_2 \mid \\
\mid \Delta[\widehat{\alpha}] \vdash \widehat{\alpha} \mid & = & 1 \\
\mid \Delta[\widehat{\alpha} = \kappa] \vdash \widehat{\alpha} \mid & = & 1+ \mid \Delta[\widehat{\alpha} = \kappa] \vdash \kappa \mid
\end{array}$$

Lemma D.12 (Substitution Kinding). *If* Δ ok, *and* $\Delta \Vdash^{kv} \kappa$, *then* $\Delta \Vdash^{kv} [\Delta]\kappa$.

Lemma D.13 (Context Extension with Defaulting is Context Extension). *If* $\Delta \twoheadrightarrow \Theta$, *then* $\Delta \longrightarrow \Theta$.

Lemma D.14 (Reflexivity of Context Extension). *If* Δ ok, *then* $\Delta \longrightarrow \Delta$.

Lemma D.15 (Well-formedness of Context Extension). *If* Δ ok, *and* $\Delta \longrightarrow \Theta$, *then* Θ ok.

Definition D.16 (Softness). *A context* Δ *is soft iff it contains only of* $\widehat{\alpha}$ *and* $\widehat{\alpha} = \kappa$ *declarations.*

Lemma D.17 (Extension Order).

(1) *If* $\Delta_1, a : \kappa, \Delta_2 \longrightarrow \Theta$, *then* $\Theta = \Theta_1, a : \kappa, \Theta_2$, *where* $\Delta_1 \longrightarrow \Theta_1$. *Moreover, if* Δ_2 **soft**, *then* Θ_2 **soft**.
(2) *If* $\Delta_1, T : \kappa, \Delta_2 \longrightarrow \Theta$, *then* $\Theta = \Theta_1, T : \kappa, \Theta_2$, *where* $\Delta_1 \longrightarrow \Theta_1$. *Moreover, if* Δ_2 **soft**, *then* Θ_2 **soft**.
(3) *If* $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta$, *then* $\Theta = \Theta_1, \Theta', \Theta_2$, *where* $\Delta_1 \longrightarrow \Theta_1$, *and* Θ' *is either* $\widehat{\alpha}$ *or* $\widehat{\alpha} = \kappa$ *for some* κ. *Moreover, if* Δ_2 **soft**, *then* Θ_2 **soft**.
(4) *If* $\Delta_1, \widehat{\alpha} = \kappa_1, \Delta_2 \longrightarrow \Theta$, *then* $\Theta = \Theta_1, \widehat{\alpha} = \kappa_2, \Theta_2$, *where* $\Delta_1 \longrightarrow \Theta_1$, *and* $[\Theta_1]\kappa_1 = [\Theta_1]\kappa_2$. *Moreover, if* Δ_2 **soft**, *then* Θ_2 **soft**.

Lemma D.18 (Substitution Extension Invariance). *If* Δ ok, *and* $\Delta \Vdash^{kv} \kappa$, *and* $\Delta \longrightarrow \Theta$, *then* $[\Theta]\kappa = [\Theta]([\Delta]\kappa)$ *and* $[\Theta]\kappa = [\Delta]([\Theta]\kappa)$.

Lemma D.19 (Transitivity of Context Extension). *If* Δ' ok, *and* $\Delta' \longrightarrow \Delta$, *and* $\Delta \longrightarrow \Theta$, *then* $\Delta' \longrightarrow \Theta$.

Lemma D.20 (Solution Admissibility for Extension). *If* $\Delta_1, \widehat{\alpha}, \Delta_2$ ok *and* $\Delta_1 \Vdash^{\mathsf{kv}} \kappa$*, then* $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha} = \kappa, \Delta_2$.

Lemma D.21 (Solved Variable Addition for Extension). *If* Δ_1, Δ_2 ok *and* $\Delta_1 \Vdash^{\mathsf{kv}} \kappa$*, then* $\Delta_1, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha} = \kappa, \Delta_2$.

Lemma D.22 (Unsolved Variable Addition). *If* Δ_1, Δ_2 ok *then* $\Delta_1, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha}, \Delta_2$.

Lemma D.23 (Parallel Admissibility). *If* $\Delta_1 \longrightarrow \Theta_1$*, and* Δ_1, Δ_2 ok*, and* $\Delta_1, \Delta_2 \longrightarrow \Theta_1, \Theta_2$*, and* Δ_2 *is fresh w.r.t.* Θ_1*, then:*

- $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha}, \Theta_2$
- *If* $\Theta_1 \Vdash^{\mathsf{kv}} \kappa$*, then* $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} = \kappa, \Theta_2$
- *If* $[\Theta_1]\kappa_1 = [\Theta_1]\kappa_2$*, then* $\Delta_1, \widehat{\alpha} = \kappa_1, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} = \kappa_2, \Theta_2$

Lemma D.24 (Parallel Extension Solution). *If* $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} = \kappa_2, \Theta_2$*, and* $[\Theta_1]\kappa_1 = [\Theta_1]\kappa_2$*, then* $\Delta_1, \widehat{\alpha} = \kappa_1, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} = \kappa_2, \Theta_2$.

Lemma D.25 (Parallel Variable Update). *If* $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} = \kappa, \Theta_2$*, and* $\Delta_1 \Vdash^{\mathsf{kv}} \kappa_1$*, and* $\Theta_1 \Vdash^{\mathsf{kv}} \kappa_2$*, and* $[\Theta_1]\kappa = [\Theta_1]\kappa_1 = [\Theta_1]\kappa_2$*, then* $\Delta_1, \widehat{\alpha} = \kappa_1, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} = \kappa_2, \Theta_2$

D.1.4 Properties of Complete Context.

Lemma D.26 (Finishing Kinding). *If* Ω ok*, and* $\Omega \Vdash^{\mathsf{kv}} \kappa$*, and* $\Omega \longrightarrow \Omega'$*, then* $[\Omega]\kappa = [\Omega']\kappa$.

Lemma D.27 (Finishing Term Contexts). *If* Ω ok*, and* $\Omega \Vdash^{\mathsf{ectx}} \Gamma$*, and* $\Omega \longrightarrow \Omega'$*, then* $[\Omega']\Gamma = [\Omega]\Gamma$.

Lemma D.28 (Stability of Complete Contexts). *If* $\Delta \longrightarrow \Omega$*, then* $[\Omega]\Delta = [\Omega]\Omega$.

Lemma D.29 (Softness Goes Away). *If* $\Delta_1, \Delta_2 \longrightarrow \Omega_1, \Omega_2$ *where* $\Delta_1 \longrightarrow \Omega_1$*, and* Δ_2 **soft***, then* $[\Omega_1, \Omega_2](\Delta_1, \Delta_2) = [\Omega_1]\Delta_1$.

Lemma D.30 (Confluence of Completeness). *If* $\Delta_1 \longrightarrow \Omega$*, and* $\Delta_2 \longrightarrow \Omega$*, then* $[\Omega]\Delta_1 = [\Omega]\Delta_2$.

Lemma D.31 (Substitution Stability). *If* Ω, Δ ok*, and* $\Omega \Vdash^{\mathsf{kv}} \kappa$*, then* $[\Omega]\kappa = [\Omega, \Delta]\kappa$.

Lemma D.32 (Type Constructor Preservation). *If* Δ ok*, then* $(T : \kappa) \in \Delta$*, and* $\Delta \longrightarrow \Omega$*, then* $(T : [\Omega]\kappa) \in [\Omega]\Delta$.

Lemma D.33 (Type Variable Preservation). *If* $(a : \kappa) \in \Delta$*, and* $\Delta \longrightarrow \Omega$*, then* $(a : [\Omega]\kappa) \in [\Omega]\Delta$.

Lemma D.34 (Finishing Completions). *If* Ω ok*, and* $\Omega \longrightarrow \Omega'$*, then* $[\Omega]\Omega = [\Omega']\Omega'$.

D.1.5 Soundness of Algorithm.

Lemma D.35 (Soundness of Kind Validating). *If* Ω ok, *and* $\Omega \Vdash^{\mathsf{kv}} \kappa$*, then* $[\Omega]\kappa$ *is a validate kind in the declarative system.*

Lemma D.36 (Soundness of Well-formed Type Context). *If* Δ ok, *and* $\Delta \longrightarrow \Omega$*, then* $[\Omega]\Delta$ *is a valid type context in the declarative system.*

Lemma D.37 (Soundness of Well-formed Term Context). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{ectx}} \Gamma$*, and* $\Delta \longrightarrow \Omega$*, then* $[\Omega]\Delta \Vdash^{\mathsf{ectx}} [\Omega]\Gamma$.

Lemma D.38 (Soundness of Promotion). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa_1 \rightsquigarrow \kappa_2 \dashv \Theta$*, then* $[\Theta]\kappa_1 = [\Theta]\kappa_2$*. If* $\Theta \longrightarrow \Omega$*, then* $[\Omega]\kappa_1 = [\Omega]\kappa_2$.

Lemma D.39 (Soundness of Unification). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{u}} \kappa_1 \approx \kappa_2 \dashv \Theta$*, then* $[\Theta]\kappa_1 = [\Theta]\kappa_2$*. If* $\Theta \longrightarrow \Omega$*, then* $[\Omega]\kappa_1 = [\Omega]\kappa_2$.

Lemma D.40 (Soundness of Application Kinding). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{kapp}} \kappa_1 \bullet \kappa_2 : \kappa_3 \dashv \Theta$*, then* $[\Theta]\kappa_1 = [\Theta]\kappa_2 \to [\Theta]\kappa_3$*. If* $\Theta \longrightarrow \Omega$*, then* $[\Omega]\kappa_1 = [\Omega]\kappa_2 \to [\Omega]\kappa_3$.

Lemma D.41 (Soundness of Kinding). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{k}} \sigma : \kappa \dashv \Theta$*, and* $\Theta \longrightarrow \Omega$*, then* $[\Omega]\Delta \vdash^{\mathsf{k}} [\Omega]\sigma : [\Omega]\kappa$.

Lemma D.42 (Soundness of Typing Data Constructor Declaration). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{dc}}_{\tau'} \mathcal{D} \rightsquigarrow \tau \dashv \Theta$*, and* $\Theta \longrightarrow \Omega$*, then* $[\Omega]\Delta \vdash^{\mathsf{dc}}_{\tau'} \mathcal{D} \rightsquigarrow \tau$.

Lemma D.43 (Soundness of Typing Datatype Declaration). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow \Gamma \dashv \Theta$*, and* $\Theta \longrightarrow \Omega$*, then* $[\Omega]\Delta \vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow [\Omega]\Gamma$.

Lemma D.44 (Soundness of Typing Program). *If* Ω ok, *and* $\Omega \Vdash^{\mathsf{ectx}} \Gamma$*, and* $\Omega; \Gamma \Vdash^{\mathsf{pgm}} pgm : \sigma$*, then* $[\Omega]\Omega; [\Omega]\Gamma \vdash^{\mathsf{pgm}} pgm : \sigma$.

D.1.6 Completeness of Algorithm.

Lemma D.45 (Completeness of Promotion). *Given* Δ ok, *and* $\Delta \longrightarrow \Omega$*, and* $\Delta \vdash \widehat{\alpha}$*, and* $\Delta \Vdash^{\mathsf{kv}} \kappa$*, and* $[\Delta]\widehat{\alpha} = \widehat{\alpha}$*, and* $[\Delta]\kappa = \kappa$*, if* κ *is free of* $\widehat{\alpha}$*, then there exists* κ_2*,* Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$*, and* $\Omega \longrightarrow \Omega'$*, and* $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa \rightsquigarrow \kappa_2 \dashv \Theta$.

Lemma D.46 (Completeness of Unification). *Given* Δ ok, *and* $\Delta \longrightarrow \Omega$*, and* $\Delta \Vdash^{\mathsf{kv}} \kappa_1$ *and* $\Delta \Vdash^{\mathsf{kv}} \kappa_2$*, and* $[\Delta]\kappa_1 = \kappa_1$ *and* $[\Delta]\kappa_2 = \kappa_2$*, if* $[\Omega]\kappa_1 = [\Omega]\kappa_2$*, then there exists* Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$*, and* $\Omega \longrightarrow \Omega'$ *and* $\Delta \Vdash^{\mathsf{u}} \kappa_1 \approx \kappa_2 \dashv \Theta$.

Lemma D.47 (Completeness of Application Kinding). *Given* Δ ok, *and* $\Delta \longrightarrow \Omega$*, and* $\Delta \Vdash^{\mathsf{kv}} \kappa$ *and* $\Delta \Vdash^{\mathsf{kv}} \kappa'$*, and* $[\Delta]\kappa = \kappa$ *and* $[\Delta]\kappa' = \kappa'$*, if* $[\Omega]\kappa = [\Omega]\kappa' \to \kappa_1$*, then there exists* κ_2*,* Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$*, and* $\Omega \longrightarrow \Omega'$ *and* $\Delta \Vdash^{\mathsf{kapp}} \kappa \bullet \kappa' : \kappa_2 \dashv \Theta$*, and* $[\Omega']\kappa_2 = \kappa_1$.

Lemma D.48 (Completeness of Kinding). *Given* Δ ok *and* $\Delta \longrightarrow \Omega$*, if* $[\Omega]\Delta \vdash^{\mathsf{k}} [\Omega]\sigma : \kappa$*, then there exists* Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$*, and* $\Omega \longrightarrow \Omega'$ *and* $\Delta \Vdash^{\mathsf{k}} \sigma : \kappa' \dashv \Theta$*, and* $[\Omega']\kappa' = \kappa$.

Lemma D.49 (Completeness of Typing Data Constructor Declaration). *Given* Δ ok *and* $\Delta \longrightarrow \Omega$, *if* $[\Omega]\Delta \vdash^{\mathsf{dc}}_{\tau'} \mathcal{D} \leadsto \tau$, *then there exists* Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$ *and* $\Delta \Vdash^{\mathsf{dc}}_{\tau'} \mathcal{D} \leadsto \tau \dashv \Theta$.

Lemma D.50 (Completeness of Typing Datatype Declaration). *Given* Δ ok, *and* $\Delta \longrightarrow \Omega$, *if* $[\Omega]\Delta \vdash^{\mathsf{dt}} \mathcal{T} \leadsto \Psi$, *then there exists* Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$ *and* $\Delta \Vdash^{\mathsf{dt}} \mathcal{T} \leadsto \Gamma \dashv \Theta$ *and* $\Psi = [\Omega']\Gamma$.

Theorem D.51 (Completeness of Typing a Group). *Given* Ω ok, *if* $[\Omega]\Omega \vdash^{\mathsf{grp}}$ $\mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto \overline{\kappa_i}^{\,i}; \overline{\Psi_i}^{\,i}$, *then there exists* $\overline{\kappa_i'}^{\,i}, \overline{\Gamma_i}^{\,i}, \Theta$, *and* Ω', *such that* $\Omega \Vdash^{\mathsf{grp}}$ $\mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto \overline{\kappa_i'}^{\,i}; \overline{\Gamma_i}^{\,i} \dashv \Theta$, *where* $\Theta \longrightarrow \Omega'$, *and* $\overline{[\Omega']\kappa_i' = \kappa_i}^{\,i}$, *and* $\overline{\Psi_i = [\Omega']\Gamma_i}^{\,i}$.

D.2 Proofs

D.2.1 Well-formedness of Declarative Type System.

Lemma D.1 (Well-formedness of Declarative Typing Data Constructor Declarations). *If* $\Sigma \vdash^{\mathsf{dc}}_{\tau_1} \mathcal{D} \leadsto \tau_2$, *then* $\Sigma \vdash^{\mathsf{k}} \tau_2 : \star$.

Proof. We have

$$\textsc{dc-decl}\ \frac{\Sigma \vdash^{\mathsf{k}} \overline{\tau_i}^{\,i} \rightarrow \tau : \star}{\Sigma \vdash^{\mathsf{dc}}_{\tau} D\,\overline{\tau_i}^{\,i} \leadsto \overline{\tau_i}^{\,i} \rightarrow \tau}$$

The goal follows trivially.

□

Lemma D.2 (Well-formedness of Declarative Typing Datatype Declaration). *If* $\Sigma \vdash^{\mathsf{dt}} \mathcal{T} \leadsto \Psi$, *then* $\Sigma \vdash \Psi$.

Proof. We have

$$\textsc{dt-decl}\ \frac{(T : \overline{\kappa_i}^{\,i} \rightarrow \star) \in \Sigma \qquad \overline{\Sigma,\ \overline{a_i : \kappa_i}^{\,i} \vdash^{\mathsf{dc}}_{T\,\overline{a_i}^{\,i}} \mathcal{D}_j \leadsto \tau_j}^{\,j}}{\Sigma \vdash^{\mathsf{dt}} \mathbf{data}\ T\,\overline{a_i}^{\,i} = \overline{\mathcal{D}_j}^{\,j} \leadsto \overline{D_j : \forall \overline{a_i : \kappa_i}^{\,i}.\tau_j}^{\,j}}$$

$$\begin{array}{l|l}
\overline{\Sigma,\ \overline{a_i : \kappa_i}^{\,i} \vdash^{\mathsf{k}} \tau_j : \star}^{\,j} & \text{By Lemma D.1} \\
\overline{\Sigma \vdash^{\mathsf{k}} \forall \overline{a_i : \kappa_i}^{\,i}.\tau_j : \star}^{\,j} & \text{By rule } \textsc{k-forall} \\
\Sigma \vdash \overline{D_j : \forall \overline{a_i : \kappa_i}^{\,i}.\tau_j}^{\,j} & \text{By rule } \textsc{ectx-dcon}
\end{array}$$

□

D.2.2 Well-formedness of Algorithmic Type System. Because by Lemma D.15 we know that if Δ ok, and $\Delta \longrightarrow \Theta$, it follows that Θ ok. So in the following lemma when there is the goal Θ ok, we prove Δ ok and $\Delta \longrightarrow \Theta$.

Lemma D.3 (Well-formedness of Promotion). *If* $\Delta_1, \widehat{\alpha}, \Delta_2$ ok, *and* $\Delta_1, \widehat{\alpha}, \Delta_2 \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa_1 \leadsto \kappa_2 \dashv \Theta$, *then* $\Theta = \Theta_1, \widehat{\alpha}, \Theta_2$, *and* $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta$, *and* $\Theta_1 \Vdash^{\mathsf{kv}} \kappa_2$, *and* Θ ok.

Proof. By induction on promotion.

- Case

$$\textsc{a-pr-star}\quad \frac{}{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \star \leadsto \star \dashv \Delta}$$

The goals hold trivially.

- Case

$$\textsc{a-pr-arrow}\quad \frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa_1 \leadsto \kappa_3 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} [\Delta_1]\kappa_2 \leadsto \kappa_4 \dashv \Theta}{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa_1 \;\rightarrow\; \kappa_2 \leadsto \kappa_3 \;\rightarrow\; \kappa_4 \dashv \Theta}$$

$$\begin{array}{l|l}
\Delta \longrightarrow \Delta_1 \wedge \Delta_1 = \Delta_{11}, \widehat{\alpha}, \Delta_{12} \wedge \Delta_{11} \Vdash^{\mathsf{kv}} \kappa_3 & \text{I.H.} \\
\Delta_1 \longrightarrow \Theta \wedge \Theta = \Theta_1, \widehat{\alpha}, \Theta_2 \wedge \Theta_1 \Vdash^{\mathsf{kv}} \kappa_4 & \text{I.H.} \\
\Delta \longrightarrow \Theta & \text{By Lemma D.19} \\
\Delta_{11} \longrightarrow \Theta_1 & \text{By Lemma D.17} \\
\Theta_1 \Vdash^{\mathsf{kv}} \kappa_3 & \text{By Lemma D.10} \\
\Theta_1 \Vdash^{\mathsf{kv}} \kappa_3 \;\rightarrow\; \kappa_4 & \text{By rule } \textsc{a-kv-arrow}
\end{array}$$

- Case

$$\textsc{a-pr-kuvarL}\quad \frac{}{\Delta[\widehat{\beta}][\widehat{\alpha}] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \widehat{\beta} \leadsto \widehat{\beta} \dashv \Delta[\widehat{\beta}][\widehat{\alpha}]}$$

The goals hold trivially.

- Case

$$\textsc{a-pr-kuvarR}\quad \frac{}{\Delta[\widehat{\alpha}][\widehat{\beta}] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \widehat{\beta} \leadsto \widehat{\beta}_1 \dashv \Delta[\widehat{\beta}_1, \widehat{\alpha}][\widehat{\beta} = \widehat{\beta}_1]}$$

Most goals hold trivially. By Lemmas D.22 and D.23 and transitivity (Lemma D.19) we can prove $\Delta[\widehat{\alpha}][\widehat{\beta}] \longrightarrow \Delta[\widehat{\beta}_1, \widehat{\alpha}][\widehat{\beta} = \widehat{\beta}_1]$.

□

Lemma D.4 (Well-formedness of Kind Unification). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{u}} \kappa_1 \approx \kappa_2 \dashv \Theta$, *then* $\Delta \longrightarrow \Theta$, *and* Θ ok.

Proof. By induction on the derivation of kind unification.

- Case

$$\textsc{a-u-refl}\quad \frac{}{\Delta \Vdash^{\mathsf{u}} \kappa \approx \kappa \dashv \Delta}$$

$$\begin{array}{l|l}
\Delta \longrightarrow \Delta & \text{By Lemma D.14}
\end{array}$$

- Case

$$\textsc{a-u-arrow}\quad \frac{\Delta \Vdash^{\mathsf{u}} \kappa_1 \approx \kappa_3 \dashv \Theta_1 \qquad \Theta_1 \Vdash^{\mathsf{u}} [\Theta_1]\kappa_2 \approx [\Theta_1]\kappa_4 \dashv \Theta}{\Delta \Vdash^{\mathsf{u}} \kappa_1 \;\rightarrow\; \kappa_2 \approx \kappa_3 \;\rightarrow\; \kappa_4 \dashv \Theta}$$

$$\begin{array}{l|l}
\Delta \longrightarrow \Theta_1 & \text{By I.H.} \\
\Theta_1 \longrightarrow \Theta & \text{By I.H.} \\
\Delta \longrightarrow \Theta & \text{By Lemma D.19}
\end{array}$$

- Case

$$\begin{array}{c}\textsc{a-u-kvarL}\\ \dfrac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa \leadsto \kappa_2 \dashv \Theta[\widehat{\alpha}]}{\Delta[\widehat{\alpha}] \Vdash^{\mathsf{u}} \widehat{\alpha} \approx \kappa \dashv \Theta[\widehat{\alpha} = \kappa_2]}\end{array}$$

$$\begin{array}{l|l}
\Theta = \Theta_1, \widehat{\alpha}, \Theta_2 \land \Delta \longrightarrow \Theta[\widehat{\alpha}] \land \Theta_1 \Vdash^{\mathsf{kv}} \kappa_2 & \text{By Lemma D.3}\\
\Theta \longrightarrow \Theta[\widehat{\alpha} = \kappa_2] & \text{By Lemma D.23}\\
\Delta \longrightarrow \Theta[\widehat{\alpha} = \kappa_2] & \text{By Lemma D.19}
\end{array}$$

- Case

$$\begin{array}{c}\textsc{a-u-kvarR}\\ \dfrac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa \leadsto \kappa_2 \dashv \Theta[\widehat{\alpha}]}{\Delta[\widehat{\alpha}] \Vdash^{\mathsf{u}} \kappa \approx \widehat{\alpha} \dashv \Theta[\widehat{\alpha} = \kappa_2]}\end{array}$$

Similar to the previous case.

□

Lemma D.5 (Well-formedness of Application Kinding). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{kapp}} \kappa_1 \bullet \kappa_2 : \kappa \dashv \Theta$, *then* $\Delta \longrightarrow \Theta$, *and* Θ ok. *Moreover, if* $\Delta \Vdash^{\mathsf{kv}} \kappa_1$, *then we have* $\Theta \Vdash^{\mathsf{kv}} \kappa$.

Proof. By induction on the derivation of application kinding.

- Case

$$\begin{array}{c}\textsc{a-kapp-kuvar}\\ \dfrac{\Delta[\widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha} = \widehat{\alpha}_1 \to \widehat{\alpha}_2] \Vdash^{\mathsf{u}} \widehat{\alpha}_1 \approx \kappa \dashv \Theta}{\Delta[\widehat{\alpha}] \Vdash^{\mathsf{kapp}} \widehat{\alpha} \bullet \kappa : \widehat{\alpha}_2 \dashv \Theta}\end{array}$$

$$\begin{array}{l|l}
\Delta[\widehat{\alpha}] \longrightarrow \Delta[\widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha}] & \text{By Lemma D.22}\\
\Delta[\widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha}] \longrightarrow \Delta[\widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha} = \widehat{\alpha}_1 \to \widehat{\alpha}_2] & \text{By Lemma D.20}\\
\Delta[\widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha} = \widehat{\alpha}_1 \to \widehat{\alpha}_2] \longrightarrow \Theta & \text{By Lemma D.4}\\
\Delta \longrightarrow \Theta & \text{By Lemma D.19}\\
\Delta[\widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha} = \widehat{\alpha}_1 \to \widehat{\alpha}_2] \vdash \widehat{\alpha}_2 & \text{By rule } \textsc{a-kv-kuvar}\\
\Theta \vdash \widehat{\alpha}_2 & \text{By Lemma D.10}
\end{array}$$

- Case

$$\begin{array}{c}\textsc{a-kapp-arrow}\\ \dfrac{\Delta \Vdash^{\mathsf{u}} \kappa_1 \approx \kappa \dashv \Theta}{\Delta \Vdash^{\mathsf{kapp}} \kappa_1 \to \kappa_2 \bullet \kappa : \kappa_2 \dashv \Theta}\end{array}$$

$$\begin{array}{l|l}
\Delta \longrightarrow \Theta & \text{By Lemma D.4}\\
\Delta \Vdash^{\mathsf{kv}} \kappa_1 \to \kappa_2 & \text{Given}\\
\Delta \Vdash^{\mathsf{kv}} \kappa_2 & \text{By inversion}\\
\Theta \Vdash^{\mathsf{kv}} \kappa_2 & \text{By Lemma D.10}
\end{array}$$

□

Lemma D.6 (Well-formedness of Kinding). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{k}} \sigma : \kappa \dashv \Theta$, *then* $\Delta \longrightarrow \Theta$, *and* Θ ok, *and* $\Theta \Vdash^{\mathsf{kv}} \kappa$ *and* $\Theta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \kappa$.

Proof. By induction on the derivation of kinding.

- Case for rules A-K-NAT, A-K-VAR, A-K-TCON, and A-K-ARROW follows trivially.

- Case

A-K-FORALL
$$\frac{\Delta \Vdash^{\mathsf{kv}} \kappa \qquad \Delta, a : \kappa \Vdash^{\mathsf{k}} \sigma : \kappa_2 \dashv \Theta, a : \kappa \qquad [\Theta]\kappa_2 = \star}{\Delta \Vdash^{\mathsf{k}} \forall a : \kappa.\sigma : \star \dashv \Theta}$$

$$\begin{array}{l|l}
\Delta, a : \kappa \longrightarrow \Theta, a : \kappa \wedge \Theta, a : \kappa \Vdash^{\mathsf{kc}} \sigma \Leftarrow \star & \text{By I.H.} \\
\Delta \longrightarrow \Theta & \text{By inversion} \\
\Theta \Vdash^{\mathsf{kv}} \star & \text{By rule A-KV-STAR} \\
\Delta \Vdash^{\mathsf{kv}} \kappa & \text{Given} \\
\Theta \Vdash^{\mathsf{kv}} \kappa & \text{By lemma D.10} \\
\Theta \Vdash^{\mathsf{kc}} \forall a : \kappa.\sigma \Leftarrow \star & \text{By rules A-KC-EQ and A-K-FORALL}
\end{array}$$

- Case

A-K-APP
$$\frac{\Delta \Vdash^{\mathsf{k}} \tau_1 : \kappa_1 \dashv \Theta_1 \qquad \Theta_1 \Vdash^{\mathsf{k}} \tau_2 : \kappa_2 \dashv \Theta_2 \qquad \Theta_2 \Vdash^{\mathsf{kapp}} [\Theta_2]\kappa_1 \bullet [\Theta_2]\kappa_2 : \kappa_3 \dashv \Theta}{\Delta \Vdash^{\mathsf{k}} \tau_1\, \tau_2 : \kappa_3 \dashv \Theta}$$

$$\begin{array}{l|l}
\Delta \longrightarrow \Theta_1 \wedge \Theta_1 \Vdash^{\mathsf{kv}} \kappa_1 \wedge \Theta_1 \Vdash^{\mathsf{kc}} \tau_1 \Leftarrow \kappa_1 & \text{I.H.} \\
\Theta_1 \longrightarrow \Theta_2 \wedge \Theta_2 \Vdash^{\mathsf{kv}} \kappa_2 \wedge \Theta_2 \Vdash^{\mathsf{kc}} \tau_2 \Leftarrow \kappa_2 & \text{I.H.} \\
\Theta_2 \longrightarrow \Theta & \text{By lemma D.5} \\
\Theta_1 \longrightarrow \Theta & \text{By lemma D.19} \\
\Delta \longrightarrow \Theta & \text{By lemma D.19} \\
\Theta_2 \Vdash^{\mathsf{kv}} \kappa_1 & \text{By lemma D.10} \\
\Theta_2 \Vdash^{\mathsf{kv}} [\Theta_2]\kappa_1 & \text{By lemma D.12} \\
\Theta \Vdash^{\mathsf{kv}} \kappa_3 & \text{By lemma D.5} \\
\Theta \Vdash^{\mathsf{kc}} \tau_1 \Leftarrow \kappa_1 & \text{By lemma D.10} \\
\Theta \Vdash^{\mathsf{kc}} \tau_2 \Leftarrow \kappa_2 & \text{By lemma D.10} \\
[\Theta]([\Theta_2]\kappa_1) = [\Theta]([\Theta_2]\kappa_2) \rightarrow [\Theta]\kappa_3 & \text{By lemma D.40} \\
[\Theta]\kappa_1 = [\Theta]\kappa_2 \rightarrow [\Theta]\kappa_3 & \text{By lemma D.18} \\
\Theta \Vdash^{\mathsf{kapp}} [\Theta]\kappa_1 \bullet [\Theta]\kappa_2 : [\Theta]\kappa_3 \dashv \Theta & \text{By rule A-KAPP-ARROW and rule A-U-REFL} \\
\Theta \Vdash^{\mathsf{kc}} \kappa\, \tau_1\, \tau_2 \Leftarrow \kappa_3 & \text{By rules A-KC-EQ and A-K-APP}
\end{array}$$

□

Lemma D.7 (Well-formedness of Typing Data Constructor Declarations). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{dc}}_{\tau'} \mathcal{D} \rightsquigarrow \tau \dashv \Theta$*, then* $\Delta \longrightarrow \Theta$*, and* Θ ok. *and* $\Theta \Vdash^{\mathsf{kc}} \tau \Leftarrow \star$.

Proof.

A-DC-DECL
$$\frac{\Delta \Vdash^{\mathsf{k}} \overline{\tau_i}^{\,i} \rightarrow \tau : \star \dashv \Theta}{\Delta \Vdash^{\mathsf{dc}}_{\tau} D\, \overline{\tau_i}^{\,i} \rightsquigarrow \overline{\tau_i}^{\,i} \rightarrow \tau \dashv \Theta}$$

Follows directly from lemma D.6.

□

Lemma D.8 (Well-formedness of Typing Datatype Declaration). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow \Gamma \dashv \Theta$, *then* $\Delta \longrightarrow \Theta$*, and* Θ ok, *and* $\Theta \Vdash^{\mathsf{ectx}} \Gamma$.

Proof.

$$\textsc{a-dt-decl}\quad \frac{(T : \kappa) \in \Delta \qquad \Delta, \overline{\widehat{\alpha}_i}^{\,i} \Vdash^{\mathsf{u}} [\Delta]\kappa \approx (\overline{\widehat{\alpha}_i}^{\,i} \rightarrow \star) \dashv \Theta_1, \overline{\widehat{\alpha}_i = \kappa_i}^{\,i} \qquad \overline{\Theta_j, \overline{a_i : \kappa_i}^{\,i} \Vdash^{\mathsf{dc}}_{T\,\overline{a_i}^{\,i}} \mathcal{D}_j \rightsquigarrow \tau_j \dashv \Theta_{j+1}, \overline{a_i : \kappa_i}^{\,i}}^{\,j}}{\Delta \Vdash^{\mathsf{dt}} \mathbf{data}\ T\,\overline{a_i}^{\,i} = \overline{\mathcal{D}_j}^{\,j\in 1..n} \rightsquigarrow \overline{D_j : \forall \overline{a_i : \kappa_i}^{\,i}.\tau_j}^{\,j} \dashv \Theta_{n+1}}$$

$$\begin{array}{l|l}
\Delta \longrightarrow \Delta, \overline{\widehat{\alpha}_i}^{\,i} & \text{By rule } \textsc{a-ctxe-kuvar} \\
\Delta, \overline{\widehat{\alpha}_i}^{\,i} \longrightarrow \Theta_1, \overline{\widehat{\alpha}_i = \kappa_i}^{\,i} & \text{By lemma D.4} \\
\Delta \longrightarrow \Theta_1 & \text{By lemma D.17} \\
\Theta_1, \overline{a_i : \kappa_i}^{\,i} \longrightarrow \Theta_{n+1}, \overline{a_i : \kappa_i}^{\,i} & \text{By lemma D.7 and lemma D.19} \\
\Theta_1 \longrightarrow \Theta_{n+1} & \text{By inversion} \\
\Delta \longrightarrow \Theta & \text{By lemma D.19} \\
\overline{\Theta_{j+1}, \overline{a_i : \kappa_i}^{\,i} \Vdash^{\mathsf{kc}} \tau_j \Leftarrow \star}^{\,j\in 1..n} & \text{By lemma D.7} \\
\overline{\Theta_n, \overline{a_i : \kappa_i}^{\,i} \Vdash^{\mathsf{kc}} \tau_j \Leftarrow \star}^{\,j\in 1..n} & \text{By lemma D.10} \\
\overline{\Theta_n \Vdash^{\mathsf{kc}} \forall \overline{a_i : \kappa_i}^{\,i}.\tau_j \Leftarrow \star}^{\,j\in 1..n} & \text{By rules } \textsc{a-kc-eq} \text{ and } \textsc{a-k-forall} \\
\Theta_n \Vdash^{\mathsf{ectx}} \overline{D_j : \forall \overline{a_i : \widehat{\alpha}_i}^{\,i}.\tau_j}^{\,j\in 1..n} & \text{By rule } \textsc{a-ectx-dcon}
\end{array}$$

□

D.2.3 Properties of Context Extension.

Lemma D.9 (Declaration Preservation). *If $\Delta \longrightarrow \Theta$, if a type constructor or a type variable or a kind unification variable is declared in Δ, then it is declared in Θ.*

Proof. By a straightforward induction on $\Delta \longrightarrow \Theta$. □

Lemma D.10 (Extension Weakening). *Given $\Delta \longrightarrow \Theta$,*

- *if $\Delta \Vdash^{\mathsf{kv}} \kappa$, then $\Theta \Vdash^{\mathsf{kv}} \kappa$;*
- *if $\Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \kappa$, then $\Theta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \kappa$.*

Proof. **Part 1** By induction on $\Delta \Vdash^{\mathsf{kv}} \kappa$.

- Case

$$\textsc{a-kv-star}\quad \frac{}{\Delta \Vdash^{\mathsf{kv}} \star}$$

The goal holds trivially.

- Case

$$\textsc{a-kv-arrow}\quad \frac{\Delta \Vdash^{\mathsf{kv}} \kappa_1 \qquad \Delta \Vdash^{\mathsf{kv}} \kappa_2}{\Delta \Vdash^{\mathsf{kv}} \kappa_1 \rightarrow \kappa_2}$$

The goal holds directly from I.H..

- Case

$$\textsc{a-kv-kuvar}\quad \frac{\widehat{\alpha} \in \Delta}{\Delta \Vdash^{\mathsf{kv}} \widehat{\alpha}}$$

The goal holds directly from lemma D.9.

Part 2 By induction on $\Delta \Vdash^{\mathsf{k}} \sigma : \kappa \dashv \Delta$. The cases for rules a-k-nat, a-k-var, a-k-tcon, and a-k-arrow holds trivially. The case for rule a-k-forall holds from I.H.. The case for rule a-k-app depends on the extension weakening of application kinding. Given the hypothesis, it's impossible for rule A-KAPP-KUVAR. For the extension weakening on rule A-KAPP-ARROW, it then depends on the extension weakening of kind unification. Given the hypothesis, it's impossible for rules A-U-KVARL and A-U-KVARR. The case for rule A-U-RREFL holds trivially, and the case for rule A-U-ARROW holds directly from I.H..

□

Lemma D.12 (Substitution Kinding). *If* Δ ok, *and* $\Delta \Vdash^{kv} \kappa$, *then* $\Delta \Vdash^{kv} [\Delta]\kappa$.

Proof. By induction on $|\ \Delta \vdash \kappa\ |$. We then case analyze κ.

- $\kappa = \star$. The goal holds trivially.
- $\kappa = \kappa_1 \rightarrow \kappa_2$. The goal directly from I.H..
- $\kappa = \widehat{\alpha}$. If $\widehat{\alpha}$ is unsolved in Δ, then the goal holds directly. Or otherwise we have $\Delta = \Delta_1, \widehat{\alpha} = \kappa, \Delta_2$. Because Δ ok, we have $\Delta_1 \Vdash^{kv} \kappa$ and $|\ \Delta_1 \vdash \kappa\ | = |\ \Delta \vdash \kappa\ |$, which is less then $|\ \Delta \vdash \widehat{\alpha}\ |$. So we apply I.H. to get the goal.

□

Lemma D.13 (Context Extension with Defaulting is Context Extension). *If* $\Delta \twoheadrightarrow \Theta$, *then* $\Delta \longrightarrow \Theta$.

Proof. By straightforward induction on $\Delta \twoheadrightarrow \Theta$. □

Lemma D.14 (Reflexivity of Context Extension). *If* Δ ok, *then* $\Delta \longrightarrow \Delta$.

Proof. By straightforward induction on Δ ok. The conclusion follows directly from the definition.

□

Lemma D.15 (Well-formedness of Context Extension). *If* Δ ok, *and* $\Delta \longrightarrow \Theta$, *then* Θ ok.

Proof. By induction on $\Delta \longrightarrow \Theta$.

- Case

A-CTXE-EMPTY

$$\frac{}{\bullet \longrightarrow \bullet}$$

Follows directly by rule A-TCTX-EMPTY.

- Case

A-CTXE-TVAR

$$\frac{\Delta \longrightarrow \Theta}{\Delta, a : \kappa \longrightarrow \Theta, a : \kappa}$$

$\Delta, a : \kappa$ ok	Given
$\Delta \Vdash^{kv} \kappa$	By inversion
$\Delta \longrightarrow \Theta$	Given
$\Theta \Vdash^{kv} \kappa$	By lemma D.10
Θ ok	I.H.
$\Theta, a : \kappa$ ok	By rule A-TCTX-TVAR

- Case

A-CTXE-TCON

$$\frac{\Delta \longrightarrow \Theta}{\Delta, T : \kappa \longrightarrow \Theta, T : \kappa}$$

This case is similar to the case for rule A-TCTX-EMPTY.

- Case

$$\textsc{a-ctxe-kuvar}\ \dfrac{\Delta \longrightarrow \Theta}{\Delta, \widehat{\alpha} \longrightarrow \Theta, \widehat{\alpha}}$$

  The goal holds directly from I.H. and rule A-TCTX-KUVAR.

- Case

$$\textsc{a-ctxe-kuvarSolved}\ \dfrac{\Delta \longrightarrow \Theta \qquad [\Theta]\kappa_1 = [\Theta]\kappa_2}{\Delta, \widehat{\alpha} = \kappa_1 \longrightarrow \Theta, \widehat{\alpha} = \kappa_2}$$

$$\begin{array}{l|l}
\Theta \text{ ok} & \text{I.H.} \\
\Delta, \widehat{\alpha} = \kappa_1 \text{ ok Given} & \\
\Delta \Vdash^{\text{kv}} \kappa_1 & \text{By inversion} \\
\Delta \longrightarrow \Theta & \text{Given} \\
\Theta \Vdash^{\text{kv}} \kappa_1 & \text{By lemma D.10}
\end{array}$$

  Suppose κ_2 is not well-formed under Θ, then it must contain kind unification variables that are not in Θ. Then it is impossible to have $[\Theta]\kappa_1 = [\Theta]\kappa_2$ given $\Theta \Vdash^{\text{kv}} \kappa_1$. Thus by contradiction we have $\Theta \Vdash^{\text{kv}} \kappa_2$. Then $\Theta, X_1 = \kappa_2$ ok by rule A-TCTX-TCON.

- Case

$$\textsc{a-ctxe-solve}\ \dfrac{\Delta \longrightarrow \Theta \qquad \Theta \Vdash^{\text{kv}} \kappa}{\Delta, \widehat{\alpha} \longrightarrow \Theta, \widehat{\alpha} = \kappa}$$

  The goal holds directly from I.H. and rule A-TCTX-KUVARSOLVED.

- Case

$$\textsc{a-ctxe-add}\ \dfrac{\Delta \longrightarrow \Theta}{\Delta \longrightarrow \Theta, \widehat{\alpha}}$$

  The goal holds directly from I.H. and rule A-TCTX-KUVAR.

- Case

$$\textsc{a-ctxe-addSolved}\ \dfrac{\Delta \longrightarrow \Theta \qquad \Theta \Vdash^{\text{kv}} \kappa}{\Delta \longrightarrow \Theta, \widehat{\alpha} = \kappa}$$

  The goal holds directly from I.H. and rule A-TCTX-KUVARSOLVED.

□

Lemma D.17 (Extension Order).

(1) If $\Delta_1, a : \kappa, \Delta_2 \longrightarrow \Theta$, then $\Theta = \Theta_1, a : \kappa, \Theta_2$, where $\Delta_1 \longrightarrow \Theta_1$. Moreover, if Δ_2 **soft**, *then Θ_2* **soft**.

(2) If $\Delta_1, T : \kappa, \Delta_2 \longrightarrow \Theta$, then $\Theta = \Theta_1, T : \kappa, \Theta_2$, where $\Delta_1 \longrightarrow \Theta_1$. Moreover, if Δ_2 **soft**, *then Θ_2* **soft**.

(3) If $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta$, then $\Theta = \Theta_1, \Theta', \Theta_2$, where $\Delta_1 \longrightarrow \Theta_1$, and Θ' is either $\widehat{\alpha}$ or $\widehat{\alpha} = \kappa$ for some κ. Moreover, if Δ_2 **soft**, *then Θ_2* **soft**.

(4) If $\Delta_1, \widehat{\alpha} = \kappa_1, \Delta_2 \longrightarrow \Theta$, then $\Theta = \Theta_1, \widehat{\alpha} = \kappa_2, \Theta_2$, where $\Delta_1 \longrightarrow \Theta_1$, and $[\Theta_1]\kappa_1 = [\Theta_1]\kappa_2$. Moreover, if Δ_2 **soft**, *then Θ_2* **soft**.

Proof. We only detail the proof for the first case, and all the rest cases are similar to the first one.

By induction on $\Delta_1, a : \kappa, \Delta_2 \longrightarrow \Theta$.

- Case $\Delta = \bullet$ by rule A-CTXE-EMPTY. This case is impossible.
- Case $\Delta_1, a : \kappa \longrightarrow \Theta', a : \kappa$ by rule A-CTXE-TVAR when Δ_2 is empty. In this case, let $\Theta_1 = \Theta'$ and Θ_2 be empty. All goals follow directly.
- Case $\Delta_1, a : \kappa, \Delta', b : \kappa_2 \longrightarrow \Theta', b : \kappa_2$ by rule A-CTXE-TVAR where $\Delta_2 = \Delta', b : \kappa_2$ and $\Delta_1, a : \kappa, \Delta' \longrightarrow \Theta'$. By I.H. we have $\Theta' = \Theta_1, a : \kappa, \Theta_2'$ and $\Delta_1 \longrightarrow \Theta_1$. Let $\Theta_2 = \Theta_2', b : \kappa_2$ and all goals follow directly.
- Case $\Delta_1, a : \kappa, \Delta', T : \kappa_2 \longrightarrow \Theta', T : \kappa_2$ by rule A-CTXE-TCON where $\Delta_2 = \Delta', T : \kappa_2$ and $\Delta_1, a : \kappa, \Delta' \longrightarrow \Theta'$. This case is similar to the above case.
- Case $\Delta_1, a : \kappa, \Delta', \widehat{\alpha} \longrightarrow \Theta', \widehat{\alpha}$ by rule A-CTXE-KUVAR where $\Delta_2 = \Delta', \widehat{\alpha}$ and $\Delta_1, a : \kappa, \Delta' \longrightarrow \Theta'$. By I.H. we have $\Theta' = \Theta_1, a : \kappa, \Theta_2'$ and $\Delta_1 \longrightarrow \Theta_1$. Let $\Theta_2 = \Theta_2', \widehat{\alpha}$ and all goals follow directly. And if Δ' **soft**, by I.H. we have Θ_2' **soft**. By definition we have Θ_2 **soft**.
- Case for rules A-CTXE-KUVARSOLVED, A-CTXE-SOLVE, A-CTXE-ADD, and A-CTXE-ADDSOLVED are similar to the above case.

□

Lemma D.18 (Substitution Extension Invariance). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{kv}} \kappa$, *and* $\Delta \longrightarrow \Theta$, *then* $[\Theta]\kappa = [\Theta]([\Delta]\kappa)$ *and* $[\Theta]\kappa = [\Delta]([\Theta]\kappa)$.

Proof. Because $\Delta \Vdash^{\mathsf{kv}} \kappa$, so every solved kind unification variable in Δ is solved in Θ. Therefore $[\Theta]\kappa = [\Delta]([\Theta]\kappa)$.

To show that $[\Theta]\kappa = [\Theta]([\Delta]\kappa)$, we do induction on $\mid \Delta \vdash \kappa \mid$.

- A-KV-STAR
$$\frac{}{\Delta \Vdash^{\mathsf{kv}} \star}$$
The goal follows trivially.
- A-KV-ARROW
$$\frac{\Delta \Vdash^{\mathsf{kv}} \kappa_1 \qquad \Delta \Vdash^{\mathsf{kv}} \kappa_2}{\Delta \Vdash^{\mathsf{kv}} \kappa_1 \rightarrow \kappa_2}$$
The goal follows directly from I.H..
- A-KV-KUVAR
$$\frac{\widehat{\alpha} \in \Delta}{\Delta \Vdash^{\mathsf{kv}} \widehat{\alpha}}$$
There are two subcases. Firstly, $\widehat{\alpha}$ is unsolved in Δ. Then $[\Theta]([\Delta]\widehat{\alpha}) = [\Theta]\widehat{\alpha}$ follows directly. Or we have $\Delta = \Delta_1, \widehat{\alpha} = \kappa, \Delta_2$. Then by lemma D.17 we have $\Theta = \Theta_1, \widehat{\alpha} = \kappa', \Theta_2$ and $[\Theta_1]\kappa = [\Theta_1]\kappa'$. Because $\mid \Delta \vdash \kappa \mid < \mid \Delta \vdash \widehat{\alpha} \mid$, by I.H., we know that $[\Theta]\kappa = [\Theta]([\Delta]\kappa)$. Therefore, $[\Theta]\widehat{\alpha} = [\Theta]\kappa' = [\Theta]\kappa = [\Theta]([\Delta]\kappa) = [\Theta]([\Delta]\widehat{\alpha})$.

□

Lemma D.19 (Transitivity of Context Extension). *If* Δ' ok, *and* $\Delta' \longrightarrow \Delta$, *and* $\Delta \longrightarrow \Theta$, *then* $\Delta' \longrightarrow \Theta$.

Proof. By induction on $\Delta \longrightarrow \Theta$.

- Case
A-CTXE-EMPTY
$$\frac{}{\bullet \longrightarrow \bullet}$$
We have $\Delta' \longrightarrow \bullet$ as given.

- Case

$$\frac{\Delta \longrightarrow \Theta}{\Delta, a : \kappa \longrightarrow \Theta, a : \kappa}\ \text{A-CTXE-TVAR}$$

$\Delta' \longrightarrow \Delta, a : \kappa$	Given
$\Delta' = \Delta_1, a : \kappa \wedge \Delta_1 \longrightarrow \Delta$	By inversion
$\Delta_1 \longrightarrow \Theta$	I.H.
$\Delta_1, a : \kappa \longrightarrow \Theta, a : \kappa$	By rule A-CTXE-TVAR

- Case

$$\frac{\Delta \longrightarrow \Theta}{\Delta, T : \kappa \longrightarrow \Theta, T : \kappa}\ \text{A-CTXE-TCON}$$

  This case is similar to the case for rule A-CTXE-TVAR.

- Case

$$\frac{\Delta \longrightarrow \Theta}{\Delta, \widehat{\alpha} \longrightarrow \Theta, \widehat{\alpha}}\ \text{A-CTXE-KUVAR}$$

  Since $\Delta' \longrightarrow \Delta, \widehat{\alpha}$, the derivation must conclude with either rule A-CTXE-KUVAR or rule A-CTXE-ADD.

  – By rule A-CTXE-KUVAR.

$\Delta' = \Delta_1, \widehat{\alpha} \wedge \Delta_1 \longrightarrow \Delta$	Given
$\Delta_1 \longrightarrow \Theta$	I.H.
$\Delta_1, \widehat{\alpha} \longrightarrow \Theta, \widehat{\alpha}$	By rule A-CTXE-KUVAR

  – By rule A-CTXE-ADD.

$\Delta' \longrightarrow \Delta$	Given
$\Delta' \longrightarrow \Theta$	I.H.
$\Delta' \longrightarrow \Theta, \widehat{\alpha}$	By rule A-CTXE-ADD

- Case

$$\frac{\Delta \longrightarrow \Theta \qquad [\Theta]\kappa_1 = [\Theta]\kappa_2}{\Delta, \widehat{\alpha} = \kappa_1 \longrightarrow \Theta, \widehat{\alpha} = \kappa_2}\ \text{A-CTXE-KUVARSOLVED}$$

  Since $\Delta' \longrightarrow \Delta, \widehat{\alpha} = \kappa_1$, the derivation must conclude with either rule A-CTXE-KUVARSOLVED or rule A-CTXE-ADDSOLVED.

  – By rule A-CTXE-KUVARSOLVED.

$\Delta' = \Delta_1, \widehat{\alpha} = \kappa_0 \wedge \Delta_1 \longrightarrow \Delta \wedge [\Delta]\kappa_0 = [\Delta]\kappa_1$	Given
$\Delta_1 \longrightarrow \Theta$	I.H.
$[\Theta]\kappa_0$	
$= [\Theta]([\Delta]\kappa_0)$	By lemma D.18
$= [\Theta]([\Delta]\kappa_1)$	Given
$= [\Theta]\kappa_1$	By lemma D.18
$= [\Theta]\kappa_2$	Given
$\Delta_1, \widehat{\alpha} = \kappa_0 \longrightarrow \Theta, \widehat{\alpha} = \kappa_2$	By rule A-CTXE-KUVAR

- By rule A-CTXE-ADDSOLVED.

$$\begin{array}{l|l} \Delta' \longrightarrow \Delta & \text{Given} \\ \Delta' \longrightarrow \Theta & \text{I.H.} \\ \Delta' \longrightarrow \Theta, \widehat{\alpha} = \kappa_2 & \text{By rule A-CTXE-ADDSOLVED} \end{array}$$

- Case

$$\begin{array}{c} \text{A-CTXE-SOLVE} \\ \dfrac{\Delta \longrightarrow \Theta \qquad \Theta \Vdash^{\mathsf{kv}} \kappa}{\Delta, \widehat{\alpha} \longrightarrow \Theta, \widehat{\alpha} = \kappa} \end{array}$$

Since $\Delta' \longrightarrow \Delta, \widehat{\alpha}$, the derivation must conclude with either rule A-CTXE-KUVAR or rule A-CTXE-ADD.

- By rule A-CTXE-KUVAR.

$$\begin{array}{l|l} \Delta' = \Delta_1, \widehat{\alpha} \wedge \Delta_1 \longrightarrow \Delta & \text{Given} \\ \Delta_1 \longrightarrow \Theta & \text{I.H.} \\ \Delta_1, \widehat{\alpha} \longrightarrow \Theta, \widehat{\alpha} = \kappa & \text{By rule A-CTXE-SOLVE} \end{array}$$

- By rule A-CTXE-ADD.

$$\begin{array}{l|l} \Delta' \longrightarrow \Delta & \text{Given} \\ \Delta' \longrightarrow \Theta & \text{I.H.} \\ \Delta' \longrightarrow \Theta, \widehat{\alpha} = \kappa & \text{By rule A-CTXE-ADDSOLVED} \end{array}$$

- Case

$$\begin{array}{c} \text{A-CTXE-ADD} \\ \dfrac{\Delta \longrightarrow \Theta}{\Delta \longrightarrow \Theta, \widehat{\alpha}} \end{array}$$

$$\begin{array}{l|l} \Delta' \longrightarrow \Theta & \text{I.H.} \\ \Delta' \longrightarrow \Theta, \widehat{\alpha} & \text{By rule A-CTXE-ADD} \end{array}$$

- Case

$$\begin{array}{c} \text{A-CTXE-ADDSOLVED} \\ \dfrac{\Delta \longrightarrow \Theta \qquad \Theta \Vdash^{\mathsf{kv}} \kappa}{\Delta \longrightarrow \Theta, \widehat{\alpha} = \kappa} \end{array}$$

$$\begin{array}{l|l} \Delta' \longrightarrow \Theta & \text{I.H.} \\ \Delta' \longrightarrow \Theta, \widehat{\alpha} = \kappa & \text{By rule A-CTXE-ADDSOLVED} \end{array}$$

□

Lemma D.20 (Solution Admissibility for Extension). *If* $\Delta_1, \widehat{\alpha}, \Delta_2$ ok *and* $\Delta_1 \Vdash^{\mathsf{kv}} \kappa$*, then* $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha} = \kappa, \Delta_2$.

Proof. By induction on Δ_2.

- Case Δ_2 is empty. Then $\Delta_1 \longrightarrow \Delta_1$ by lemma D.14, and $\Delta_1, \widehat{\alpha} \longrightarrow \Delta_1, \widehat{\alpha} = \kappa$ holds by rule A-CTXE-SOLVE.
- Case $\Delta_2 = \Delta_2', a : \kappa$. By I.H., we $\Delta_1, \widehat{\alpha}, \Delta_2' \longrightarrow \Delta_1, \widehat{\alpha} = \kappa, \Delta_2'$. Then by rule A-CTXE-TVAR we are done.
- Case $\Delta_2 = \Delta_2', T : \kappa$. By I.H. and rule A-TCTXE-TCON.

- Case $\Delta_2 = \Delta_2', \widehat{\alpha}$. By I.H. and rule A-CTXE-KUVAR.
- Case $\Delta_2 = \Delta_2', \widehat{\alpha} = \kappa$. By I.H. and rule A-CTXE-KUVARSOLVED.

□

Lemma D.21 (Solved Variable Addition for Extension). *If* Δ_1, Δ_2 ok *and* $\Delta_1 \Vdash^{\mathsf{kv}} \kappa$*, then* $\Delta_1, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha} = \kappa, \Delta_2$.

Proof. The proof is exactly the same as the one for lemma D.20. Except for the case when Δ_2 is empty, we use rule A-CTXE-ADDSOLVED.

□

Lemma D.22 (Unsolved Variable Addition). *If* Δ_1, Δ_2 ok *then* $\Delta_1, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha}, \Delta_2$.

Proof. The proof is exactly the same as the one for lemma D.20. Except for the case when Δ_2 is empty, we use rule A-CTXE-ADD.

□

Lemma D.23 (Parallel Admissibility). *If* $\Delta_1 \longrightarrow \Theta_1$*, and* Δ_1, Δ_2 ok*, and* $\Delta_1, \Delta_2 \longrightarrow \Theta_1, \Theta_2$*, and* Δ_2 *is fresh w.r.t.* Θ_1*, then:*

- $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha}, \Theta_2$
- *If* $\Theta_1 \Vdash^{\mathsf{kv}} \kappa$*, then* $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} = \kappa, \Theta_2$
- *If* $[\Theta_1]\kappa_1 = [\Theta_1]\kappa_2$*, then* $\Delta_1, \widehat{\alpha} = \kappa_1, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} = \kappa_2, \Theta_2$

Proof. **Part 1** By induction on Θ_2.

- $\Theta_2 = \bullet$. Because Δ_2 is fresh w.r.t. Θ_1, we must have $\Delta_2 = \bullet$. We have $\Delta_1, \widehat{\alpha} \longrightarrow \Theta_1, \widehat{\alpha}$ by rule A-CTXE-KUVAR.
- $\Theta_2 = \Theta_2', a : \kappa$. Then the derivation of $\Delta_1, \Delta_2 \longrightarrow \Theta_1, \Theta_2$ must conclude with rule A-CTXE-TVAR. It must be $\Delta_2 = \Delta_2', a : \kappa$. (Or otherwise if $(a : \kappa) \in \Delta_1$, then we must have $(a : \kappa) \in \Theta_1$ by lemma D.9, and Θ_1, Θ_2 is no longer well-formed.)

$\Delta_1, \Delta_2', a : \kappa \longrightarrow \Theta_1, \Theta_2', a : \kappa$	Given
$\Delta_1, \Delta_2' \longrightarrow \Theta_1, \Theta_2'$	By inversion
$\Delta_1, \widehat{\alpha}, \Delta_2' \longrightarrow \Theta_1, \widehat{\alpha}, \Theta_2'$	I.H.
$\Delta_1, \widehat{\alpha}, \Delta_2', a : \kappa \longrightarrow \Theta_1, \widehat{\alpha}, \Theta_2', a : \kappa$	By rule A-CTXE-TVAR

- $\Theta_2 = \Theta_2', T : \kappa$ This case is similar to the case when $\Theta_2 = \Theta_2', a : \kappa$, except that we reason using rule A-CTXE-TCON.
- $\Theta_2 = \Theta_2', \widehat{\alpha}_1$ Then the derivation of $\Delta_1, \Delta_2 \longrightarrow \Theta_1, \Theta_2$ must conclude with either rule A-CTXE-KUVAR or rule A-CTXE-ADD.
  - Subcase: the derivation concludes with rule A-CTXE-KUVAR. It must be $\Delta_2 = \Delta_2', \widehat{\alpha}_1$.

$\Delta_1, \Delta_2' \longrightarrow \Theta_1, \Theta_2'$	Given
$\Delta_1, \widehat{\alpha}, \Delta_2' \longrightarrow \Theta_1, \widehat{\alpha}, \Theta_2'$	I.H.
$\Delta_1, \widehat{\alpha}, \Delta_2', \widehat{\alpha} \longrightarrow \Theta_1, \widehat{\alpha}, \Theta_2', \widehat{\alpha}_1$	By rule A-CTXE-KUVAR

  - Subcase: the derivation concludes with rule A-CTXE-ADD.

$\Delta_1, \Delta_2 \longrightarrow \Theta_1, \Theta_2'$	Given
$\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta_1, \Theta_2'$	I.H.
$\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha}, \Theta_2', \widehat{\alpha}_1$	By rule A-CTXE-ADD

- $\Theta_2 = \Theta_2', \widehat{\alpha}_1 = \kappa$. Then the derivation of $\Delta_1, \Delta_2 \longrightarrow \Theta_1, \Theta_2$ must conclude with either rule A-CTXE-KUVARSOLVED or rule A-CTXE-ADDSOLVED or rule A-CTXE-SOLVE. In either case, the reasoning is similar to the case when $\Theta_2 = \Theta_2', \widehat{\alpha}_1$.

Part 2 Similar to Part 1, except that when $\Theta_2 = \bullet$, we apply rule A-CTXE-SOLVE.

Part 3 Similar to Part 1, except that when $\Theta_2 = \bullet$, we apply rule A-CTXE-KUVARSOLVED.

□

Lemma D.24 (Parallel Extension Solution). *If* $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} = \kappa_2, \Theta_2$, *and* $[\Theta_1]\kappa_1 = [\Theta_1]\kappa_2$, *then* $\Delta_1, \widehat{\alpha} = \kappa_1, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} = \kappa_2, \Theta_2$.

Proof. By induction on Θ_2.

- Case Θ_2 is empty. Then Δ_2 must be empty. Then $\Delta_1, \widehat{\alpha} \longrightarrow \Theta_1, \widehat{\alpha} = \kappa_2$. By inversion we have $\Delta_1 \longrightarrow \Theta_1$. And $\Delta_1, \widehat{\alpha} = \kappa_1 \longrightarrow \Theta_1, \widehat{\alpha} = \kappa_2$ holds by rule A-CTXE-KUVARSOLVED.
- Case $\Theta_2 = \Theta_2', a : \kappa$. Then $\Delta_2 = \Delta_2', a : \kappa$. By I.H., we $\Delta_1, \widehat{\alpha} = \kappa_1, \Delta_2' \longrightarrow \Theta_1, \widehat{\alpha} = \kappa_2, \Theta_2'$. Then by rule A-TCTXE-TVAR we are done.
- Case $\Theta_2 = \Theta_2, T : \kappa$. By I.H. and rule A-TCTXE-TCON.
- Case $\Theta_2 = \Theta_2, \widehat{\alpha}_2$. By I.H. and rule A-CTXE-KUVAR.
- Case $\Theta_2 = \Theta_2, \widehat{\alpha}_2 = \kappa$. By I.H. and rule A-CTXE-KUVARSOLVED.

□

Lemma D.25 (Parallel Variable Update). *If* $\Delta_1, \widehat{\alpha}, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} = \kappa, \Theta_2$, *and* $\Delta_1 \Vdash^{kv} \kappa_1$, *and* $\Theta_1 \Vdash^{kv} \kappa_2$, *and* $[\Theta_1]\kappa = [\Theta_1]\kappa_1 = [\Theta_1]\kappa_2$, *then* $\Delta_1, \widehat{\alpha} = \kappa_1, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} = \kappa_2, \Theta_2$

Proof. The proof is exactly the same as the one for lemma D.24. Except for the case when Θ_2 is empty, we use rule A-CTXE-SOLVE.

□

D.2.4 Properties of Complete Context.

Lemma D.26 (Finishing Kinding). *If* Ω ok, *and* $\Omega \Vdash^{kv} \kappa$, *and* $\Omega \longrightarrow \Omega'$, *then* $[\Omega]\kappa = [\Omega']\kappa$.

Proof. By lemma D.18 we know $[\Omega']\kappa = [\Omega']([\Omega]\kappa)$. Because $[\Omega]\kappa$ contains no unsolved kind unification variable, we have $[\Omega']([\Omega]\kappa) = [\Omega]\kappa$. Therefore $[\Omega']\kappa = [\Omega]\kappa$.

□

Lemma D.27 (Finishing Term Contexts). *If* Ω ok, *and* $\Omega \Vdash^{ectx} \Gamma$, *and* $\Omega \longrightarrow \Omega'$, *then* $[\Omega']\Gamma = [\Omega]\Gamma$.

Proof. Essentially, $[\Omega]\Gamma$ only works for all kinds that appear in types in Γ. By $\Omega \Vdash^{ectx} \Gamma$, we have that any kind κ that appears in Γ has $\Omega \Vdash^{kv} \kappa$. So our goal follows directly from lemma D.26.

□

Lemma D.28 (Stability of Complete Contexts). *If* $\Delta \longrightarrow \Omega$, *then* $[\Omega]\Delta = [\Omega]\Omega$.

Proof. By induction on $\Delta \longrightarrow \Omega$.

- Case

  A-CTXE-EMPTY
  $$\frac{}{\bullet \longrightarrow \bullet}$$

  The goal follows trivially.
- Case

  A-CTXE-TVAR
  $$\frac{\Delta \longrightarrow \Theta}{\Delta, a : \kappa \longrightarrow \Theta, a : \kappa}$$

  We have $\Delta = \Delta', a : \kappa$, $\Omega = \Omega', a : \kappa$ and $\Delta' \longrightarrow \Omega'$.

$$\begin{array}{l|l}
[\Omega]\Delta & \\
= [\Omega', a : \kappa](\Delta', a : \kappa) & \text{By definition} \\
= [\Omega']\Delta', a : [\Omega']\kappa & \text{By definition} \\
= [\Omega']\Omega', a : [\Omega']\kappa & \text{By I.H.} \\
= [\Omega', a : \kappa](\Omega', a : \kappa) & \text{By definition}
\end{array}$$

- Case
$$\begin{array}{c}\textsc{a-ctxe-tcon}\\ \dfrac{\Delta \longrightarrow \Theta}{\Delta, T : \kappa \longrightarrow \Theta, T : \kappa}\end{array}$$
This case is similar to the case for rule A-CTXE-TVAR.
- Case
$$\begin{array}{c}\textsc{a-ctxe-kuvar}\\ \dfrac{\Delta \longrightarrow \Theta}{\Delta, \widehat{\alpha} \longrightarrow \Theta, \widehat{\alpha}}\end{array}$$
This case is impossible as Ω is a complete context.
- Case
$$\begin{array}{c}\textsc{a-ctxe-kuvarSolved}\\ \dfrac{\Delta \longrightarrow \Theta \qquad [\Theta]\kappa_1 = [\Theta]\kappa_2}{\Delta, \widehat{\alpha} = \kappa_1 \longrightarrow \Theta, \widehat{\alpha} = \kappa_2}\end{array}$$
We have $\Delta = \Delta', \widehat{\alpha} = \kappa_1$, $\Omega = \Omega', \widehat{\alpha} = \kappa_2$ and $[\Omega']\kappa_1 = [\Omega']\kappa_2$.

$$\begin{array}{l|l}
[\Omega]\Delta & \\
= [\Omega', \widehat{\alpha} = \kappa_2](\Delta', \widehat{\alpha} = \kappa_1) & \text{By definition} \\
= [\Omega']\Delta' & \text{By definition} \\
= [\Omega']\Omega' & \text{By I.H.} \\
= [\Omega', \widehat{\alpha} = \kappa_2](\Omega', \widehat{\alpha} = \kappa_2) & \text{By definition}
\end{array}$$

- Case
$$\begin{array}{c}\textsc{a-ctxe-solve}\\ \dfrac{\Delta \longrightarrow \Theta \qquad \Theta \Vdash^{\mathsf{kv}} \kappa}{\Delta, \widehat{\alpha} \longrightarrow \Theta, \widehat{\alpha} = \kappa}\end{array}$$
This case is similar to the case for rule A-CTXE-UVARSOLVED.
- Case
$$\begin{array}{c}\textsc{a-ctxe-add}\\ \dfrac{\Delta \longrightarrow \Theta}{\Delta \longrightarrow \Theta, \widehat{\alpha}}\end{array}$$
This case is impossible as Ω is a complete context.
- Case
$$\begin{array}{c}\textsc{a-ctxe-addSolved}\\ \dfrac{\Delta \longrightarrow \Theta \qquad \Theta \Vdash^{\mathsf{kv}} \kappa}{\Delta \longrightarrow \Theta, \widehat{\alpha} = \kappa}\end{array}$$
This case is similar to the case for rule A-CTXE-UVARSOLVED.

□

Lemma D.29 (Softness Goes Away). *If $\Delta_1, \Delta_2 \longrightarrow \Omega_1, \Omega_2$ where $\Delta_1 \longrightarrow \Omega_1$, and Δ_2* **soft**, *then $[\Omega_1, \Omega_2](\Delta_1, \Delta_2) = [\Omega_1]\Delta_1$.*

Proof. By induction on Δ_2 and the goal follows directly from the definition of context application.

□

Lemma D.30 (Confluence of Completeness). *If* $\Delta_1 \longrightarrow \Omega$, *and* $\Delta_2 \longrightarrow \Omega$, *then* $[\Omega]\Delta_1 = [\Omega]\Delta_2$.

Proof. By lemma D.28 we have $[\Omega]\Delta_1 = [\Omega]\Omega$ and $[\Omega]\Delta_2 = [\Omega]\Omega$. Therefore $[\Omega]\Delta_1 = [\Omega]\Delta_2$. □

Lemma D.31 (Substitution Stability). *If* Ω, Δ ok, *and* $\Omega \Vdash^{\mathsf{kv}} \kappa$, *then* $[\Omega]\kappa = [\Omega, \Delta]\kappa$.

Proof. By straightforward induction on $\mid \Omega \vdash \kappa \mid$.

□

Lemma D.32 (Type Constructor Preservation). *If* Δ ok, *then* $(T : \kappa) \in \Delta$, *and* $\Delta \longrightarrow \Omega$, *then* $(T : [\Omega]\kappa) \in [\Omega]\Delta$.

Proof. Suppose $\Delta = \Delta_1, T : \kappa, \Delta_2$. Then by lemma D.17 we know $\Omega = \Omega_1, T : \kappa, \Omega_2$, $\Delta_1 \longrightarrow \Omega_1$. So $(T : [\Omega_1]\kappa) \in [\Omega]\Delta$ according to the definition of context application. Because Δ ok, and $\Delta \longrightarrow \Omega$, by lemma D.15 we have Ω ok. So by inversion we have $\Omega_1 \Vdash^{\mathsf{kv}} \kappa$. By lemma D.31 we have $[\Omega]\kappa = [\Omega_1]\kappa$. Therefore $(T : [\Omega]\kappa) \in [\Omega]\Delta$.

□

Lemma D.33 (Type Variable Preservation). *If* $(a : \kappa) \in \Delta$, *and* $\Delta \longrightarrow \Omega$, *then* $(a : [\Omega]\kappa) \in [\Omega]\Delta$.

Proof. This lemma is similar to lemma D.32.

□

Lemma D.34 (Finishing Completions). *If* Ω ok, *and* $\Omega \longrightarrow \Omega'$, *then* $[\Omega]\Omega = [\Omega']\Omega'$.

Proof. By induction on $\Omega \longrightarrow \Omega'$.

- Case

$$\dfrac{}{\bullet \longrightarrow \bullet}\ \text{A-CTXE-EMPTY}$$

  The goal follows trivially.
- Case

$$\dfrac{\Delta \longrightarrow \Theta}{\Delta, a : \kappa \longrightarrow \Theta, a : \kappa}\ \text{A-CTXE-TVAR}$$

  So we have $\Omega = \Omega_1, a : \kappa$, and $\Omega' = \Omega_1', a : \kappa$.

$$\begin{array}{l|l}
[\Omega_1]\Omega_1 = [\Omega_1']\Omega_1' & \text{By I.H.} \\
[\Omega_1, a : \kappa](\Omega_1, a : \kappa) = [\Omega_1]\Omega_1, a : [\Omega_1]\kappa & \text{By definition} \\
[\Omega_1', a : \kappa](\Omega_1', a : \kappa) = [\Omega_1']\Omega_1', a : [\Omega_1']\kappa & \text{By definition} \\
\Omega \text{ ok} & \text{Given} \\
\Omega_1 \text{ ok} \land \Omega_1 \Vdash^{\mathsf{kv}} \kappa & \text{By inversion} \\
[\Omega_1]\kappa = [\Omega_1']\kappa & \text{By lemma D.26} \\
[\Omega_1, a : \kappa](\Omega_1, a : \kappa) = [\Omega_1', a : \kappa](\Omega_1', a : \kappa) & \text{Follows from the equations}
\end{array}$$

- Cases for rules A-CTXE-TCON, A-CTXE-KUVARSOLVED, A-CTXE-SOLVE, and A-CTXE-ADDSOLVED are similar to the case for rule A-CTXE-TVAR.
- Cases for rules A-CTXE-KUVAR and A-CTXE-ADD are impossible as Ω and Ω' are complete contexts.

□

D.2.5 Soundness of Algorithm.

Lemma D.35 (Soundness of Kind Validating). *If* Ω ok, *and* $\Omega \Vdash^{kv} \kappa$, *then* $[\Omega]\kappa$ *is a validate kind in the declarative system.*

Proof. By induction on the size of $|\ \Omega \vdash \kappa\ |$. Then case analyze on κ.

- Case $\kappa = \star$. Follows trivially by $[\Omega]\star = \star$.
- Case $\kappa = \kappa_1 \to \kappa_2$. Follows directly from I.H..
- Case $\kappa = \widehat{\alpha}$. Ω must be $\Omega_1, \widehat{\alpha} = \kappa, \Omega_2$, and $[\Omega]\widehat{\alpha} = [\Omega]\kappa$. By I.H., we know $[\Omega]\kappa$ is a well-formed kind.

□

Lemma D.36 (Soundness of Well-formed Type Context). *If* Δ ok, *and* $\Delta \longrightarrow \Omega$, *then* $[\Omega]\Delta$ *is a valid type context in the declarative system.*

Proof. By induction on the well-formedness of type context.

- Case

$$\overset{\text{A-TCTX-EMPTY}}{\dfrac{}{\bullet\ \mathsf{ok}}}$$

Holds trivially.

- Case

$$\overset{\text{A-TCTX-TVAR}}{\dfrac{\Delta\ \mathsf{ok} \qquad \Delta \Vdash^{kv} \kappa}{\Delta, a : \kappa\ \mathsf{ok}}}$$

$$\begin{array}{l|l}
\Delta, a : \kappa \longrightarrow \Omega & \text{Given} \\
\Omega = \Omega_1, a : \kappa, \Omega_2 \land \Omega_2\ \textbf{soft} \land \Delta \longrightarrow \Omega_1 & \text{By lemma D.17} \\
[\Omega](\Delta, a : \kappa) & \\
= [\Omega_1, a : \kappa, \Omega_2](\Delta, a : \kappa) & \\
= [\Omega_1, a : \kappa](\Delta, a : \kappa) & \text{By lemma D.29} \\
= [\Omega_1]\Delta, a : [\Omega_1]\kappa & \text{By definition} \\
[\Omega_1]\Delta \text{ is a valid type context} & \text{I.H.} \\
[\Omega_1]\kappa \text{ is a validate kind} & \text{By lemma D.35} \\
[\Omega_1]\Delta, a : [\Omega_1]\kappa \text{ is a valid type context} &
\end{array}$$

- Case

$$\overset{\text{A-TCTX-TCON}}{\dfrac{\Delta\ \mathsf{ok} \qquad \Delta \Vdash^{kv} \kappa}{\Delta, T : \kappa\ \mathsf{ok}}}$$

This case is similar to the case rule A-TCTX-TVAR.

- Case

$$\overset{\text{A-TCTX-KUVAR}}{\dfrac{\Delta\ \mathsf{ok}}{\Delta, \widehat{\alpha}\ \mathsf{ok}}}$$

$$\begin{array}{l|l}
\Delta, \widehat{\alpha} \longrightarrow \Omega & \text{Given} \\
\Omega = \Omega_1, \widehat{\alpha} = \kappa, \Omega_2 \land \Omega_2\ \textbf{soft} \land \Delta \longrightarrow \Omega_1 & \text{By lemma D.17} \\
[\Omega](\Delta, \widehat{\alpha}) & \\
= [\Omega_1, \widehat{\alpha} = \kappa, \Omega_2](\Delta, \widehat{\alpha}) &
\end{array}$$

$= [\Omega_1]\Delta$	By lemma D.29
$[\Omega_1]\Delta$ is a valid type context	I.H.

- Case

  A-TCTX-KUVARSolved
  $$\frac{\Delta\ \mathsf{ok} \qquad \Delta \Vdash^{\mathsf{kv}} \kappa}{\Delta, \widehat{\alpha} = \kappa\ \mathsf{ok}}$$

  This case is similar to the case rule A-TCTX-KUVAR.

□

Lemma D.37 (Soundness of Well-formed Term Context). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{ectx}} \Gamma$, *and* $\Delta \longrightarrow \Omega$, *then* $[\Omega]\Delta \Vdash^{\mathsf{ectx}} [\Omega]\Gamma$.

Proof. By induction on the judgment of well-formed term context.

- Case

  A-ECTX-EMPTY
  $$\frac{}{\Delta \Vdash^{\mathsf{ectx}} \bullet}$$

  Follows trivially by rule ECTX-EMPTY.

- Case

  A-ECTX-VAR
  $$\frac{\Delta \Vdash^{\mathsf{ectx}} \Gamma \qquad \Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \star}{\Delta \Vdash^{\mathsf{ectx}} \Gamma, x : \sigma}$$

$[\Omega]\Delta \Vdash^{\mathsf{ectx}} [\Omega]\Gamma$	I.H.
$[\Omega]\Delta \vdash^{\mathsf{k}} [\Omega]\sigma : [\Omega]\star$	By lemma D.41
$[\Omega]\Delta \vdash^{\mathsf{k}} [\Omega]\sigma : \star$	By definition
$[\Omega](\Gamma, x : \sigma) = [\Omega]\Gamma, x : [\Omega]\sigma$	By definition
$[\Omega]\Delta \Vdash^{\mathsf{ectx}} [\Omega](\Gamma, x : \sigma)$	By rule ECTX-VAR

- Case

  A-ECTX-DCON
  $$\frac{\Delta \Vdash^{\mathsf{ectx}} \Gamma \qquad \Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \star}{\Delta \Vdash^{\mathsf{ectx}} \Gamma, D : \sigma}$$

  This case is similar to the case rule A-ECTX-VAR.

□

Lemma D.38 (Soundness of Promotion). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa_1 \rightsquigarrow \kappa_2 \dashv \Theta$, *then* $[\Theta]\kappa_1 = [\Theta]\kappa_2$. *If* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\kappa_1 = [\Omega]\kappa_2$.

Proof. By lemma D.18, we know $[\Omega]\kappa_1 = [\Omega]([\Theta]\kappa_1)$ and $[\Omega]\kappa_2 = [\Omega]([\Theta]\kappa_2)$. Thus we only need to prove that $[\Theta]\kappa_1 = [\Theta]\kappa_2$.

By a straightforward induction on the promotion judgment. All cases follow trivially.

□

Lemma D.39 (Soundness of Unification). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{u}} \kappa_1 \approx \kappa_2 \dashv \Theta$, *then* $[\Theta]\kappa_1 = [\Theta]\kappa_2$. *If* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\kappa_1 = [\Omega]\kappa_2$.

Proof. By lemma D.18, we know $[\Omega]\kappa_1 = [\Omega]([\Theta]\kappa_1)$ and $[\Omega]\kappa_2 = [\Omega]([\Theta]\kappa_2)$. Thus we only need to prove that $[\Theta]\kappa_1 = [\Theta]\kappa_2$.

By induction on the unification judgment.

- Case

$$\frac{}{\Delta \Vdash^{\mathsf{u}} \kappa \approx \kappa \dashv \Delta}\ \textsc{a-u-refl}$$

$[\Delta]\kappa = [\Delta]\kappa$.

- Case

$$\frac{\Delta \Vdash^{\mathsf{u}} \kappa_1 \approx \kappa_3 \dashv \Theta_1 \qquad \Theta_1 \Vdash^{\mathsf{u}} [\Theta_1]\kappa_2 \approx [\Theta_1]\kappa_4 \dashv \Theta}{\Delta \Vdash^{\mathsf{u}} \kappa_1 \rightarrow \kappa_2 \approx \kappa_3 \rightarrow \kappa_4 \dashv \Theta}\ \textsc{a-u-arrow}$$

$\Delta \longrightarrow \Theta_1$	By lemma D.4
$\Theta_1 \longrightarrow \Theta$	By lemma D.4
$\Delta \longrightarrow \Theta$	By lemma D.19
$[\Theta_1]\kappa_1 = [\Theta_1]\kappa_3$	By I.H.
$[\Theta]\kappa_1 = [\Theta]([\Theta_1]\kappa_1)$	By lemma D.18
$[\Theta]\kappa_3 = [\Theta]([\Theta_1]\kappa_3)$	By lemma D.18
$[\Theta]\kappa_1 = [\Theta]\kappa_3$	Follows directly
$[\Theta]\kappa_2 = [\Theta]\kappa_4$	Similarly
$[\Theta](\kappa_1 \rightarrow \kappa_2) = [\Theta](\kappa_3 \rightarrow \kappa_4)$	Follows directly

- Case

$$\frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa \rightsquigarrow \kappa_2 \dashv \Theta[\widehat{\alpha}]}{\Delta[\widehat{\alpha}] \Vdash^{\mathsf{u}} \widehat{\alpha} \approx \kappa \dashv \Theta[\widehat{\alpha} = \kappa_2]}\ \textsc{a-u-kvarL}$$

$[\Theta[\widehat{\alpha}]]\kappa = [\Theta[\widehat{\alpha}]]\kappa_2$	By Lemma D.38
$[\Theta[\widehat{\alpha} = \kappa_2]]\widehat{\alpha} = [\Theta[\widehat{\alpha} = \kappa_2]]\kappa_2$	By the definition of context application

By definition of promotion we know κ is free of $\widehat{\alpha}$

By Lemma D.3 we know κ_2 is free of $\widehat{\alpha}$.

$[\Theta[\widehat{\alpha} = \kappa_2]]\kappa_2 = [\Theta[\widehat{\alpha}]]\kappa_2 = [\Theta[\widehat{\alpha}]]\kappa = [\Theta[\widehat{\alpha} = \kappa_2]]\kappa$ |

- Case

$$\frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa \rightsquigarrow \kappa_2 \dashv \Theta[\widehat{\alpha}]}{\Delta[\widehat{\alpha}] \Vdash^{\mathsf{u}} \kappa \approx \widehat{\alpha} \dashv \Theta[\widehat{\alpha} = \kappa_2]}\ \textsc{a-u-kvarR}$$

This case is similar to the rule A-U-KVARL case.

□

Lemma D.40 (Soundness of Application Kinding). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{kapp}} \kappa_1 \bullet \kappa_2 : \kappa_3 \dashv \Theta$, *then* $[\Theta]\kappa_1 = [\Theta]\kappa_2 \rightarrow [\Theta]\kappa_3$. *If* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\kappa_1 = [\Omega]\kappa_2 \rightarrow [\Omega]\kappa_3$.

Proof. By lemma D.18, we know $[\Omega]\kappa_1 = [\Omega]([\Theta]\kappa_1)$ and $[\Omega]\kappa_2 = [\Omega]([\Theta]\kappa_2)$ and $[\Omega]\kappa_3 = [\Omega]([\Theta]\kappa_3)$. Thus we only need to prove that $[\Theta]\kappa_1 = [\Theta]\kappa_2 \rightarrow [\Theta]\kappa_3$.

By induction on the application kinding judgment.

- Case

$$\frac{\Delta[\widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha} = \widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2] \Vdash^{\mathsf{u}} \widehat{\alpha}_1 \approx \kappa \dashv \Theta}{\Delta[\widehat{\alpha}] \Vdash^{\mathsf{kapp}} \widehat{\alpha} \bullet \kappa : \widehat{\alpha}_2 \dashv \Theta}\ \textsc{a-kapp-kuvar}$$

$[\Theta]\widehat{\alpha}_1 = [\Theta]\kappa$	By lemma D.39

$$\begin{array}{l|l}
\Delta[\widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha} = \widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2] \longrightarrow \Theta & \text{By lemma D.4} \\
[\Theta]\widehat{\alpha} & \\
= [\Theta]([\Delta[\widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha} = \widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2]]\widehat{\alpha}) & \text{By lemma D.18} \\
= [\Theta]([\Delta[\widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha} = \widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2]]\widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2) & \text{By definition} \\
= [\Theta](\widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2) & \text{By lemma D.18} \\
= [\Theta]\widehat{\alpha}_1 \rightarrow [\Theta]\widehat{\alpha}_2 & \text{By definition} \\
= [\Theta]\kappa \rightarrow [\Theta]\widehat{\alpha}_2 & \text{By substituting the equation}
\end{array}$$

- Case

$$\frac{\Delta \Vdash^{u} \kappa_1 \approx \kappa \dashv \Theta}{\Delta \Vdash^{kapp} \kappa_1 \rightarrow \kappa_2 \bullet \kappa : \kappa_2 \dashv \Theta}\ \text{A-KAPP-ARROW}$$

$$\begin{array}{l|l}
[\Theta]\kappa_1 = [\Theta]\kappa & \text{By lemma D.39} \\
[\Theta](\kappa_1 \rightarrow \kappa_2) & \\
= [\Theta]\kappa_1 \rightarrow [\Theta]\kappa_2 & \text{By definition} \\
= [\Theta]\kappa \rightarrow [\Theta]\kappa_2 & \text{By substituting the equation}
\end{array}$$

□

Lemma D.41 (Soundness of Kinding). *If* Δ ok, *and* $\Delta \Vdash^{k} \sigma : \kappa \dashv \Theta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{k} [\Omega]\sigma : [\Omega]\kappa$.

Proof. By induction on the kinding judgment.

- Case

$$\frac{}{\Delta \Vdash^{k} \mathsf{Int} : \star \dashv \Delta}\ \text{A-K-NAT}$$

$[\Omega]\Delta \vdash^{k} [\Omega]\mathsf{Int} : [\Omega]\star$ By rule K-NAT.

- Case

$$\frac{(a : \kappa) \in \Delta}{\Delta \Vdash^{k} a : \kappa \dashv \Delta}\ \text{A-K-VAR}$$

$$\begin{array}{l|l}
(a : \kappa) \in \Delta & \text{Given} \\
(a : [\Omega]\kappa) \in [\Omega]\Delta & \text{By lemma D.33} \\
[\Omega]\Delta \vdash [\Omega]a : [\Omega]\kappa & \text{By rule K-VAR}
\end{array}$$

- Case

$$\frac{(T : \kappa) \in \Delta}{\Delta \Vdash^{k} T : \kappa \dashv \Delta}\ \text{A-K-TCON}$$

This case is similar to the rule A-K-VAR case.

- Case

$$\frac{}{\Delta \Vdash^{k} \rightarrow : \star \rightarrow \star \rightarrow \star \dashv \Delta}\ \text{A-K-ARROW}$$

$[\Omega]\Delta \vdash^{k} [\Omega] \rightarrow : [\Omega](\star \rightarrow \star \rightarrow \star)$ By rule K-ARROW.

- Case

$$\frac{\Delta \Vdash^{\mathsf{kv}} \kappa \qquad \Delta, a : \kappa \Vdash^{\mathsf{k}} \sigma : \kappa_2 \dashv \Theta, a : \kappa \qquad [\Theta]\kappa_2 = \star}{\Delta \Vdash^{\mathsf{k}} \forall a : \kappa . \sigma : \star \dashv \Theta}\ \textsc{a-k-forall}$$

$$\begin{array}{l|l}
\Theta \longrightarrow \Omega & \text{Given} \\
\Theta, a : \kappa \longrightarrow \Omega, a : \kappa & \text{By rule } \textsc{a-ctxe-tvar} \\
[\Omega, a : \kappa](\Delta, a : \kappa) \vdash^{\mathsf{k}} [\Omega, a : \kappa]\sigma : [\Omega, a : \kappa]\kappa_2 & \text{I.H.} \\
[\Omega]\Delta, a : [\Omega]\kappa \vdash^{\mathsf{k}} [\Omega, a : \kappa]\sigma : [\Omega, a : \kappa]([\Theta, a : \kappa]\kappa_2) & \text{By Lemma D.18} \\
[\Omega]\Delta, a : [\Omega]\kappa \vdash^{\mathsf{k}} [\Omega, a : \kappa]\sigma : [\Omega, a : \kappa]\star & \text{By definition} \\
[\Omega]\Delta, a : [\Omega]\kappa \vdash^{\mathsf{k}} [\Omega]\sigma : [\Omega]\star & \text{By property of context application} \\
[\Omega]\Delta \vdash^{\mathsf{k}} \forall a : [\Omega]\kappa . [\Omega]\sigma : [\Omega]\star & \text{By rule } \textsc{k-forall}
\end{array}$$

- Case

$$\frac{\Delta \Vdash^{\mathsf{k}} \tau_1 : \kappa_1 \dashv \Theta_1 \qquad \Theta_1 \Vdash^{\mathsf{k}} \tau_2 : \kappa_2 \dashv \Theta_2 \qquad \Theta_2 \Vdash^{\mathsf{kapp}} [\Theta_2]\kappa_1 \bullet [\Theta_2]\kappa_2 : \kappa_3 \dashv \Theta}{\Delta \Vdash^{\mathsf{k}} \tau_1\ \tau_2 : \kappa_3 \dashv \Theta}\ \textsc{a-k-app}$$

$$\begin{array}{l|l}
\Theta_1 \longrightarrow \Theta_2 & \text{By lemma D.6} \\
\Theta_2 \longrightarrow \Theta & \text{By lemma D.4} \\
\Theta_1 \longrightarrow \Omega & \text{By lemma D.19} \\
[\Omega]\Delta \vdash^{\mathsf{k}} [\Omega]\tau_1 : [\Omega]\kappa_1 & \text{I.H.} \\
[\Omega]\Delta \vdash^{\mathsf{k}} [\Omega]\tau_2 : [\Omega]\kappa_2 & \text{Similarly} \\
[\Omega]\kappa_1 = [\Omega](\kappa_2 \ \rightarrow\ \kappa_3) = [\Omega]\kappa_2 \ \rightarrow\ [\Omega]\kappa_3 & \text{By lemma D.40} \\
[\Omega]\Delta \vdash^{\mathsf{k}} [\Omega](\tau_1\ \tau_2) : [\Omega]\kappa_3 & \text{By rule } \textsc{k-app-p}
\end{array}$$

□

Lemma D.42 (Soundness of Typing Data Constructor Declaration). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{dc}}_{\tau'} \mathcal{D} \rightsquigarrow \tau \dashv \Theta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{\mathsf{dc}}_{\tau'} \mathcal{D} \rightsquigarrow \tau$.

Proof. We have

$$\frac{\Delta \Vdash^{\mathsf{k}} \overline{\tau_i}^{\,i} \ \rightarrow \tau : \star \dashv \Theta}{\Delta \Vdash^{\mathsf{dc}}_{\tau} D\, \overline{\tau_i}^{\,i} \rightsquigarrow \overline{\tau_i}^{\,i} \ \rightarrow \tau \dashv \Theta}\ \textsc{a-dc-decl}$$

Follows directly from lemma D.41 and rule $\textsc{dc-decl}$.

□

Lemma D.43 (Soundness of Typing Datatype Declaration). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow \Gamma \dashv \Theta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow [\Omega]\Gamma$.

Proof. We have

$$\frac{(T : \kappa) \in \Delta \qquad \Delta, \overline{\widehat{\alpha}_i}^{\,i} \Vdash^{\mathsf{u}} [\Delta]\kappa \approx (\overline{\widehat{\alpha}_i}^{\,i} \ \rightarrow \star) \dashv \Theta_1, \overline{\widehat{\alpha}_i = \kappa_i}^{\,i} \qquad \overline{\Theta_j, \overline{a_i : \kappa_i}^{\,i} \Vdash^{\mathsf{dc}}_{T\,\overline{a_i}^{\,i}} \mathcal{D}_j \rightsquigarrow \tau_j \dashv \Theta_{j+1}, \overline{a_i : \kappa_i}^{\,i}}^{\,j}}{\Delta \Vdash^{\mathsf{dt}} \textbf{data}\ T\, \overline{a_i}^{\,i} = \overline{\mathcal{D}_j}^{\,j \in 1..n} \rightsquigarrow \overline{D_j : \forall \overline{a_i : \kappa_i}^{\,i} . \tau_j}^{\,j} \dashv \Theta_{n+1}}\ \textsc{a-dt-decl}$$

$$\begin{array}{l|l}
\Delta, \overline{\widehat{\alpha}_i}^{\,i} \longrightarrow \Theta_1, \overline{\widehat{\alpha}_i = \kappa_i}^{\,i} & \text{By lemma D.4} \\
\Delta \longrightarrow \Theta_1 & \text{By Lemma D.17}
\end{array}$$

$\overline{\Theta_j,\ \overline{a_i : \kappa_i}^{\,i} \longrightarrow \Theta_{j+1},\ \overline{a_i : \kappa_i}^{\,i}}^{\,j}$	By lemma D.7
$\overline{\Theta_j \longrightarrow \Theta_{j+1}}^{\,j}$	By Lemma D.17
$\Theta_{n+1} \longrightarrow \Omega$	Given
$\overline{\Theta_j \longrightarrow \Omega}^{\,j}$	By Lemma D.19
$\Delta \longrightarrow \Omega$	By Lemma D.19
$\Theta_{n+1},\ \overline{a_i : \kappa_i}^{\,i} \longrightarrow \Omega,\ \overline{a_i : \kappa_i}^{\,i}$	By rule A-CTXE-TVAR
$\overline{\Theta_{j+1},\ \overline{a_i : \kappa_i}^{\,i} \longrightarrow \Omega,\ \overline{a_i : \kappa_i}^{\,i}}^{\,j}$	By Lemma D.19
$\overline{[\Omega,\ \overline{a_i : \kappa_i}^{\,i}](\Theta_{j+1},\ \overline{a_i : \kappa_i}^{\,i}) \vdash^{\mathsf{dc}}_{T\,\overline{a_i}^{\,i}} \mathcal{D}_j \rightsquigarrow [\Omega]\tau_j}^{\,j}$	By Lemma D.42
$\overline{[\Omega]\Theta_{j+1},\ \overline{a_i : [\Omega]\kappa_i}^{\,i} \vdash^{\mathsf{dc}}_{T\,\overline{a_i}^{\,i}} \mathcal{D}_j \rightsquigarrow [\Omega]\tau_j}^{\,j}$	By definition
$[\Omega]\Delta,\ \overline{a_i : [\Omega]\kappa_i}^{\,i} \vdash^{\mathsf{dc}}_{T\,\overline{a_i}^{\,i}} \mathcal{D}_j \rightsquigarrow [\Omega]\tau_j$	(1) By Lemma D.30
$(T : \kappa) \in \Delta$	Given
$(T : [\Omega]\kappa) \in [\Omega]\Delta$	(2) By lemma D.32
$\Theta_1,\ \overline{\widehat{\alpha}_i = \kappa_i}^{\,i} \longrightarrow \Omega,\ \overline{\widehat{\alpha}_i = \kappa_i}^{\,i}$	By rule A-CTXE-KUVARSOLVED
$[\Omega,\ \overline{\widehat{\alpha}_i = \kappa_i}^{\,i}]([\Delta]\kappa) = [\Omega,\ \overline{\widehat{\alpha}_i = \kappa_i}^{\,i}](\overline{\widehat{\alpha}_i}^{\,i} \rightarrow \star)$	By Lemma D.39
$[\Omega]([\Delta]\kappa) = \overline{[\Omega]\kappa_i}^{\,i} \rightarrow \star$	By definition
$[\Omega]\kappa = \overline{[\Omega]\kappa_i}^{\,i} \rightarrow \star$	(3) By Lemma D.18
$[\Omega]\Delta \vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow [\Omega]\Gamma$	By rule DT-DECL and (1), (2), (3)

□

Lemma D.44 (Soundness of Typing Program). *If* Ω ok, *and* $\Omega \Vdash^{\mathsf{ectx}} \Gamma$, *and* $\Omega; \Gamma \Vdash^{\mathsf{pgm}} pgm : \sigma$, *then* $[\Omega]\Omega; [\Omega]\Gamma \vdash^{\mathsf{pgm}} pgm : \sigma$.

Proof. By induction on the typing program judgment.

- Case

$$\textsc{a-pgm-expr}\quad \frac{[\Omega]\Omega; [\Omega]\Gamma \vdash e : \sigma}{\Omega; \Gamma \Vdash^{\mathsf{pgm}} e : \sigma}$$

The conclusion holds directly from the hypothesis and rule PGM-EXPR.

- Case

$$\textsc{a-pgm-dt}\quad \frac{\overline{\Theta_i \Vdash^{\mathsf{dt}} \mathcal{T}_i \rightsquigarrow \Gamma_i \dashv \Theta_{i+1}}^{\,i} \qquad \begin{array}{c}\Theta_1 = \Omega,\ \overline{\widehat{\alpha}_i}^{\,i},\ \overline{T_i : \widehat{\alpha}_i}^{\,i} \\ \Theta_{n+1} \longrightarrow\!\!\!\!\twoheadrightarrow \Omega'\end{array} \qquad \Omega'; \Gamma,\ \overline{\Gamma_i}^{\,i} \Vdash^{\mathsf{pgm}} pgm : \sigma}{\Omega; \Gamma \Vdash^{\mathsf{pgm}} \mathbf{rec}\ \overline{\mathcal{T}_i}^{\,i \in 1..n}; pgm : \sigma}$$

$\overline{\Theta_i \longrightarrow \Theta_{i+1}}^{\,i}$	By lemma D.8
$\Theta_{n+1} \longrightarrow \Omega'$	By lemma D.13
$\overline{\Theta_i \longrightarrow \Omega'}^{\,i}$	By lemma D.19
$\Omega,\ \overline{\widehat{\alpha}_i}^{\,i \in 1..n},\ \overline{T_i : \widehat{\alpha}_i}^{\,i \in 1..n} \longrightarrow \Omega'$	$\Theta_1 = \Omega,\ \overline{\widehat{\alpha}_i}^{\,i \in 1..n},\ \overline{T_i : \widehat{\alpha}_i}^{\,i \in 1..n}$
$\Omega' = \Omega'',\ \overline{\widehat{\alpha}_i = \kappa_i, \Omega_i}^{\,i \in 1..n},\ \overline{T_i : \widehat{\alpha}_i, \Omega'_i}^{\,i \in 1..n}$	By repeatedly applying lemma D.17
$\overline{\Omega_i\ \mathbf{soft}}^{\,i} \wedge \overline{\Omega'_i\ \mathbf{soft}}^{\,i}$	Above
$\Omega \longrightarrow \Omega''$	Above
$[\Omega']\Omega'$	

$$\begin{array}{l|l}
= [\Omega'', \overline{\widehat{\alpha}_i = \kappa_i, \Omega_i}^{i\in 1..n}, \overline{T_i : \widehat{\alpha}_i, \Omega'_i}^{i\in 1..n}] & \\
\quad (\Omega'', \overline{\widehat{\alpha}_i = \kappa_i, \Omega_i}^{i\in 1..n}, \overline{T_i : \widehat{\alpha}_i, \Omega'_i}^{i\in 1..n}) & \\
= [\Omega'']\Omega'', \overline{T_i : \kappa_i}^{i\in 1..n} & \text{for some } \kappa_i\text{, by definition} \\
= [\Omega]\Omega, \overline{T_i : \kappa_i}^{i\in 1..n} & \text{By lemma D.34} \\
\overline{[\Omega']\Theta_i \vdash^{\mathsf{dt}} \mathcal{T}_i \leadsto [\Omega']\Gamma_i}^{i\in 1..n} & \text{By lemma D.43} \\
\overline{[\Omega']\Theta_i = [\Omega']\Omega'}^{i\in 1..n} & \text{By lemma D.28} \\
\overline{[\Omega']\Omega' \vdash^{\mathsf{dt}} \mathcal{T}_i \leadsto [\Omega']\Gamma_i}^{i\in 1..n} & \text{By substituting the equation} \\
\overline{[\Omega]\Omega, \overline{T_i : \kappa_i}^{i\in 1..n} \vdash^{\mathsf{dt}} \mathcal{T}_i \leadsto [\Omega']\Gamma_i}^{i\in 1..n} & \text{(1) By substituting the equation} \\
[\Omega'](\Gamma, \overline{\Gamma_i}^{i\in 1..n}) = [\Omega']\Gamma, \overline{[\Omega']\Gamma_i}^{i\in 1..n} & \text{By definition} \\
[\Omega']\Gamma & \\
= [\Omega'', \overline{\widehat{\alpha}_i = \kappa_i, \Omega_i}^{i\in 1..n}, \overline{T_i : \widehat{\alpha}_i, \Omega'_i}^{i\in 1..n}]\Gamma & \\
= [\Omega'']\Gamma & \text{By definition and freshness} \\
= [\Omega]\Gamma & \text{By lemma D.27} \\
[\Omega']\Omega'; [\Omega'](\Gamma, \overline{\Gamma_i}^{i\in 1..n}) \vdash^{\mathsf{pgm}} pgm : \sigma & \text{I.H.} \\
[\Omega]\Omega, \overline{T_i : \kappa_i}^{i\in 1..n}; [\Omega]\Gamma, \overline{[\Omega']\Gamma_i}^{i\in 1..n} \vdash^{\mathsf{pgm}} pgm : \sigma & \text{(2) By substituting equations} \\
[\Omega]\Omega; [\Omega]\Gamma \vdash^{\mathsf{pgm}} pgm : \sigma & \text{By rule PGM-DT and (1), (2)}
\end{array}$$

□

D.2.6 Completeness of Algorithm.

Lemma D.45 (Completeness of Promotion). *Given* Δ ok*, and* $\Delta \longrightarrow \Omega$*, and* $\Delta \vdash \widehat{\alpha}$*, and* $\Delta \Vdash^{\mathsf{kv}} \kappa$*, and* $[\Delta]\widehat{\alpha} = \widehat{\alpha}$*, and* $[\Delta]\kappa = \kappa$*, if* κ *is free of* $\widehat{\alpha}$*, then there exists* κ_2*,* Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$*, and* $\Omega \longrightarrow \Omega'$*, and* $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa \leadsto \kappa_2 \dashv \Theta$*.*

Proof. By induction on κ.

- $\kappa = \star$. Then by rule A-PR-STAR, we have $\Theta = \Delta$, and $\Omega' = \Omega$.
- $\kappa = \kappa_1 \ \rightarrow\ \kappa_2$.

$$\begin{array}{l|l}
\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa_1 \leadsto \kappa_3 \dashv \Delta_1 \wedge \Delta_1 \longrightarrow \Omega_1 \wedge \Omega \longrightarrow \Omega_1 & \text{I.H.} \\
\Delta_1 \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} [\Delta_1]\kappa_2 \leadsto \kappa_4 \dashv \Theta \wedge \Theta \longrightarrow \Omega' \wedge \Omega_1 \longrightarrow \Omega' & \text{I.H.} \\
\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa_1 \ \rightarrow\ \kappa_2 \leadsto \kappa_3 \ \rightarrow\ \kappa_4 \dashv \Theta & \text{By rule A-PR-ARROW} \\
\Omega \longrightarrow \Omega' & \text{By lemma D.19}
\end{array}$$

- $\kappa = \widehat{\beta}$.
  - $\widehat{\beta}$ is to the left of $\widehat{\alpha}$. Then by rule A-PR-KUVARL, we have $\Theta = \Delta$, and $\Omega' = \Omega$.
  - $\widehat{\beta}$ is to the right of $\widehat{\alpha}$. Then by rule A-PR-KUVARR, we have $\Theta = \Delta[\widehat{\beta}_2, \widehat{\alpha}][\widehat{\beta} = \widehat{\beta}_2]$.

$$\begin{array}{l|l}
\Omega = \Omega[\widehat{\alpha} = \kappa_3][\widehat{\beta} = \kappa_4] & \text{By Lemma D.17} \\
\Delta[\widehat{\alpha}][\widehat{\beta}] \longrightarrow \Omega & \text{Given} \\
\Delta[\widehat{\beta}_2, \widehat{\alpha}][\widehat{\beta}] \longrightarrow \Omega[\widehat{\beta}_2 = [\Omega]\kappa_4, \widehat{\alpha} = \kappa_3][\widehat{\beta} = \kappa_4] & \text{By Lemma D.23} \\
\Delta[\widehat{\beta}_2, \widehat{\alpha}][\widehat{\beta} = \widehat{\beta}_2] \longrightarrow \Omega[\widehat{\beta}_2 = [\Omega]\kappa_4, \widehat{\alpha} = \kappa_3][\widehat{\beta} = \kappa_4] & \text{By Lemma D.24} \\
\Omega_1 = \Omega[\widehat{\beta}_2 = [\Omega]\kappa_4, \widehat{\alpha} = \kappa_3][\widehat{\beta} = \kappa_4] & \text{Let}
\end{array}$$

□

Lemma D.46 (Completeness of Unification). *Given* Δ ok, *and* $\Delta \longrightarrow \Omega$, *and* $\Delta \Vdash^{\mathsf{kv}} \kappa_1$ *and* $\Delta \Vdash^{\mathsf{kv}} \kappa_2$, *and* $[\Delta]\kappa_1 = \kappa_1$ *and* $[\Delta]\kappa_2 = \kappa_2$, *if* $[\Omega]\kappa_1 = [\Omega]\kappa_2$, *then there exists* Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$ *and* $\Delta \Vdash^{\mathsf{u}} \kappa_1 \approx \kappa_2 \dashv \Theta$.

Proof. By case analysis on κ_1 on κ_2.

- $\kappa_1 = \star$ and $\kappa_2 = \star$. Then by rule A-U-REFL, we have $\Theta = \Delta$, and $\Omega' = \Omega$.
- $\kappa_1 = \kappa_{11} \rightarrow \kappa_{12}$ and $\kappa_2 = \kappa_{21} \rightarrow \kappa_{22}$.

$[\Omega](\kappa_{11} \rightarrow \kappa_{12})$	
$= [\Omega]\kappa_{11} \rightarrow [\Omega]\kappa_{12}$	By definition
$= [\Omega](\kappa_{21} \rightarrow \kappa_{22})$	Given
$= [\Omega]\kappa_{21} \rightarrow [\Omega]\kappa_{22}$	By definition
$[\Omega]\kappa_{11} = [\Omega]\kappa_{21} \wedge [\Omega]\kappa_{12} = [\Omega]\kappa_{22}$	Follows directly
$\Delta \Vdash^{\mathsf{u}} \kappa_{11} \approx \kappa_{12} \dashv \Theta_1 \wedge \Theta_1 \longrightarrow \Omega_1 \wedge \Omega \longrightarrow \Omega_1$	(1) I.H.
$[\Omega_1]([\Theta_1]\kappa_{12}) = [\Omega_1]\kappa_{12} = [\Omega_1]([\Omega]\kappa_{12})$	By lemma D.18
$= [\Omega_1]([\Omega]\kappa_{22})$	Known
$= [\Omega_1]\kappa_{22} = [\Omega_1]([\Theta_1]\kappa_{22})$	By lemma D.18
$\Theta_1 \Vdash^{\mathsf{u}} [\Theta_1]\kappa_{21} \approx [\Theta_2]\kappa_{22} \dashv \Theta \wedge \Theta \longrightarrow \Omega' \wedge \Omega_1 \longrightarrow \Omega'$	(2) I.H.
$\Delta \Vdash^{\mathsf{u}} \kappa_{11} \rightarrow \kappa_{12} \approx \kappa_{21} \rightarrow \kappa_{22} \dashv \Theta$	By rule A-U-ARROW and (1) (2)
$\Omega \longrightarrow \Omega'$	By lemma D.19

- $\kappa_2 = \widehat{\alpha}$. Then we have $[\Omega]\widehat{\alpha} = [\Omega]\kappa_1$.
  - $\kappa_1 = \widehat{\alpha}$. Then by rule A-U-REFL, we have $\Theta = \Delta$, and $\Omega' = \Omega$.
  - Otherwise κ_1 must be free of $\widehat{\alpha}$.

$\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \kappa_1 \rightsquigarrow \kappa_2 \dashv \Theta_1 \wedge \Theta_1 \longrightarrow \Omega' \wedge \Omega \longrightarrow \Omega'$	By Lemma D.45
$\Delta \vdash \widehat{\alpha}$	Given
$\Theta_1 = \Theta_{11}, \widehat{\alpha}, \Theta_{12}$	By definition of promotion
$\Theta = \Theta_{11}, \widehat{\alpha} = \kappa_2, \Theta_{12}$	Let
$\Delta \Vdash^{\mathsf{u}} \kappa_1 \approx \widehat{\alpha} \dashv \Theta$	By rule A-U-KVARR
$[\Omega']\widehat{\alpha}$	
$= [\Omega']([\Omega]\widehat{\alpha})$	By Lemma D.18
$= [\Omega']([\Omega]\kappa_1)$	Given
$= [\Omega']\kappa_1$	By Lemma D.18
$= [\Omega']\kappa_2$	By Lemma D.38
$\Theta \longrightarrow \Omega'$	By Lemma D.24

- The case when $\kappa_1 = \widehat{\alpha}$ is the same.

□

Lemma D.47 (Completeness of Application Kinding). *Given* Δ ok, *and* $\Delta \longrightarrow \Omega$, *and* $\Delta \Vdash^{\mathsf{kv}} \kappa$ *and* $\Delta \Vdash^{\mathsf{kv}} \kappa'$, *and* $[\Delta]\kappa = \kappa$ *and* $[\Delta]\kappa' = \kappa'$, *if* $[\Omega]\kappa = [\Omega]\kappa' \rightarrow \kappa_1$, *then there exists* κ_2, Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$ *and* $\Delta \Vdash^{\mathsf{kapp}} \kappa \bullet \kappa' : \kappa_2 \dashv \Theta$, *and* $[\Omega']\kappa_2 = \kappa_1$.

Proof. By induction on κ.

- $\kappa = \widehat{\alpha}$ for some $\widehat{\alpha}$ and $[\Omega]\widehat{\alpha} = [\Omega]\kappa' \rightarrow \kappa_1$.

$\Delta = \Delta_1, \widehat{\alpha}, \Delta_2$	Assume
$\Delta \longrightarrow \Delta_1, \widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha}, \Delta_2$	By lemma D.22
$\Delta_1, \widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha}, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha} = \widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2, \Delta_2$	By lemma D.23

$\Delta_3 = \Delta_1, \widehat{\alpha}_1, \widehat{\alpha}_2, \widehat{\alpha} = \widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2, \Delta_2$	Let
$\Omega = \Omega_1, \widehat{\alpha} = \kappa_3, \Omega_2$	Assume
$\Omega \longrightarrow \Omega_1, \widehat{\alpha}_1 = [\Omega]\kappa', \widehat{\alpha}_2 = \kappa_1, \widehat{\alpha} = \kappa_3, \Omega_2$	By lemma D.21
$\Omega \longrightarrow \Omega_1, \widehat{\alpha}_1 = [\Omega]\kappa', \widehat{\alpha}_2 = \kappa_1, \widehat{\alpha} = \widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2, \Omega_2$	By lemma D.23 and lemma D.19
$\Omega_3 = \Omega_1, \widehat{\alpha}_1 = [\Omega]\kappa', \widehat{\alpha}_2 = \kappa_1, \widehat{\alpha} = \widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2, \Omega_2$	Let
$\Delta \longrightarrow \Omega$	Given
$\Delta_3 \longrightarrow \Omega_3$	By lemma D.23 and lemma D.25
$\Delta_3 \Vdash^{\mu} \widehat{\alpha}_1 \approx \kappa' \dashv \Theta \wedge \Theta \longrightarrow \Omega' \wedge \Omega_3 \longrightarrow \Omega'$	By lemma D.46
$\Delta \Vdash^{\mathsf{kapp}} \widehat{\alpha} \bullet \kappa' : \widehat{\alpha}_2 \dashv \Theta$	By rule A-KAPP-KUVAR
$\Omega \longrightarrow \Omega'$	By lemma D.19
$[\Omega']\widehat{\alpha}_2$	
$= [\Omega_3]\widehat{\alpha}_2$	By lemma D.26
$= \kappa_1$	

- Case $\kappa = \kappa_{21} \rightarrow \kappa_{22}$.

$[\Omega]\kappa_{21} = [\Omega]\kappa' \wedge [\Omega]\kappa_{22} = \kappa_1$	Follows directly
$\Delta \Vdash^{\mu} \kappa_{21} \approx \kappa_1 \dashv \Theta \wedge \Theta \longrightarrow \Omega' \wedge \Omega \longrightarrow \Omega'$	By lemma D.46
$\Delta \Vdash^{\mathsf{kapp}} \kappa \bullet \kappa_1 : \kappa_{22} \dashv \Theta$	By rule A-KAPP-ARROW
$[\Omega']\kappa_{22} = [\Omega]\kappa_{22} = \kappa_1$	By lemma D.26

□

Lemma D.48 (Completeness of Kinding). *Given* Δ ok *and* $\Delta \longrightarrow \Omega$, *if* $[\Omega]\Delta \vdash^{\mathsf{k}} [\Omega]\sigma : \kappa$, *then there exists* Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$ *and* $\Delta \Vdash^{\mathsf{k}} \sigma : \kappa' \dashv \Theta$, *and* $[\Omega']\kappa' = \kappa$.

Proof. By induction on the kinding judgment.

- Case

K-NAT
$$\frac{}{\Sigma \vdash^{\mathsf{k}} \mathsf{Int} : \star}$$

$\Delta \Vdash^{\mathsf{k}} \mathsf{Int} : \star \dashv \Delta$	By rule A-K-NAT
$\Theta = \Delta$	Let
$\Omega' = \Omega$	Let

- Case

K-VAR
$$\frac{(a : \kappa) \in \Sigma}{\Sigma \vdash^{\mathsf{k}} a : \kappa}$$

$(a : \kappa) \in [\Omega]\Delta$	Given
$(a : \kappa_2) \in \Delta \wedge [\Omega]\kappa_2 = \kappa$	By inversion
$\Delta \Vdash^{\mathsf{k}} a : \kappa_2 \dashv \Delta$	By rule A-K-VAR
$\Theta = \Delta$	Let
$\Omega' = \Omega$	Let

- Case

K-TCON
$$\frac{(T : \kappa) \in \Sigma}{\Sigma \vdash^{\mathsf{k}} T : \kappa}$$

Similar as the case for rule K-VAR.

- Case

$$\frac{}{\Sigma \vdash^{k} \rightarrow : \star \rightarrow \star \rightarrow \star}\ \textsc{K-Arrow}$$

Similar as the case for rule K-NAT.

- Case

$$\frac{\Sigma, a : \kappa \vdash^{k} \sigma : \star}{\Sigma \vdash^{k} \forall a : \kappa . \sigma : \star}\ \textsc{K-Forall}$$

$\Delta, a : \kappa \longrightarrow \Omega, a : \kappa$	By rule A-CTXE-TVAR
$[\Omega, a : \kappa](\Delta, a : \kappa) \vdash^{k} [\Omega]\sigma : \star$	Given
$\Delta, a : \kappa \Vdash^{k} \sigma : \kappa_1 \dashv \Theta_1 \land \Theta_1 \longrightarrow \Omega_1 \land \Omega, a : \kappa \longrightarrow \Omega_1 \land [\Omega_1]\kappa_1 = \star$	I.H.
$\Theta_1 = \Theta, a : \kappa$	By inversion
$\Delta \Vdash^{k} \forall a : \kappa . \sigma : \kappa \dashv \Theta$	By rule A-K-FORAL
$\Omega, a : \kappa \longrightarrow \Omega_1$	Known
$\Omega_1 = \Omega_{11}, a : \kappa, \Omega_{12} \land \Omega \longrightarrow \Omega_{11}$	By lemma D.17
$\Omega' = \Omega_{11}$	Let

- Case

$$\frac{\Sigma \vdash^{k} \tau_1 : \kappa_1 \rightarrow \kappa_2 \quad \Sigma \vdash^{k} \tau_2 : \kappa_1}{\Sigma \vdash^{k} \tau_1\ \tau_2 : \kappa_2}\ \textsc{K-App}$$

$[\Omega]\Delta \vdash^{k} [\Omega]\tau_1 : \kappa_1 \rightarrow \kappa_2$	Given
$\Delta \Vdash^{k} \tau_1 : \kappa_1' \dashv \Theta_1 \land \Theta_1 \longrightarrow \Omega_1 \land \Omega \longrightarrow \Omega_1 \land [\Omega_1]\kappa_1' = \kappa_1 \rightarrow \kappa_2$	I.H.
$\Delta \longrightarrow \Theta_1$	By lemma D.4
$\Delta \longrightarrow \Omega_1$	By lemma D.19
$[\Omega]\Delta \vdash^{k} [\Omega]\tau_2 : \kappa_1$	Given
$[\Omega]\tau_2 = \tau_2 = [\Omega_1]\tau_2$	By definition
$[\Omega]\Delta \vdash^{k} [\Omega_1]\tau_2 : \kappa_1$	Follows directly
$[\Omega]\Delta$	
$= [\Omega]\Omega$	By lemma D.28
$= [\Omega_1]\Omega_1$	By lemma D.34
$= [\Omega_1]\Theta_1$	By lemma D.28
$[\Omega_1]\Theta_1 \vdash^{k} [\Omega_1]\tau_2 : \kappa_1$	Follows directly
$\Theta_1 \Vdash^{k} \tau_2 : \kappa_2' \dashv \Theta_2 \land \Theta_2 \longrightarrow \Omega_2 \land \Omega_1 \longrightarrow \Omega_2 \land [\Omega_2]\kappa_2' = \kappa_1$	I.H.
$[\Omega_2]\kappa_1' = [\Omega_1]\kappa_1' = \kappa_1 \rightarrow \kappa_2$	By lemma D.34
$\Theta_2 \Vdash^{kapp} [\Theta_2]\kappa_1' \bullet [\Theta_2]\kappa_2' : \kappa_3 \dashv \Theta \land \Theta \longrightarrow \Omega' \land \Omega_2 \longrightarrow \Omega' \land [\Omega']\kappa_3 = \kappa_2$	By lemma D.47
$\Delta \Vdash^{k} \tau_1\ \tau_2 : \kappa_3 \dashv \Theta$	By rule A-K-APP
$\Omega \longrightarrow \Omega'$	By lemma D.19

□

Lemma D.49 (Completeness of Typing Data Constructor Declaration). *Given* Δ ok *and* $\Delta \longrightarrow \Omega$, *if* $[\Omega]\Delta \vdash^{dc}_{\tau'} \mathcal{D} \rightsquigarrow \tau$, *then there exists* Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$ *and* $\Delta \Vdash^{dc}_{\tau'} \mathcal{D} \rightsquigarrow \tau \dashv \Theta$.

Proof. Given

$$\textsc{dc-decl}\quad \frac{\Sigma \vdash^{\mathsf{k}} \overline{\tau_i}^{\,i} \rightarrow \tau : \star}{\Sigma \vdash^{\mathsf{dc}}_{\tau} D\,\overline{\tau_i}^{\,i} \rightsquigarrow \overline{\tau_i}^{\,i} \rightarrow \tau}$$

Follows directly from lemma D.48 and rule A-DC-DECL.

□

Lemma D.50 (Completeness of Typing Datatype Declaration). *Given* Δ ok, *and* $\Delta \longrightarrow \Omega$, *if* $[\Omega]\Delta \vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow \Psi$, *then there exists* Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$ *and* $\Delta \Vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow \Gamma \dashv \Theta$ *and* $\Psi = [\Omega']\Gamma$.

Proof. We have

$$\textsc{dt-decl}\quad \frac{(T : \overline{\kappa_i}^{\,i} \rightarrow \star) \in \Sigma \qquad \overline{\Sigma,\, \overline{a_i : \kappa_i}^{\,i} \vdash^{\mathsf{dc}}_{T\,\overline{a_i}^{\,i}} \mathcal{D}_j \rightsquigarrow \tau_j}^{\,j}}{\Sigma \vdash^{\mathsf{dt}} \textbf{data}\ T\,\overline{a_i}^{\,i} = \overline{\mathcal{D}_j}^{\,j} \rightsquigarrow \overline{D_j : \forall \overline{a_i : \kappa_i}^{\,i}.\tau_j}^{\,j}}$$

$(T : \overline{\kappa_i}^{\,i} \rightarrow \star) \in [\Omega]\Delta$	Given
$(T : \kappa) \in \Delta \wedge [\Omega]\kappa = \overline{\kappa_i}^{\,i} \rightarrow \star$	By inversion
$\Delta \longrightarrow \Omega$	Given
$\Delta,\, \overline{\widehat{\alpha}_i}^{\,i} \longrightarrow \Omega,\, \overline{\widehat{\alpha}_i = \kappa_i}^{\,i}$	By rule A-CTXE-SOLVE
$[\Omega,\, \overline{\widehat{\alpha}_i = \kappa_i}^{\,i}]\kappa = [\Omega,\, \overline{\widehat{\alpha}_i = \kappa_i}^{\,i}](\overline{\widehat{\alpha}_i}^{\,i} \rightarrow \star) = \overline{\kappa_i}^{\,i} \rightarrow \star$	Follows directly
$\Delta,\, \overline{\widehat{\alpha}_i}^{\,i} \Vdash^{\mathsf{u}} [\Delta]\kappa \approx \overline{\widehat{\alpha}_i}^{\,i} \rightarrow \star \dashv \Delta_1$	
$\wedge \Delta_1 \longrightarrow \Omega_1 \wedge \Omega,\, \overline{\widehat{\alpha}_i = \kappa_i}^{\,i} \longrightarrow \Omega_1$	By lemma D.46
$\Delta_1 = \Theta_1,\, \overline{\widehat{\alpha}_i = \kappa'_i}^{\,i}$	By unification and promotion
$\Delta,\, \overline{\widehat{\alpha}_i}^{\,i} \longrightarrow \Delta_1$	By lemma D.4
$\Delta \longrightarrow \Theta_1$	By Lemma D.17
$\Omega_1 = \Omega_{11},\, \overline{\widehat{\alpha}_i = \kappa''_i}^{\,i}, \Omega_{12} \wedge \Omega_{12}$ **soft** $\wedge\, \Theta_1 \longrightarrow \Omega_{11}$	
$\wedge [\Omega_{11}]\kappa''_i = [\Omega_{11}]\kappa'_i$	By Lemma D.17
$\Omega \longrightarrow \Omega_{11} \wedge [\Omega_{11}]\kappa''_1 = \kappa_i$	By Lemma D.17
$\Theta_1,\, \overline{a_i : \kappa'_i}^{\,i} \longrightarrow \Omega_{11},\, \overline{a_i : \kappa'_i}^{\,i}$	By rule A-CTXE-TVAR
$[\Omega_{11},\, \overline{a_i : \kappa'_i}^{\,i}](\Theta_1,\, \overline{a_i : \kappa'_i}^{\,i})$	
$= [\Omega_{11}]\Theta_1,\, \overline{a_i : [\Omega_{11}]\kappa'_i}^{\,i}$	By definition
$= [\Omega_{11}]\Theta_1,\, \overline{a_i : \kappa_i}^{\,i}$	By equations
$= [\Omega_{11}]\Omega_{11},\, \overline{a_i : \kappa_i}^{\,i}$	By Lemma D.28
$= [\Omega]\Omega,\, \overline{a_i : \kappa_i}^{\,i}$	By Lemma D.34
$= [\Omega]\Delta,\, \overline{a_i : \kappa_i}^{\,i}$	By Lemma D.28
$[\Omega]\Delta,\, \overline{a_i : \kappa_i}^{\,i} \vdash^{\mathsf{dc}}_{T\,\overline{a_i}^{\,i}} \mathcal{D}_1 \rightsquigarrow \tau_1$	Given
$[\Omega_{11},\, \overline{a_i : \kappa'_i}^{\,i}](\Theta_1,\, \overline{a_i : \kappa'_i}^{\,i}) \vdash^{\mathsf{dc}}_{T\,\overline{a_i}^{\,i}} \mathcal{D}_1 \rightsquigarrow \tau_1$	By equations
$(\Theta_1,\, \overline{a_i : \kappa'_i}^{\,i}) \Vdash^{\mathsf{dc}}_{T\,\overline{a_i}^{\,i}} \mathcal{D}_1 \rightsquigarrow \tau_1 \dashv \Theta_2,\, \overline{a_i : \kappa'_i}^{\,i}$	By Lemma D.49 and property of kinding
$\wedge \Theta_2,\, \overline{a_i : \kappa'_i}^{\,i} \longrightarrow \Omega_2 \wedge \Omega_{11},\, \overline{a_i : \kappa'_i}^{\,i} \longrightarrow \Omega_2$	By Lemma D.49
$\Omega_2 = \Omega_{21},\, \overline{a_i : \kappa'_i}^{\,i}, \Omega_{22} \wedge \Omega_{11} \longrightarrow \Omega_{21} \wedge \Omega_{22}$ **soft**	By Lemma D.17
$\Theta_2 \longrightarrow \Omega_{21}$	By Lemma D.17

We repeat the process for each j. Let $\Theta_{n+1}, \overline{a_i : \kappa'_i}^{\,i}$ and Ω_{n+1} be the final output context and the complete context. And $\Theta_{n+1}, \overline{a_i : \kappa'_i}^{\,i} \longrightarrow \Omega_{n+1}$.
By Lemma D.17 we have $\Omega_{n+1} = \Omega', \overline{a_i : \kappa'_i}^{\,i}, \Omega'' \land \Theta_{n+1} \longrightarrow \Omega' \land \Omega''$ **soft**.
By lemma D.19 we have $\Omega \longrightarrow \Omega'$.
By equations we have $[\Omega']\kappa'_i = [\Omega_{11}]\kappa'_i = \kappa_i$.
So collecting all the hypothesis, by rule A-DT-DECL we get $\Delta \Vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow \Gamma \dashv \Theta$. And $[\Omega']\Gamma = \Psi$.

□

Theorem D.51 (Completeness of Typing a Group). *Given* Ω ok, *if* $[\Omega]\Omega \vdash^{\mathsf{grp}} \textbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \rightsquigarrow \overline{\kappa_i}^{\,i}; \overline{\Psi_i}^{\,i}$, *then there exists* $\overline{\kappa'_i}^{\,i}, \overline{\Gamma_i}^{\,i}, \Theta$, *and* Ω', *such that* $\Omega \Vdash^{\mathsf{grp}} \textbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \rightsquigarrow \overline{\kappa'_i}^{\,i}; \overline{\Gamma_i}^{\,i} \dashv \Theta$, *where* $\Theta \longrightarrow \Omega'$, *and* $\overline{[\Omega']\kappa'_i = \kappa_i}^{\,i}$, *and* $\overline{\Psi_i = [\Omega']\Gamma_i}^{\,i}$.

Proof. We have

$$\begin{array}{l|l}
\Theta_1 = \Omega, \overline{\widehat{\alpha}_i}^{\,i\in 1..n}, \overline{T_i : \widehat{\alpha}_i}^{\,i\in 1..n} & \text{Let} \\
\Omega_1 = \Omega, \overline{\widehat{\alpha}_i = \kappa_i}^{\,i\in 1..n}, \overline{T_i : \widehat{\alpha}_i}^{\,i\in 1..n} & \text{Let} \\
\Theta_1 \longrightarrow \Omega_1 & \text{By lemma D.20} \\
[\Omega]\Omega, \overline{T_i : \kappa_i}^{\,i} \vdash^{\mathsf{dt}} \mathcal{T}_1 \rightsquigarrow \Psi_1 & \text{Given} \\
[\Omega_1]\Theta_1 \vdash^{\mathsf{dt}} \mathcal{T}_1 \rightsquigarrow \Psi_1 & \text{I.E.} \\
\Theta_1 \Vdash^{\mathsf{dt}} \mathcal{T}_1 \rightsquigarrow \Gamma_1 \dashv \Theta_2 \land \Theta_2 \longrightarrow \Omega_2 \land \Omega_1 \longrightarrow \Omega_2 \land \Psi_1 = [\Omega_2]\Gamma_1 & \text{(1) By lemma D.50} \\
[\Omega]\Omega, \overline{T_i : \kappa_i}^{\,i} \vdash^{\mathsf{dt}} \mathcal{T}_2 \rightsquigarrow \Psi_2 & \text{Given} \\
[\Omega_2]\Theta_2 & \\
= [\Omega_2]\Omega_2 & \text{By lemma D.28} \\
= [\Omega_1]\Omega_1 & \text{By lemma D.34} \\
= [\Omega]\Omega, \overline{T_i : \kappa_i}^{\,i} & \text{By definition} \\
[\Omega_2]\Theta_2 \vdash^{\mathsf{dt}} \mathcal{T}_2 \rightsquigarrow \Psi_2 & \text{Substitute the equation} \\
\Theta_2 \Vdash^{\mathsf{dt}} \mathcal{T}_2 \rightsquigarrow \Gamma_2 \dashv \Theta_3 \land \Theta_3 \longrightarrow \Omega_3 \land \Omega_2 \longrightarrow \Omega_3 \land \Psi_2 = [\Omega_3]\Gamma_2 & \text{(2) By lemma D.50} \\
\text{By repeating the process from (1) to (2) for each } i\text{, we can get} & \\
\overline{\Theta_i \Vdash^{\mathsf{dt}} \mathcal{T}_i \rightsquigarrow \Gamma_i \dashv \Theta_{i+1}}^{\,i\in 1..n} \land \Theta_{n+1} \longrightarrow \Omega_{n+1} \land \overline{\Psi_i = [\Omega_{i+1}]\Gamma_i}^{\,i} & \\
\Omega' = \Omega_{n+1} & \text{Let} \\
\overline{[\Omega']\widehat{\alpha}_i = [\Omega']([\Omega_1]\widehat{\alpha}_i)}^{\,i} & \text{By lemma D.18} \\
\overline{[\Omega']\widehat{\alpha}_i = \kappa_i}^{\,i} & \text{Namely} \\
\overline{[\Omega']\Gamma_i = [\Omega_{i+1}]\Gamma_i}^{\,i} & \text{By lemma D.27} \\
\overline{[\Omega']\Gamma_i = \Psi_i}^{\,i} & \text{Namely}
\end{array}$$

□

E PROOF FOR HASKELL98 WITH KIND PARAMETERS

E.1 List of Lemmas

Theorem E.1 (Principality of Haskell98 with Kind Parameters). *If* $\Sigma \vdash^{\mathsf{grp}} \textbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \rightsquigarrow \overline{\kappa_i}^{\,i}; \overline{\Psi_i}^{\,i}$, *then there exists some* $\overline{\kappa'_i}^{\,i}$ *such that* $\Sigma \vdash \textbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \rightsquigarrow^{\mathsf{p}} \overline{\kappa'_i}^{\,i}$.

Theorem E.2 (Completeness of Typing Programs with Kind Parameters). *Given algorithmic contexts* Ω, Γ, *and a program* pgm, *if* $[\Omega]\Omega; [\Omega]\Gamma \vdash^{\mathsf{pgm}} pgm : \sigma$, *then* $\Omega; \Gamma \Vdash^{\mathsf{pgm}} pgm : \sigma$.

E.2 Proofs

Theorem E.1 (Principality of Haskell98 with Kind Parameters). *If* $\Sigma \vDash^{\mathsf{grp}} \mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto \overline{\kappa_i}^{\,i}\,; \overline{\Psi_i}^{\,i}$*, then there exists some* $\overline{\kappa_i'}^{\,i}$ *such that* $\Sigma \vdash \mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto^{\mathsf{p}} \overline{\kappa_i'}^{\,i}$.

Proof. We have

$$\begin{array}{l|l}
\Sigma \vDash^{\mathsf{grp}} \mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto \overline{\kappa_i}^{\,i}\,; \overline{\Psi_i}^{\,i} & \text{Given} \\
\Omega = \Sigma & \text{Let} \\
\Omega \Vdash^{\mathsf{grp}} \mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto \overline{\kappa_i'}^{\,i}\,; \overline{\Gamma_i}^{\,i} \dashv \Theta & \text{By theorem D.51} \\
\Theta \longrightarrow \Omega' \wedge \overline{[\Omega']\kappa_i' = \kappa_i}^{\,i} \wedge \overline{[\Omega']\Gamma_i = \Psi_i}^{\,i} & \text{By theorem D.51}
\end{array}$$

We solve all unsolved kind unification variables in Θ with fresh kind parameters to get Ω_1. Then we choose $\overline{\kappa_i'' = [\Omega_1]\kappa_i'}^{\,i}$, and we prove $\Sigma \vdash \mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto^{\mathsf{p}} \overline{\kappa_i''}^{\,i}$.

$$\begin{array}{l|l}
\Theta \longrightarrow \Omega_1 & \text{By lemma D.20} \\
\Omega, \overline{\widehat{\alpha}_i}^{\,i}, \overline{T_i : \widehat{\alpha}_i}^{\,i} \longrightarrow \Omega_1 & \text{By Lemma D.19} \\
\Omega_1 = \Omega_{11}, \widehat{\alpha}_i = \kappa_{11}, \Omega_{12}, \overline{T_i : \widehat{\alpha}_i, \Omega_i'}^{\,i} & \\
\wedge \Omega \longrightarrow \Omega_{11} \wedge \Omega_{12}\ \mathbf{soft} \wedge \overline{\Omega_i'\ \mathbf{soft}}^{\,i} & \text{By Lemma D.17} \\
[\Omega_1]\Theta & \\
= [\Omega_1](\Omega, \overline{\widehat{\alpha}_i}^{\,i}, \overline{T_i : \widehat{\alpha}_i}^{\,i}) & \text{By Lemma D.30} \\
= [\Omega]\Omega, \overline{T_i : [\Omega_1]\widehat{\alpha}_i}^{\,i} & \text{By definition and Lemma D.29} \\
= \Sigma, \overline{T_i : \kappa_i''}^{\,i} & \text{By equations} \\
\Sigma \vDash^{\mathsf{grp}} \mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto \overline{\kappa_i''}^{\,i}\,; \overline{\Psi_i'}^{\,i} & \text{repeat Lemma D.43}
\end{array}$$

For any $\overline{\kappa_i}^{\,i}$ such that $\Sigma \vDash^{\mathsf{grp}} \mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto \overline{\kappa_i}^{\,i}\,; \overline{\Psi_i}^{\,i}$, by theorem D.51 we know there exists some Ω' such that $\Theta \longrightarrow \Omega'$ and $\overline{[\Omega']\kappa_i' = \kappa_i}^{\,i}$ and $\overline{[\Omega']\Gamma_i = \Psi_i}^{\,i}$. Now we construct a kind parameter substitution S. If in Θ, we have an unsolved kind unification variable $\widehat{\alpha}$, which maps to a parameter P in Ω_1. Then S maps P to $[\Omega']\widehat{\alpha}$. Because $\Theta \longrightarrow \Omega'$, then $S(\Omega_1) \longrightarrow \Omega'$ by lemma D.24. So $S(\kappa_i'') = S([\Omega_1]\kappa_i') = [S(\Omega_1)]\kappa_i'$. By lemma D.26, we have $[S(\Omega_1)]\kappa_i' = [\Omega']\kappa_i' = \kappa_i$. Similarly we have $S(\Psi_i') = \Psi_i$.

□

Theorem E.2 (Completeness of Typing Programs with Kind Parameters). *Given algorithmic contexts* Ω, Γ, *and a program* pgm, *if* $[\Omega]\Omega; [\Omega]\Gamma \vDash^{\mathsf{pgm}} pgm : \sigma$, *then* $\Omega; \Gamma \Vdash^{\mathsf{pgm}} pgm : \sigma$.

Proof. By induction on typing programs.

- Case

$$\textsc{pgm-expr}\quad \frac{\Sigma; \Psi \vdash e : \sigma}{\Sigma; \Psi \vDash^{\mathsf{pgm}} e : \sigma}$$

  Follows trivially by rule A-PGM-EXPR.

- Case

$$\textsc{pgm-dtP}\quad \frac{\Sigma \vDash^{\mathsf{grp}} \mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto \overline{\kappa_i}^{\,i}\,; \overline{\Psi_i}^{\,i} \qquad \Sigma \vdash \mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto^{\mathsf{p}} \overline{\kappa_i}^{\,i} \qquad \Sigma, \overline{T_i : S^{\star}(\kappa_i)}^{\,i}; \Psi, \overline{S^{\star}(\Psi_i)}^{\,i} \vDash^{\mathsf{pgm}} pgm : \sigma}{\Sigma; \Psi \vDash^{\mathsf{pgm}} \mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i}\,; pgm : \sigma}$$

$\Sigma \vdash^{\mathsf{grp}} \textbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto \overline{\kappa_i}^{\,i} ; \overline{\Psi_i}^{\,i}$	Given
$\Sigma \vdash \textbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto^{\mathsf{p}} \overline{\kappa_i}^{\,i}$	Given
$\Omega \Vdash^{\mathsf{grp}} \textbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \leadsto \overline{\kappa'_i}^{\,i} ; \overline{\Gamma_i}^{\,i} \dashv \Theta$	By theorem D.51
$\Theta \longrightarrow \Omega'$	Above
$\overline{[\Omega']\kappa'_i = \kappa_i}^{\,i}$	Above
$\overline{[\Omega']\Gamma_i = \Psi_i}^{\,i}$	Above

Because from theorem E.1 we know that if we solve all unsolved kind unification variables in Θ with fresh parameters to get Ω_1, then $\overline{[\Omega_1]\kappa'_i}^{\,i}$ are principal kinds. Because $\overline{\kappa_i}^{\,i}$ are principal kinds, then $\overline{[\Omega_1]\kappa'_i}^{\,i}$ and $\overline{\kappa_i}^{\,i}$ are equivalent up to renaming of type parameters. Suppose $\Theta \longrightarrow\!\!\!\!\!\rightarrow \Omega_2$, then $\overline{[\Omega_2]\kappa'_i = S^{\star}(\kappa_i)}^{\,i}$. Similarly we can prove $\overline{[\Omega_2]\Gamma_i = S^{\star}(\Psi_i)}^{\,i}$.

$[\Omega_2]\Omega_2$	
$= [\Omega,\ \overline{\widehat{\alpha}_i = [\Omega_2]\widehat{\alpha}_i}^{\,i\in 1..n},\ \overline{T_i : \widehat{\alpha}_i}^{\,i\in 1..n}]$	
$(\Omega,\ \overline{\widehat{\alpha}_i = [\Omega_2]\widehat{\alpha}_i}^{\,i\in 1..n},\ \overline{T_i : \widehat{\alpha}_i}^{\,i\in 1..n})$	By lemma D.34
$= [\Omega]\Omega,\ \overline{T_i : [\Omega]([\Omega_2]\widehat{\alpha}_i)}^{\,i\in 1..n}$	By definition
$= [\Omega]\Omega,\ \overline{T_i : S^{\star}(\kappa_i)}^{\,i\in 1..n}$	By substituting the equation
$[\Omega_2](\Gamma,\ \overline{\Gamma_i}^{\,i})$	
$= [\Omega_2]\Gamma,\ \overline{[\Omega_2]\Gamma_i}^{\,i}$	By definition
$= [\Omega]\Gamma,\ \overline{S^{\star}(\Psi_i)}^{\,i}$	By lemma D.27 and lemma D.31
$[\Omega]\Omega,\ \overline{T_i : S^{\star}(k_i)}^{\,i} ; [\Omega]\Gamma,\ \overline{S^{\star}(\Psi_i)}^{\,i} \vdash_p pgm : \sigma$	Given
$[\Omega_2]\Omega_2 ; [\Omega_2](\Gamma,\ \overline{\Gamma_i}^{\,i}) \vdash^{\mathsf{pgm}} pgm : \sigma$	By substituting the equations
$\Omega_2 ; \Gamma,\ \overline{\Gamma_i}^{\,i} \Vdash^{\mathsf{pgm}} pgm : \sigma$	I.H.
$\Omega ; \Gamma \Vdash^{\mathsf{pgm}} \textbf{rec}\,\overline{\mathcal{T}_i}^{\,i} ; pgm : \sigma$	By rule A-PGM-DT

□

F PROOF FOR POLYKINDS

F.1 List of Lemmas

F.1.1 Well-formedness of Declarative Type System.

Lemma F.1. *If* $\Sigma \vdash^{\mathsf{ela}} \mu_1 : \eta_1$, *and* $\Sigma \vdash^{\mathsf{inst}} \mu_1 : \eta_1 \sqsubseteq \eta_2 \leadsto \mu_2$, *then* $\Sigma \vdash^{\mathsf{ela}} \mu_2 : \eta_2$.

Lemma F.2. *We have:*

- *if* $\Sigma \vdash^{\mathsf{k}} \sigma : \eta \leadsto \mu$, *then* $\Sigma \vdash^{\mathsf{ela}} \mu : \eta$;
- *if* $\Sigma \vdash^{\mathsf{kc}} \sigma \Leftarrow \eta \leadsto \mu$, *then* $\Sigma \vdash^{\mathsf{ela}} \mu : \eta$.

Lemma F.3. *If* $\Sigma \vdash^{\mathsf{sig}} \mathcal{S} \leadsto T : \eta$, *then* $\Sigma \vdash^{\mathsf{ela}} \eta : \star$.

Lemma F.4. *If* $\Sigma \vdash^{\mathsf{dc}}_{p} \mathcal{D} \leadsto \mu$, *then* $\Sigma \vdash^{\mathsf{ela}} \mu : \star$.

Lemma F.5. *If* $\Sigma \vdash^{\mathsf{dt}} \mathcal{T} \leadsto \Psi$, *then* $\Sigma \vdash \Psi$.

F.1.2 Well-formedness of Algorithmic Type System.

Lemma F.6 (Well-formedness of Promotion). *If* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2$ ok*, and* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta$*, then* $\Theta = \Theta_1, \widehat{\alpha} : \omega, \Theta_2$*, and* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Theta$*, and* $\Theta_1 \Vdash^{\mathsf{ela}} \rho_2 : \omega_2$*, and* Θ ok.

Lemma F.7 (Well-formedness of Unification). *If* Δ ok*, and* $\Delta \Vdash^{\mathsf{u}} \kappa_1 \approx \kappa_2 \dashv \Theta$*, then* $\Delta \longrightarrow \Theta$*, and* Θ ok.

Lemma F.8 (Well-formedness of Instantiation). *If* $\Delta \Vdash^{\mathsf{inst}} \rho_1 : \eta_1 \sqsubseteq \eta_2 \rightsquigarrow \rho_2 \dashv \Theta$*, and* $\Delta \Vdash^{\mathsf{ela}} \rho_1 : \eta_1$*, then* $\Delta \longrightarrow \Theta$*, and* Θ ok*, and* $\Theta \Vdash^{\mathsf{ela}} \rho_2 : [\Theta]\eta_2$*.*

Lemma F.9 (Well-formedness of Unsolved). *If* Δ_1, Δ_2 ok *then* $\Delta_1, \mathsf{unsolved}(\Delta_2)$ ok.

Lemma F.10 (Well-formedness of Kinding). *Given* Δ ok,

- *if* $\Delta \Vdash^{\mathsf{k}} \sigma : \eta \rightsquigarrow \mu \dashv \Theta$*, then* $\Delta \longrightarrow \Theta$*, and* $\Theta \Vdash^{\mathsf{ela}} \mu : [\Theta]\eta$*;*
- *if* $\Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \eta \rightsquigarrow \mu \dashv \Theta$*, then* $\Delta \longrightarrow \Theta$*, and* $\Theta \Vdash^{\mathsf{ela}} \mu : [\Theta]\eta$*.*
- *if* $\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \eta) \bullet \tau : \omega \rightsquigarrow \rho_2 \dashv \Theta$*, and* $\Delta \Vdash^{\mathsf{ela}} \rho_1 : \eta$*, then* $\Delta \longrightarrow \Theta$*, and* $\Theta \Vdash^{\mathsf{ela}} \rho_2 : [\Theta]\omega$*.*

Lemma F.11 (Well-formedness of Elaborated Kinding). *If* Δ ok*, and* $\Delta \Vdash^{\mathsf{ela}} \mu : \eta$*, then* $\Delta \Vdash^{\mathsf{ela}} \eta : \star$*, and* $[\Delta]\eta = \eta$*.*

Lemma F.12 (Well-formedness of Typing Data Constructor Decl.). *If* Δ ok*, and* $\Delta \Vdash^{\mathsf{dc}}_{\rho} \mathcal{D} \rightsquigarrow \mu \dashv \Theta$*, then* $\Delta \longrightarrow \Theta$*, and* $\Theta \Vdash^{\mathsf{ela}} \mu : \star$*.*

Lemma F.13 (Well-formedness of Typing Datatype Declaration). *If* Δ ok*, and* $\Delta \Vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow \Gamma \dashv \Theta$*, then* $\Delta \longrightarrow \Theta$*, and* $\Theta \Vdash^{\mathsf{ectx}} \Gamma$*.*

F.1.3 Properties of Context Extension.

Lemma F.14 (Declaration Preservation). *If* $\Delta \longrightarrow \Theta$*, if a type constructor or a type variable or a kind unification variable is declared in* Δ*, then it is declared in* Θ*.*

Lemma F.15 (Extension Weakening). *Given* $\Delta \longrightarrow \Theta$*, if* $\Delta \Vdash^{\mathsf{ela}} \mu : \eta$*, then* $\Theta \Vdash^{\mathsf{ela}} \mu : [\Theta]\eta$*.*

Definition F.16 (Contextual Size).

$$
\begin{array}{lcl}
\mid \Delta \vdash \star \mid & = & 1 \\
\mid \Delta \vdash a \mid & = & 1 \\
\mid \Delta \vdash \mathsf{Int} \mid & = & 1 \\
\mid \Delta \vdash T \mid & = & 1 \\
\mid \Delta \vdash \rightarrow \mid & = & 1 \\
\mid \Delta \vdash \omega_1\, \omega_2 \mid & = & 1 + \mid \Delta \vdash \omega_1 \mid + \mid \Delta \vdash \omega_2 \mid \\
\mid \Delta \vdash \omega_1\ @\omega_2 \mid & = & 1 + \mid \Delta \vdash \omega_1 \mid + \mid \Delta \vdash \omega_2 \mid \\
\mid \Delta[\widehat{\alpha} : \omega] \vdash \widehat{\alpha} \mid & = & 1 \\
\mid \Delta[\widehat{\alpha} : \omega = \rho] \vdash \widehat{\alpha} \mid & = & 1 + \mid \Delta[\widehat{\alpha} : \omega = \rho] \vdash \omega \mid \\
\mid \Delta \vdash \forall a : \rho.\omega \mid & = & 1 + \mid \Delta \vdash \rho \mid + \mid \Delta \vdash \omega \mid \\
\mid \Delta \vdash \forall \{a : \rho\}.\omega \mid & = & 1 + \mid \Delta \vdash \rho \mid + \mid \Delta \vdash \omega \mid
\end{array}
$$

Lemma F.17 (Substitution Kinding). *If* Δ ok*, and* $\Delta \Vdash^{\mathsf{ela}} \mu : \eta$*, then* $\Delta \Vdash^{\mathsf{ela}} [\Delta]\mu : \eta$*.*

Lemma F.18 (Soft Substitution Kinding). *If* Δ_1, Δ_2 ok, *and* Δ_2 **soft**, *and* $\Delta_1, \Delta_2 \Vdash^{\mathsf{ela}} \mu : \eta$, *then* $\Delta_1, \mathsf{unsolved}(\Delta_2) \Vdash^{\mathsf{ela}} [\Delta_2]\mu : \eta$.

Lemma F.19 (Reflexivity of Context Extension). *If* Δ ok, *then* $\Delta \longrightarrow \Delta$.

Lemma F.20 (Well-formedness of Context Extension). *If* Δ ok, *and* $\Delta \longrightarrow \Theta$, *then* Θ ok.

Definition F.21 (Softness). *A context* Δ *is soft iff it contains only of* $\widehat{\alpha}$ *and* $\widehat{\alpha} = \kappa$ *declarations, including local scopes.*

Lemma F.22 (Extension Order).

(1) *If* $\Delta_1, a : \omega, \Delta_2 \longrightarrow \Theta$, *then* $\Theta = \Theta_1, a : \omega, \Theta_2$, *where* $\Delta_1 \longrightarrow \Theta_1$. *Moreover, if* Δ_2 **soft**, *then* Θ_2 **soft**.
(2) *If* $\Delta_1, T : \eta, \Delta_2 \longrightarrow \Theta$, *then* $\Theta = \Theta_1, T : \eta, \Theta_2$, *where* $\Delta_1 \longrightarrow \Theta_1$. *Moreover, if* Δ_2 **soft**, *then* Θ_2 **soft**.
(3) *If* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Theta$, *then* $\Theta = \Theta_1, \Theta', \Theta_2$, *where* $\Delta_1 \longrightarrow \Theta_1$, *and* Θ' *is either* $\widehat{\alpha} : \omega$ *or* $\widehat{\alpha} : \omega = \rho$ *for some* ρ. *Moreover, if* Δ_2 **soft**, *then* Θ_2 **soft**.
(4) *If* $\Delta_1, \widehat{\alpha} : \omega = \rho_1, \Delta_2 \longrightarrow \Theta$, *then* $\Theta = \Theta_1, \widehat{\alpha} : \omega = \rho_2, \Theta_2$, *where* $\Delta_1 \longrightarrow \Theta_1$, *and* $[\Theta_1]\rho_1 = [\Theta_1]\rho_2$. *Moreover, if* Δ_2 **soft**, *then* Θ_2 **soft**.
(5) *If* $\Delta_1, \{\Delta\}, \Delta_2 \longrightarrow \Theta$, *then* $\Theta = \Theta_1, \{\Theta\}, \Theta_2$, *where* $\Delta_1 \longrightarrow \Theta_1$, *Moreover, if* Δ_2 **soft**, *then* Θ_2 **soft**.

Lemma F.23 (Substitution Extension Invariance). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{ela}} \mu : \eta$, *and* $\Delta \longrightarrow \Theta$, *then* $[\Theta]\kappa = [\Theta]([\Delta]\mu)$ *and* $[\Theta]\kappa = [\Delta]([\Theta]\mu)$.

Lemma F.24 (Transitivity of Context Extension). *If* Δ' ok, *and* $\Delta' \longrightarrow \Delta$, *and* $\Delta \longrightarrow \Theta$, *then* $\Delta' \longrightarrow \Theta$.

Lemma F.25 (Solution Admissibility for Extension).

- *If* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2$ ok *and* $\Delta_1 \Vdash^{\mathsf{ela}} \rho : [\Delta_1]\omega$, *then* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha} : \omega = \rho, \Delta_2$.
- *If* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega, \Delta_4\}, \Delta_2$ ok *and* $\Delta_1, \Delta_3 \Vdash^{\mathsf{ela}} \rho : [\Delta_1, \Delta_3]\omega$, *then* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega, \Delta_4\}, \Delta_2 \longrightarrow \Delta_1, \{\Delta_3, \widehat{\alpha} : \omega = \rho, \Delta_4\}, \Delta_2$.

Lemma F.26 (Solved Variable Addition for Extension).

- *If* Δ_1, Δ_2 ok *and* $\Delta_1 \Vdash^{\mathsf{ela}} \rho : [\Delta_1]\omega$, *then* $\Delta_1, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha} : \omega = \rho, \Delta_2$.
- *If* $\Delta_1, \{\Delta_2, \Delta_3\}, \Delta_4$ ok *and* $\Delta_1, \Delta_2 \Vdash^{\mathsf{ela}} \rho : [\Delta_1, \Delta_2]\omega$, *then* $\Delta_1, \{\Delta_2, \Delta_3\}, \Delta_4 \longrightarrow \Delta_1, \{\Delta_2, \widehat{\alpha} : \omega = \rho, \Delta_3\}, \Delta_4$.

Lemma F.27 (Unsolved Variable Addition).

- *If* Δ_1, Δ_2 ok *and* $\Delta_1 \Vdash^{\mathsf{ela}} \omega : \star$ *then* $\Delta_1, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha} : \omega, \Delta_2$.
- *If* $\Delta_1, \{\Delta_2, \Delta_3\}, \Delta_4$ ok *and* $\Delta_1, \Delta_2 \Vdash^{\mathsf{ela}} \omega : \star$, *then* $\Delta_1, \{\Delta_2, \Delta_3\}, \Delta_4 \longrightarrow \Delta_1, \{\Delta_2, \widehat{\alpha} : \omega, \Delta_3\}, \Delta_4$.

Lemma F.28 (Parallel Admissibility).

- *If* $\Delta_1 \longrightarrow \Theta_1$, *and* Δ_1, Δ_2 ok, *and* $\Delta_1, \Delta_2 \longrightarrow \Theta_1, \Theta_2$, *and* Δ_2 *is fresh w.r.t.* Θ_1, *then:*
  - *if* $\Delta_1 \Vdash^{\mathsf{ela}} \omega : \star$, *then* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega, \Theta_2$;
  - *if* $\Theta_1 \Vdash^{\mathsf{ela}} \rho : [\Theta_1]\omega$, *then* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \rho, \Theta_2$;
  - *if* $[\Theta_1]\rho_1 = [\Theta_1]\rho_2$, *then* $\Delta_1, \widehat{\alpha} : \omega = \rho_1, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \rho_2, \Theta_2$.
- *If* $\Delta_1, \{\Delta_3\} \longrightarrow \Theta_1, \{\Theta_3\}$, *and* $\Delta_1, \{\Delta_3, \Delta_4\}, \Delta_2$ ok, *and* $\Delta_1, \{\Delta_3, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \Theta_4\}, \Theta_2$, *and* Δ_2, Δ_4 *is fresh w.r.t.* Θ_1, Θ_3, *then:*

- *if* $\Delta_1, \{\Delta_3\} \Vdash^{\mathsf{ela}} \omega : \star$, *then* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega, \Theta_4\}, \Theta_2$;
- *if* $\Theta_1, \{\Theta_3\} \Vdash^{\mathsf{ela}} \rho : [\Theta_1, \Theta_3]\omega$, *then* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega = \rho, \Theta_4\}, \Theta_2$;
- *if* $[\Theta_1, \Theta_3]\rho_1 = [\Theta_1, \Theta_3]\rho_2$, *then* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega = \rho_1, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega = \rho_2, \Theta_4\}, \Theta_2$.

Lemma F.29 (Parallel Extension Solution).

- *If* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \rho_2, \Theta_2$, *and* $[\Theta_1]\rho_1 = [\Theta_1]\rho_2$, *then* $\Delta_1, \widehat{\alpha} : \omega = \rho_1, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \kappa_2, \Theta_2$.
- *If* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega = \rho_2, \Theta_4\}, \Theta_2$, *and* $[\Theta_1, \Theta_3]\rho_1 = [\Theta_1, \Theta_3]\rho_2$, *then* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega = \rho_1, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega = \rho_2, \Theta_4\}, \Theta_2$.

Lemma F.30 (Parallel Variable Update).

- *If* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \rho, \Theta_2$, *and* $\Delta_1 \Vdash^{\mathsf{ela}} \rho_1 : [\Delta_1]\omega$, *and* $\Theta_1 \Vdash^{\mathsf{ela}} \rho_2 : [\Theta_1]\omega$, *and* $[\Theta_1]\rho = [\Theta_1]\rho_1 = [\Theta_1]\rho$, *then* $\Delta_1, \widehat{\alpha} : \omega = \rho_1, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \rho_2, \Theta_2$
- *If* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega = \rho, \Theta_4\}, \Theta_2$, *and* $\Delta_1, \Delta_3 \Vdash^{\mathsf{ela}} \rho_1 : [\Delta_1, \Delta_3]\omega$, *and* $\Theta_1, \Theta_3 \Vdash^{\mathsf{ela}} \rho_2 : [\Theta_1, \Theta_3]\omega$, *and* $[\Theta_1]\rho = [\Theta_1]\rho_1 = [\Theta_1]\rho$, *then* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega = \rho_1, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega = \rho_2, \Theta_4\}, \Theta_2$

F.1.4 Properties of Complete Context.

Lemma F.31 (Finishing Kinding). *If* Ω ok, *and* $\Omega \Vdash^{\mathsf{ela}} \rho : \omega$, *and* $\Omega \longrightarrow \Omega'$, *then* $[\Omega]\rho = [\Omega']\rho$.

Lemma F.32 (Finishing Term Contexts). *If* Ω ok, *and* $\Omega \Vdash^{\mathsf{ectx}} \Gamma$, *and* $\Omega \longrightarrow \Omega'$, *then* $[\Omega']\Gamma = [\Omega]\Gamma$.

Lemma F.33 (Stability of Complete Contexts). *If* $\Delta \longrightarrow \Omega$, *then* $[\Omega]\Delta = [\Omega]\Omega$.

Lemma F.34 (Softness Goes Away). *If* $\Delta_1, \Delta_2 \longrightarrow \Omega_1, \Omega_2$ *where* $\Delta_1 \longrightarrow \Omega_1$, *and* Δ_2 **soft**, *then* $[\Omega_1, \Omega_2](\Delta_1, \Delta_2) = [\Omega_1]\Delta_1$.

Lemma F.35 (Confluence of Completeness). *If* $\Delta_1 \longrightarrow \Omega$, *and* $\Delta_2 \longrightarrow \Omega$, *then* $[\Omega]\Delta_1 = [\Omega]\Delta_2$.

Lemma F.36 (Substitution Stability). *If* Ω, Δ ok, *and* $\Omega \Vdash^{\mathsf{ela}} \rho : \omega$, *then* $[\Omega]\rho = [\Omega, \Delta]\rho$.

Lemma F.37 (Type Constructor Preservation). *If* Δ ok, *then* $(T : \eta) \in \Delta$, *and* $\Delta \longrightarrow \Omega$, *then* $(T : [\Omega]\eta) \in [\Omega]\Delta$.

Lemma F.38 (Type Variable Preservation). *If* $(a : \omega) \in \Delta$, *and* $\Delta \longrightarrow \Omega$, *then* $(a : [\Omega]\omega) \in [\Omega]\Delta$.

Lemma F.39 (Finishing Completions). *If* Ω ok, *and* $\Omega \longrightarrow \Omega'$, *then* $[\Omega']\Omega'$ *is a well-formed permutation of* $[\Omega]\Omega$.

F.1.5 Decidability.

Lemma F.40 (Promotion Preserves $\langle\Delta\rangle$). *If* $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \omega_2 \dashv \Theta$, *then* $\langle\Delta\rangle = \langle\Theta\rangle$.

Lemma F.41 (Unification Makes Progress). *If* $\Delta \Vdash^{\mathsf{u}} \omega_1 \approx \omega_2 \dashv \Theta$, *then either* $\Theta = \Delta$, *or* $\langle\Theta\rangle < \langle\Delta\rangle$.

Lemma F.42 (Promotion Preserves $|\rho|$). *Given a context* $\Delta[\widehat{\alpha}]$ ok, *if* $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \omega_2 \dashv \Theta$, *then for all* ρ, *we have* $|[\Delta]\rho| = |[\Theta]\rho|$.

Theorem F.43 (Promotion is Decidable). *Given a context* $\Delta[\widehat{\alpha}]$ ok, *and a kind* ρ_1 *with* $[\Delta]\rho_1 = \rho_1$, *it is decidable whether there exists* Θ *such that* $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \omega_2 \dashv \Theta$.

Theorem F.44 (Unification is Decidable). *Given a context* Δ ok, *and kinds* ρ_1 *and* ρ_2, *where* $[\Delta]\rho_1 = \rho_1$, *and* $[\Delta]\rho_2 = \rho_2$, *it is decidable whether there exists* Θ *such that* $\Delta \Vdash^{\mathsf{u}} \rho_1 \approx \rho_2 \dashv \Theta$.

F.1.6 Source of Unification Variables.

Lemma F.45 (Source of Unification Variables). *If* $\Delta \Vdash^{\mathsf{k}} \sigma : \eta \leadsto \mu \dashv \Theta$, *then for any* $\widehat{\alpha} \in \mathsf{unsolved}(\Theta)$, *either* $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\mu)$, *or there exists* $\widehat{\beta} \in \mathsf{unsolved}(\Delta)$ *such that* $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\widehat{\beta})$.

F.1.7 Soundness of Algorithm.

Lemma F.46 (Soundness of Promotion). *If* Δ ok, *and* $[\Delta]\omega_1 = \omega_1$, *and* $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \omega_2 \dashv \Theta$, *then* $[\Theta]\omega_1 = [\Theta]\omega_2 = \omega_2$. *If* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\omega_1 = [\Omega]\omega_2$.

Lemma F.47 (Soundness of Unification). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{u}} \omega_1 \approx \omega_2 \dashv \Theta$, *then* $[\Theta]\omega_1 = [\Theta]\omega_2$. *If* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\omega_1 = [\Omega]\omega_2$.

Lemma F.48 (Soundness of Instantiation). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{ela}} \mu_1 : \eta$, *and* $\Delta \Vdash^{\mathsf{ela}} \omega : \star$, *and* $\Delta \Vdash^{\mathsf{inst}} \mu_1 : \eta \sqsubseteq \omega \leadsto \mu_2 \dashv \Theta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{\mathsf{inst}} [\Omega]\mu_1 : [\Omega]\eta \sqsubseteq [\Omega]\omega \leadsto [\Omega]\mu_2$.

Lemma F.49 (Soundness of Kinding). *If* Δ ok, *we have*

- *if* $\Delta \Vdash^{\mathsf{k}} \sigma : \eta \leadsto \mu \dashv \Theta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{\mathsf{k}} \sigma : [\Omega]\eta \leadsto [\Omega]\mu$;
- *if* $\Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \eta \leadsto \mu \dashv \Theta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{\mathsf{kc}} \sigma \Leftarrow [\Omega]\eta \leadsto [\Omega]\mu$.
- *if* $\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \eta) \bullet \tau : \omega \leadsto \rho_2 \dashv \Theta$, *and* $\Delta \Vdash^{\mathsf{ela}} \rho_1 : \eta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{\mathsf{inst}} [\Omega]\rho_1 : [\Omega]\eta \sqsubseteq (\omega_1 \ \rightarrow \ [\Omega]\omega) \leadsto \rho_3$, *and* $[\Omega]\Delta \vdash^{\mathsf{kc}} \tau \Leftarrow \omega_1 \leadsto \rho_4$. *and* $[\Omega]\rho_2 = \rho_3\,\rho_4$.

Lemma F.50 (Soundness of Elaborated Kinding). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{ela}} \mu : \eta$, *and* $\Delta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{\mathsf{ela}} [\Omega]\mu : [\Omega]\eta$.

Lemma F.51 (Soundness of Typing Signature). *If* Δ ok, *and* $\Omega \Vdash^{\mathsf{sig}} \mathcal{S} \leadsto T : \eta$, *then* $[\Omega]\Omega \vdash^{\mathsf{sig}} \mathcal{S} \leadsto T : \eta$.

Lemma F.52 (Soundness of Typing Data Constructor Decl.). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{dc}}_{\rho} \mathcal{D} \leadsto \mu \dashv \Theta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{\mathsf{dc}}_{([\Omega]\rho)} \mathcal{D} \leadsto [\Omega]\mu$.

Lemma F.53 (Soundness of Typing Datatype Decl.). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{dt}} \mathcal{T} \leadsto \Gamma \dashv \Theta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{\mathsf{dt}} \mathcal{T} \leadsto [\Omega]\Gamma$.

Lemma F.54 (Soundness of Typing Program). *If* $\Omega; \Gamma \Vdash^{\mathsf{pgm}} pgm : \mu$, *then* $[\Omega]\Omega; [\Omega]\Gamma \vdash^{\mathsf{pgm}} pgm : [\Omega]\mu$.

F.1.8 Principality.

Lemma F.55 (Completeness of Promotion). *Given* Δ ok, *and* $\Delta \longrightarrow \Omega$, *and* $\widehat{\alpha} \in \Delta$, *and* $\Delta \Vdash^{\mathsf{ela}} \rho : \omega$, *and* $[\Delta]\widehat{\alpha} = \widehat{\alpha}$, *and* $[\Delta]\rho = \rho$, *if* κ *is free of* $\widehat{\alpha}$, *then there exists* ρ_2, Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$, *and* $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho \rightsquigarrow \rho_2 \dashv \Theta$.

Lemma F.56 (Completeness of Unification). *Given* Δ ok, *and* $\Delta \longrightarrow \Omega$, *and* $\Delta \Vdash^{\mathsf{ela}} \rho_1 : \omega$ *and* $\Delta \Vdash^{\mathsf{ela}} \rho_2 : \omega$, *and* $[\Delta]\rho_1 = \rho_1$ *and* $[\Delta]\rho_2 = \rho_2$, *if* $[\Omega]\rho_1 = [\Omega]\rho_2$, *then there exists* Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$ *and* $\Delta \Vdash^{\mathsf{u}} \rho_1 \approx \rho_2 \dashv \Theta$.

Lemma F.57 (Completeness of Instantiation). *Given* $\Delta \longrightarrow \Omega$, *and* $\Delta \Vdash^{\mathsf{ela}} \rho : \eta$ *and* $\Delta \Vdash^{\mathsf{ela}} \omega : \star$, *and* $[\Delta]\eta = \eta$ *and* $[\Delta]\omega = \omega$, *if* $[\Omega]\Delta \vdash^{\mathsf{inst}} [\Omega]\rho_1 : [\Omega]\eta \sqsubseteq [\Omega]\omega \leadsto \rho_2$, *then there exists* ρ_2', Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$ *and* $\Delta \Vdash^{\mathsf{inst}} \rho_1 : \eta \sqsubseteq \omega \leadsto \rho_2' \dashv \Theta$, *and* $[\Omega']\rho_2' = \rho_2$.

Lemma F.58 (Principality of Kinding).

- *Given* $\Delta \longrightarrow \Omega$, *if* $[\Omega]\Delta \vdash^{\mathsf{k}} \sigma : \eta \rightsquigarrow \mu$, *and* $\Delta \Vdash^{\mathsf{k}} \sigma : \eta' \rightsquigarrow \mu' \dashv \Theta$, *then there exists* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$. *Moreover,* $[\Omega']\eta' = \eta$. *Furthermore, if* μ *and* μ' *are monotypes, then* $[\Omega']\mu' = \mu$.
- *Given* $\Delta \longrightarrow \Omega$, *if* $[\Omega]\Delta \vdash^{\mathsf{kc}} \sigma \Leftarrow [\Omega]\eta \rightsquigarrow \mu$, *and* $\Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \eta \rightsquigarrow \mu' \dashv \Theta$, *then there exists* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$. *Furthermore, if* μ *and* μ' *are monotypes, then* $[\Omega']\mu' = \mu$.
- *Given* $\Delta \longrightarrow \Omega$, *if* $[\Omega]\Delta \vdash^{\mathsf{inst}} [\Omega]\rho_1 : [\Omega]\eta \sqsubseteq (\omega_1 \ \rightarrow \ \omega_2) \rightsquigarrow \rho_3$, *and* $[\Omega]\Delta \vdash^{\mathsf{kc}} \tau \Leftarrow \omega_1 \rightsquigarrow \rho_4$ *and* $\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \eta) \bullet \tau : \omega \rightsquigarrow \rho_2 \dashv \Theta$, *then there exists* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$. *Moreover,* $[\Omega']\omega = \omega_2$. *Further,* $[\Omega']\rho_2 = \rho_3\,\rho_4$.

Lemma F.59 (Principality of Typing Data Constructor Declaration). *Given* $\Delta \longrightarrow \Omega$, *if* $[\Omega]\Delta \vdash^{\mathsf{dc}}_{\rho} \mathcal{D} \rightsquigarrow \mu_1$, *and* $\Delta \Vdash^{\mathsf{dc}}_{\rho} \mathcal{D} \rightsquigarrow \mu_2 \dashv \Theta$, *then there exists* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$.

Lemma F.60 (Principality of Typing Datatype Declaration). *Given* $\Delta \longrightarrow \Omega$, *if* $[\Omega]\Delta \vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow \Psi$, *and* $\Delta \Vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow \Gamma \dashv \Theta$, *then there exists* Ω' *such that* $\Theta \longrightarrow \Omega'$, *and* $\Omega \longrightarrow \Omega'$.

Theorem F.61 (Principality of Typing a Datatype Declaration Group). *If* $\Omega \Vdash^{\mathsf{grp}} \ \mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \ \rightsquigarrow \ \overline{\eta_i}^{\,i}; \overline{\Gamma_i}^{\,i}$, *then whenever* $[\Omega]\Omega \vdash^{\mathsf{grp}} \ \mathbf{rec}\,\overline{\mathcal{T}_i}^{\,i} \ \rightsquigarrow \ \overline{\eta_i'}^{\,i}; \overline{\Psi_i}^{\,i}$ *holds, we have* $[\Omega]\Omega \vdash [\Omega]\eta_i \leq \eta_i'$.

F.2 Proofs

Lemma F.1. *If* $\Sigma \vdash^{\mathsf{ela}} \mu_1 : \eta_1$, *and* $\Sigma \vdash^{\mathsf{inst}} \mu_1 : \eta_1 \sqsubseteq \eta_2 \rightsquigarrow \mu_2$, *then* $\Sigma \vdash^{\mathsf{ela}} \mu_2 : \eta_2$.

Proof. By induction on the derivation.

- Case

$$\frac{}{\Sigma \vdash^{\mathsf{inst}} \mu : \omega \sqsubseteq \omega \rightsquigarrow \mu}\ \textsc{inst-refl}$$

The goal follows trivially.

- Case

$$\frac{\Sigma \vdash^{\mathsf{ela}} \rho : \omega_1 \qquad \Sigma \vdash^{\mathsf{inst}} \mu_1 \mathbin{@}\rho : \eta[a \mapsto \rho] \sqsubseteq \omega_2 \rightsquigarrow \mu_2}{\Sigma \vdash^{\mathsf{inst}} \mu_1 : \forall a : \omega_1 . \eta \sqsubseteq \omega_2 \rightsquigarrow \mu_2}\ \textsc{inst-forall}$$

Given $\Sigma \vdash^{\mathsf{ela}} \mu_1 : \forall a : \omega . \eta_1$, and $\Sigma \vdash^{\mathsf{ela}} \rho : \omega$, we know by rule ELA-KAPP that $\Sigma \vdash^{\mathsf{ela}} \mu_1 \mathbin{@}\rho : \eta_1[a \mapsto \rho]$. So the goal follows by I.H..

- Case

$$\frac{\Sigma \vdash^{\mathsf{ela}} \rho : \omega_1 \qquad \Sigma \vdash^{\mathsf{inst}} \mu_1 \mathbin{@}\rho : \eta[a \mapsto \rho] \sqsubseteq \omega_2 \rightsquigarrow \mu_2}{\Sigma \vdash^{\mathsf{inst}} \mu_1 : \forall \{a : \omega_1\} . \eta \sqsubseteq \omega_2 \rightsquigarrow \mu_2}\ \textsc{inst-forall-infer}$$

Similar as the previous case.

□

Lemma F.2. *We have:*

- *if* $\Sigma \vdash^{\mathsf{k}} \sigma : \eta \rightsquigarrow \mu$, *then* $\Sigma \vdash^{\mathsf{ela}} \mu : \eta$;
- *if* $\Sigma \vdash^{\mathsf{kc}} \sigma \Leftarrow \eta \rightsquigarrow \mu$, *then* $\Sigma \vdash^{\mathsf{ela}} \mu : \eta$.

Proof. By induction on the derivation.

Part 1 • Case rules KTT-STAR, KTT-NAT, KTT-VAR, KTT-TCON, and KTT-ARROW follow directly from rules ELA-STAR, ELA-NAT, ELA-VAR, ELA-TCON, and ELA-ARROW respectively.

- Case

$$\textsc{ktt-app}\quad \frac{\Sigma \vdash^{\mathsf{k}} \tau_1 : \eta_1 \rightsquigarrow \rho_1 \qquad \Sigma \vdash^{\mathsf{inst}} \rho_1 : \eta_1 \sqsubseteq (\omega_1 \ \rightarrow\ \omega_2) \rightsquigarrow \rho_2 \qquad \Sigma \vdash^{\mathsf{kc}} \tau_2 \Leftarrow \omega_1 \rightsquigarrow \rho_3}{\Sigma \vdash^{\mathsf{k}} \tau_1\, \tau_2 : \omega_2 \rightsquigarrow \rho_2\, \rho_3}$$

$$\begin{array}{l|l}
\Sigma \vdash^{\mathsf{ela}} \rho_1 : \eta & \text{I.H.} \\
\Sigma \vdash^{\mathsf{ela}} \rho_2 : \omega_1 \ \rightarrow\ \omega_2 & \text{By lemma F.1} \\
\Sigma \vdash^{\mathsf{ela}} \rho_3 : \omega_1 & \text{By part 2} \\
\Sigma \vdash^{\mathsf{ela}} \rho_1\, \rho_3 : \omega_2 & \text{By rule } \textsc{ela-app}
\end{array}$$

- The rest cases follow directly from I.H. and part 2.

Part 2

$$\textsc{kc-sub}\quad \frac{\Sigma \vdash^{\mathsf{k}} \sigma : \eta \rightsquigarrow \mu_1 \qquad \Sigma \vdash^{\mathsf{inst}} \mu_1 : \eta \sqsubseteq \omega \rightsquigarrow \mu_2}{\Sigma \vdash^{\mathsf{kc}} \sigma \Leftarrow \omega \rightsquigarrow \mu_2}$$

$$\begin{array}{l|l}
\Sigma \vdash^{\mathsf{ela}} \mu_1 : \eta_1 & \text{I.H.} \\
\Sigma \vdash^{\mathsf{ela}} \mu_2 : \eta_2 & \text{By lemma F.1}
\end{array}$$

□

Lemma F.3. *If* $\Sigma \vdash^{\mathsf{sig}} \mathcal{S} \rightsquigarrow T : \eta$, *then* $\Sigma \vdash^{\mathsf{ela}} \eta : \star$.

Proof. We have

$$\textsc{sig-tt}\quad \frac{\rceil\sigma\lceil \qquad \phi \in \mathcal{Q}(\sigma) \qquad \phi^{\mathsf{c}} \in \mathcal{Q}(\eta) \qquad \Sigma, \phi^{\mathsf{c}} \vdash^{\mathsf{k}} \forall\{\phi\}.\sigma : \star \rightsquigarrow \eta}{\Sigma \vdash^{\mathsf{sig}} \mathbf{data}\ T : \sigma \rightsquigarrow T : \forall\{\phi^{\mathsf{c}}\}.\eta}$$

$$\begin{array}{l|l}
\Sigma, \phi^{\mathsf{c}} \vdash^{\mathsf{k}} \forall\{\phi\}.K : \star \rightsquigarrow \eta & \text{Given} \\
\Sigma, \phi^{\mathsf{c}} \vdash^{\mathsf{ela}} \eta : \star & \text{By lemma F.2} \\
\phi^{\mathsf{c}} \text{ is well-formed} & \\
\Sigma \vdash^{\mathsf{ela}} \forall\{\phi^{\mathsf{c}}\}.\eta : \star & \text{By rule } \textsc{ela-forall-infer}
\end{array}$$

□

Lemma F.4. *If* $\Sigma \vdash^{\mathsf{dc}}_{\rho} \mathcal{D} \rightsquigarrow \mu$, *then* $\Sigma \vdash^{\mathsf{ela}} \mu : \star$.

Proof. We have

$$\textsc{dc-tt}\quad \frac{\phi^{\mathsf{c}} \in \mathcal{Q}(\mu\backslash_{\Sigma, \overline{\tau_i}^{\,i}}) \qquad \Sigma, \phi^{\mathsf{c}} \vdash^{\mathsf{k}} \forall\phi.\overline{\tau_i}^{\,i} \ \rightarrow \rho : \star \rightsquigarrow \mu}{\Sigma \vdash^{\mathsf{dc}}_{\rho} \forall\phi.D\, \overline{\tau_i}^{\,i} \rightsquigarrow \forall\{\phi^{\mathsf{c}}\}.\mu}$$

$$\begin{array}{l|l}
\Sigma, \phi^{\mathsf{c}} \vdash^{\mathsf{k}} \forall\phi.\overline{\tau_i}^{\,i} \ \rightarrow \rho : \star \rightsquigarrow \mu & \text{Given} \\
\Sigma, \phi^{\mathsf{c}} \vdash^{\mathsf{ela}} \mu : \star & \text{By lemma F.2} \\
\phi^{\mathsf{c}} \text{ is well-formed} & \\
\Sigma \vdash^{\mathsf{ela}} \forall\{\phi^{\mathsf{c}}\}.\mu : \star & \text{By rule } \textsc{ela-forall-infer}
\end{array}$$

□

Lemma F.5. *If* $\Sigma \vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow \Psi$, *then* $\Sigma \vdash \Psi$.

Proof. We have

$$\frac{(T : \forall\{\phi^{\mathrm{c}}_1\}.\forall\phi^{\mathrm{c}}_2.\ \overline{\omega_i}^{\,i} \rightarrow \star) \in \Sigma \qquad \overline{\Sigma, \phi^{\mathrm{c}}_1, \phi^{\mathrm{c}}_2,\ \overline{a_i : \omega_i}^{\,i} \vdash^{\mathrm{dc}}_{(T\ @\phi^{\mathrm{c}}_1\ @\phi^{\mathrm{c}}_2\ \overline{a_i}^{\,i})} \mathcal{D}_j \rightsquigarrow \mu_j}^{\,j}}{\Sigma \vdash^{\mathrm{dt}} \mathbf{data}\ T\, \overline{a_i}^{\,i} = \overline{\mathcal{D}_j}^{\,j} \rightsquigarrow \overline{D_j : \forall\{\phi^{\mathrm{c}}_1\}.\forall\phi^{\mathrm{c}}_2.\ \forall\overline{a_i : \omega_i}^{\,i}.\mu_j}^{\,j}}\ \textsc{dt-tt}$$

$$\begin{array}{l|l}
\overline{\Sigma, \phi^{\mathrm{c}}_1, \phi^{\mathrm{c}}_2,\ \overline{a_i : \omega_i}^{\,i} \vdash^{\mathrm{ela}} \mu_j : \star}^{\,j} & \text{By lemma F.4} \\
\overline{\Sigma \vdash^{\mathrm{ela}} \forall\{\phi^{\mathrm{c}}_1\}.\forall\phi^{\mathrm{c}}_2.\ \forall\overline{a_i : \omega_i}^{\,i}.\mu_j : \star}^{\,j} & \text{By rule } \textsc{ela-forall} \\
\Sigma \vdash \overline{D_j : \forall\{\phi^{\mathrm{c}}_1\}.\forall\phi^{\mathrm{c}}_2.\ \forall\overline{a_i : \omega_i}^{\,i}.\mu_j}^{\,j} & \text{By rule } \textsc{ectx-dcon}
\end{array}$$

□

F.2.1 *Well-formedness of Algorithmic Type System.*

Lemma F.6 (Well-formedness of Promotion). *If* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2$ ok, *and* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \Vdash^{\mathrm{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta$, *then* $\Theta = \Theta_1, \widehat{\alpha} : \omega, \Theta_2$, *and* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Theta$, *and* $\Theta_1 \Vdash^{\mathrm{ela}} \rho_2 : \omega_2$, *and* Θ ok.

Proof. For most cases, the goal follows from F.19.

The only interesting case is for

$$\frac{\Delta \Vdash^{\mathrm{pr}}_{\widehat{\alpha}} [\Delta]\rho \rightsquigarrow \rho_1 \dashv \Theta[\widehat{\alpha}][\widehat{\beta} : \rho]}{\Delta[\widehat{\alpha}][\widehat{\beta} : \rho] \Vdash^{\mathrm{pr}}_{\widehat{\alpha}} \widehat{\beta} \rightsquigarrow \widehat{\beta_1} \dashv \Theta[\widehat{\beta_1} : \rho_1, \widehat{\alpha}][\widehat{\beta} : \rho = \widehat{\beta_1}]}\ \textsc{a-pr-kuvarR-tt}$$

Suppose $\Theta[\widehat{\beta_1} : \rho_1, \widehat{\alpha}][\widehat{\beta} : \rho = \widehat{\beta_1}] = \Theta_1, \widehat{\beta_1} : \rho_1, \widehat{\alpha} : \rho_2, \Theta_2, \widehat{\beta} : \rho = \widehat{\beta_1}, \Theta_3$. Then we are given that $\Theta_1 \Vdash^{\mathrm{ela}} \widehat{\beta_1} : [\Theta_1]\rho_1$. By lemma F.46, we know that $\Theta_1 \Vdash^{\mathrm{ela}} \widehat{\beta_1} : \rho_1$. By lemma F.11, we have $\Theta_1 \Vdash^{\mathrm{ela}} \rho_1 : \star$.

So by I.H., we have $\Delta \longrightarrow \Theta[\widehat{\alpha}][\widehat{\beta} : \rho]$. By lemma F.27 and lemma F.24, we know $\Delta \longrightarrow \Theta[\widehat{\beta_1} : \rho_1, \widehat{\alpha}][\widehat{\beta} : \rho]$. Then by lemma F.25 and lemma F.24, we have $\Delta \longrightarrow \Theta[\widehat{\beta_1} : \rho_1, \widehat{\alpha}][\widehat{\beta} : \rho = \widehat{\beta_1}]$.

□

Lemma F.7 (Well-formedness of Unification). *If* Δ ok, *and* $\Delta \Vdash^{\mathrm{u}} \kappa_1 \approx \kappa_2 \dashv \Theta$, *then* $\Delta \longrightarrow \Theta$, *and* Θ ok.

Proof. By induction on the derivation.

- The case for rule $\textsc{a-u-refl-tt}$ follows directly from lemma F.19.
- Case

$$\frac{\Delta \Vdash^{\mathrm{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta_1, \widehat{\alpha} : \omega_1, \Theta_2 \qquad \Theta_1 \Vdash^{\mathrm{ela}} \rho_2 : \omega_2 \qquad \Theta_1 \Vdash^{\mathrm{u}} [\Theta_1]\omega_1 \approx \omega_2 \dashv \Theta_3}{\Delta \Vdash^{\mathrm{u}} \widehat{\alpha} \approx \rho_1 \dashv \Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2}\ \textsc{a-u-kvarL-tt}$$

$$\begin{array}{l|l}
\Delta \longrightarrow \Theta_1, \widehat{\alpha} : \omega_1, \Theta_2 & \text{By lemma F.6} \\
\Theta_1 \longrightarrow \Theta_3 & \text{I.H.} \\
\Theta_1, \widehat{\alpha} : \omega_1, \Theta_2 \longrightarrow \Theta_3, \widehat{\alpha} : \omega_1, \Theta_2 & \text{By lemma F.27 and lemma F.25} \\
\Theta_1, \widehat{\alpha} : \omega_1, \Theta_2 \longrightarrow \Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2 & \text{lemma F.25} \\
\Delta \longrightarrow \Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2 & \text{lemma F.24}
\end{array}$$

- The case for rule $\textsc{a-u-kvarR-tt}$ is similar as the previous case.

- Case

$$\textsc{a-u-kvarL-lo-tt}\;\frac{\begin{array}{cc}\Delta_1, \Delta_2 \mathrel{+\!\!+}^{\mathsf{mv}} \widehat{\alpha} : \omega_1 \leadsto \Theta & \Delta[\{\Theta\}] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4 \\ \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{ela}} \rho_2 : \omega_2 & \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{u}} [\Theta_1, \Theta_2]\omega_1 \approx \omega_2 \dashv \Theta_5, \{\Theta_6\}\end{array}}{\Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}] \Vdash^{\mathsf{u}} \widehat{\alpha} \approx \rho_1 \dashv \Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4}$$

$\mathsf{topo}\,(\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2) = \Theta$	moving gives one topological sort of the input
$\Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}] \longrightarrow \Delta[\{\Theta\}]$	By definition
$\Delta[\{\Theta\}] \longrightarrow \Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4$	lemma F.6
$\Theta_1, \{\Theta_2\} \longrightarrow \Theta_5, \{\Theta_6\}$	I.H.
$\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4 \longrightarrow \Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4$	By lemma F.27 and lemma F.25
$\Theta_5, \{\Theta_6\} \Vdash^{\mathsf{ela}} \rho_2 : [\Theta_5, \Theta_6]\omega_2$	By lemma F.15
$[\Theta_5, \Theta_6]\omega_2 = [\Theta_5, \Theta_6]([\Theta_1, \Theta_2]\omega_1)$	Lemma F.47
$[\Theta_5, \Theta_6]\omega_2 = [\Theta_5, \Theta_6]\omega_1$	Lemma F.23
$\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4 \longrightarrow \Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4$	lemma F.25
$\Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}] \longrightarrow \Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4$	By lemma F.24

- The case for rule A-U-KVARR-LO-TT is similar as the previous case.
- The case for rule A-U-APP follows directly from I.H. and lemma F.24.
- The case for rule A-U-KAPP follows directly from I.H. and lemma F.24.

□

Lemma F.8 (Well-formedness of Instantiation). *If* $\Delta \Vdash^{\mathsf{inst}} \rho_1 : \eta_1 \sqsubseteq \eta_2 \leadsto \rho_2 \dashv \Theta$, *and* $\Delta \Vdash^{\mathsf{ela}} \rho_1 : \eta_1$, *then* $\Delta \longrightarrow \Theta$, *and* Θ ok, *and* $\Theta \Vdash^{\mathsf{ela}} \rho_2 : [\Theta]\eta_2$.

Proof. By induction on the derivation.

- Case

$$\textsc{a-inst-refl}\;\frac{\Delta \Vdash^{\mathsf{u}} \omega_1 \approx \omega_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{inst}} \mu : \omega_1 \sqsubseteq \omega_2 \leadsto \mu \dashv \Theta}$$

$\Delta \longrightarrow \Theta$	lemma F.7
$[\Theta]\omega_1 = [\Theta]\omega_2$	lemma F.47
$\Delta \Vdash^{\mathsf{ela}} \mu : \omega_1$	Given
$\Theta \Vdash^{\mathsf{ela}} \mu : [\Theta]\omega_1$	lemma F.15
$\Theta \Vdash^{\mathsf{ela}} \mu : [\Theta]\omega_2$	By equations

- Case

$$\textsc{a-inst-forall}\;\frac{\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{inst}} \mu_1 \mathbin{@}\widehat{\alpha} : \eta[a \mapsto \widehat{\alpha}] \sqsubseteq \omega_2 \leadsto \mu_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{inst}} \mu_1 : \forall a : \omega_1 . \eta \sqsubseteq \omega_2 \leadsto \mu_2 \dashv \Theta}$$

$\Delta \longrightarrow \Delta, \widehat{\alpha} : \omega_1$	rule A-CTXE-ADD-TT
$\Delta, \widehat{\alpha} : \omega_1 \longrightarrow \Theta \land \Theta \Vdash^{\mathsf{ela}} \mu_2 : [\Theta]\omega_2$	I.H.
$\Delta \longrightarrow \Theta$	lemma F.24

- The case for rule A-INST-FORALL-INFER is similar to the previous case.

□

Lemma F.10 (Well-formedness of Kinding). *Given* Δ ok,

- *if* $\Delta \Vdash^{\mathsf{k}} \sigma : \eta \rightsquigarrow \mu \dashv \Theta$, *then* $\Delta \longrightarrow \Theta$, *and* $\Theta \Vdash^{\mathsf{ela}} \mu : [\Theta]\eta$;
- *if* $\Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \eta \rightsquigarrow \mu \dashv \Theta$, *then* $\Delta \longrightarrow \Theta$, *and* $\Theta \Vdash^{\mathsf{ela}} \mu : [\Theta]\eta$.
- *if* $\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \eta) \bullet \tau : \omega \rightsquigarrow \rho_2 \dashv \Theta$, *and* $\Delta \Vdash^{\mathsf{ela}} \rho_1 : \eta$, *then* $\Delta \longrightarrow \Theta$, *and* $\Theta \Vdash^{\mathsf{ela}} \rho_2 : [\Theta]\omega$.

Proof. By induction on the derivation.

Part 1 • The case for rules A-KTT-STAR, A-KTT-KUVAR, A-KTT-NAT, A-KTT-VAR, A-KTT-TCON, and A-KTT-ARROW is trivial.

- Case

$$\textsc{a-ktt-forall}\quad \frac{\Delta \Vdash^{\mathsf{kc}} \kappa \Leftarrow \star \rightsquigarrow \omega \dashv \Delta_1 \qquad \Delta_1, a : \omega \Vdash^{\mathsf{kc}} \sigma \Leftarrow \star \rightsquigarrow \mu \dashv \Delta_2, a : \omega, \Delta_3 \qquad \Delta_3 \hookrightarrow a}{\Delta \Vdash^{\mathsf{k}} \forall a : \kappa.\sigma : \star \rightsquigarrow \forall a : \omega.[\Delta_3]\mu \dashv \Delta_2, \mathsf{unsolved}(\Delta_3)}$$

$$\begin{array}{l|l}
\Delta \longrightarrow \Delta_1 \wedge \Delta_1, a : \omega \longrightarrow \Delta_2, a : \omega, \Delta_3 & \text{Part 2} \\
\Delta_1 \longrightarrow \Delta_2 & \text{By inversion} \\
\Delta_2 \longrightarrow \Delta_2, \mathsf{unsolved}(\Delta_3) & \text{By rules A-CTXE-ADD-TT and A-CTXE-ADDSOLVED-TT} \\
\Delta \longrightarrow \Delta_2, \mathsf{unsolved}(\Delta_3) & \text{lemma F.24} \\
\Delta_1 \Vdash^{\mathsf{ela}} \omega : \star \wedge \Delta_2, a : \omega, \Delta_3 \Vdash^{\mathsf{ela}} \mu : \star & \text{I.H.} \\
\Delta_2, \mathsf{unsolved}(\Delta_3) \Vdash^{\mathsf{ela}} \omega : \star. & \text{lemma F.15} \\
\Delta_2, a : \omega, \mathsf{unsolved}(\Delta_3) \Vdash^{\mathsf{ela}} [\Delta_3]\mu : \star. & \text{By Lemma F.18}
\end{array}$$

Because $\mathsf{unsolved}(\Delta_3)$ does not depend on a, we can reorder the context to get that $\Delta_2, \mathsf{unsolved}(\Delta_3), a : \omega \Vdash^{\mathsf{ela}} [\Delta_3]\mu : \star$. So by rule A-ELA-FORALL we get $\Delta_2, \mathsf{unsolved}(\Delta_3) \Vdash^{\mathsf{ela}} \forall a : \omega.[\Delta_3]\mu : \star$.

- The case for rule A-KTT-FORALLI is similar as the previous case.
- Case

$$\textsc{a-ktt-app}\quad \frac{\Delta \Vdash^{\mathsf{k}} \tau_1 : \eta_1 \rightsquigarrow \rho_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{kapp}} (\rho_1 : [\Delta_1]\eta_1) \bullet \tau_2 : \omega \rightsquigarrow \rho \dashv \Theta}{\Delta \Vdash^{\mathsf{k}} \tau_1\ \tau_2 : \omega \rightsquigarrow \rho \dashv \Theta}$$

$$\begin{array}{l|l}
\Delta \longrightarrow \Delta_1 \wedge \Delta_1 \Vdash^{\mathsf{ela}} \rho_1 : [\Delta_1]\eta_1 & \text{Part 2} \\
\Delta_2 \longrightarrow \Theta \wedge \Theta \Vdash^{\mathsf{ela}} \rho : [\Theta]\omega & \text{Part 3} \\
\Delta \longrightarrow \Theta & \text{By lemma F.24}
\end{array}$$

- Case

$$\textsc{a-ktt-kapp}\quad \frac{\Delta \Vdash^{\mathsf{k}} \tau_1 : \eta \rightsquigarrow \rho_1 \dashv \Delta_1 \qquad [\Delta_1]\eta = \forall a : \omega.\eta_2 \qquad \Delta_1 \Vdash^{\mathsf{kc}} \tau_2 \Leftarrow \omega \rightsquigarrow \rho_2 \dashv \Delta_2}{\Delta \Vdash^{\mathsf{k}} \tau_1\ @\tau_2 : \eta_2[a \mapsto \rho_2] \rightsquigarrow \rho_1\ @\rho_2 \dashv \Delta_2}$$

$$\begin{array}{l|l}
\Delta \longrightarrow \Delta_1 \wedge \Delta_1 \Vdash^{\mathsf{ela}} \rho_1 : [\Delta_1]\eta & \text{I.H.} \\
\Delta_1 \longrightarrow \Delta_2 \wedge \Delta_2 \Vdash^{\mathsf{ela}} \rho_2 : [\Delta_2]\omega & \text{Part 2} \\
\Delta \longrightarrow \Delta_2 & \text{lemma F.24} \\
\Delta_1 \Vdash^{\mathsf{ela}} \rho_1 : \forall a : \omega.\eta_2 & \text{by equations} \\
\Delta_2 \Vdash^{\mathsf{ela}} \rho_1 : \forall a : [\Delta_2]\omega.[\Delta_2]\eta_2 & \text{lemma F.15} \\
\Delta_2 \Vdash^{\mathsf{ela}} \rho_1\ @\rho_2 : ([\Delta_2]\eta_2)[a \mapsto [\Delta_2]\rho_2] & \text{By rule A-ELA-KAPP} \\
\Delta_2 \Vdash^{\mathsf{ela}} \rho_1\ @\rho_2 : [\Delta_2](\eta_2[a \mapsto \rho_2]) & \text{By substitution}
\end{array}$$

- Case rule A-KTT-KAPP-INFER is similar as the previous case.

Part 2 We have

$$\begin{array}{l}\textsc{a-kc-sub}\\ \dfrac{\Delta \Vdash^{\mathsf{k}} \sigma : \eta \rightsquigarrow \mu_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{inst}} \mu_1 : [\Delta_1]\eta \sqsubseteq [\Delta_1]\omega \rightsquigarrow \mu_2 \dashv \Delta_2}{\Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \omega \rightsquigarrow \mu_2 \dashv \Delta_2}\end{array}$$

$$\begin{array}{l|l}
\Delta \longrightarrow \Delta_1 \wedge \Delta_1 \Vdash^{\mathsf{ela}} \mu_1 : [\Delta_1]\eta & \text{Part 1}\\
\Delta_1 \longrightarrow \Delta_2 \wedge \Delta_2 \Vdash^{\mathsf{ela}} \mu_2 : [\Delta_2]([\Delta_1]\omega_2) & \text{By lemma F.8}\\
\Delta \longrightarrow \Delta_2 & \text{lemma F.24}\\
[\Delta_2]([\Delta_1]\omega) = [\Delta_2]\omega & \text{lemma F.23}\\
\Delta_2 \Vdash^{\mathsf{ela}} \mu_2 : [\Delta_2]\omega & \text{by equations}
\end{array}$$

Part 3 By induction on the judgment.

- Case

$$\begin{array}{l}\textsc{a-kapp-tt-arrow}\\ \dfrac{\Delta \Vdash^{\mathsf{kc}} \tau \Leftarrow \omega_1 \rightsquigarrow \rho_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \omega_1 \rightarrow \omega_2) \bullet \tau : \omega_2 \rightsquigarrow \rho_1\, \rho_2 \dashv \Theta}\end{array}$$

$$\begin{array}{l|l}
\Delta \longrightarrow \Theta \wedge \Theta \Vdash^{\mathsf{ela}} \rho_2 : [\Theta]\omega_1 & \text{By Part 2}\\
\Delta \Vdash^{\mathsf{ela}} \rho_1 : \omega_1 \rightarrow \omega_2 & \text{Given}\\
\Theta \Vdash^{\mathsf{ela}} \rho_1 : [\Theta]\omega_1 \rightarrow [\Theta]\omega_2 & \text{By Lemma F.15}\\
\Theta \Vdash^{\mathsf{ela}} \rho_1\, \rho_2 : [\Theta]\omega_2 & \text{By rule } \textsc{a-ela-app}
\end{array}$$

- Case

$$\begin{array}{l}\textsc{a-kapp-tt-forall}\\ \dfrac{\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{kapp}} (\rho_1\ @\widehat{\alpha} : \eta[a \mapsto \widehat{\alpha}]) \bullet \tau : \omega \rightsquigarrow \rho \dashv \Theta}{\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \forall a : \omega_1.\eta) \bullet \tau : \omega \rightsquigarrow \rho \dashv \Theta}\end{array}$$

$$\begin{array}{l|l}
\Delta \Vdash^{\mathsf{ela}} \rho_1 : \forall a : \omega_1.\eta & \text{Given}\\
\Delta \longrightarrow \Delta, \widehat{\alpha} : \omega_1 & \text{By rule } \textsc{a-ctxe-add-tt}\\
\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{ela}} \rho_1 : \forall a : \omega_1.\eta & \text{By Lemma F.15}\\
\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{ela}} \widehat{\alpha} : \omega_1 & \text{By rule } \textsc{a-ela-kuvar} \text{ and Lemma F.11}\\
\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{ela}} \rho_1\ @\widehat{\alpha} : \eta[a \mapsto \widehat{\alpha}] & \text{By rule } \textsc{a-ela-kapp}\\
\Delta, \widehat{\alpha} : \omega_1 \longrightarrow \Theta \wedge \Theta \Vdash^{\mathsf{ela}} \rho : [\Theta]\omega & \text{I.H.}\\
\Delta \longrightarrow \Theta & \text{By Lemma F.24}
\end{array}$$

- The case for rule A-KAPP-TT-FORALL-INFER is similar to the previous case.
- Case

$$\begin{array}{l}\textsc{a-kapp-tt-kuvar}\\ \dfrac{\Delta_1, \widehat{\alpha}_1 : \star, \widehat{\alpha}_2 : \star, \widehat{\alpha} : \omega = (\widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2), \Delta_2 \Vdash^{\mathsf{kc}} \tau \Leftarrow \widehat{\alpha}_1 \rightsquigarrow \rho_2 \dashv \Theta}{\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \Vdash^{\mathsf{kapp}} (\rho_1 : \widehat{\alpha}) \bullet \tau : \widehat{\alpha}_2 \rightsquigarrow \rho_1\, \rho_2 \dashv \Theta}\end{array}$$

$$\begin{array}{l|l}
\Delta_1, \widehat{\alpha} : \kappa, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha}_1 : \star, \widehat{\alpha}_2 : \star, \widehat{\alpha} : \kappa = (\widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2), \Delta_2 & \text{By Lemma F.25 and Lemma F.27}\\
\Delta_1, \widehat{\alpha}_1 : \star, \widehat{\alpha}_2 : \star, \widehat{\alpha} : \kappa = (\widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2), \Delta_2 \longrightarrow \Theta \wedge \Theta \Vdash^{\mathsf{ela}} \rho_2 : [\Theta]\widehat{\alpha}_1 & \text{By Part 2}\\
\Delta_1, \widehat{\alpha} : \kappa, \Delta_2 \longrightarrow \Theta & \text{By Lemma F.24}\\
\Delta_1, \widehat{\alpha} : \kappa, \Delta_2 \Vdash^{\mathsf{ela}} \rho_1 : \widehat{\alpha} & \text{Given}\\
\Theta \Vdash^{\mathsf{ela}} \rho_1 : [\Theta]\widehat{\alpha} & \text{By Lemma F.15}\\
\Theta \Vdash^{\mathsf{ela}} \rho_1 : [\Theta]\widehat{\alpha}_1 \rightarrow [\Theta]\widehat{\alpha}_2 & \text{By Lemma F.23}\\
\Theta \Vdash^{\mathsf{ela}} \rho_1\, \rho_2 : [\Theta]\widehat{\alpha}_2 & \text{By rule } \textsc{a-ela-app}
\end{array}$$

□

Lemma F.11 (Well-formedness of Elaborated Kinding). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{ela}} \mu : \eta$, *then* $\Delta \Vdash^{\mathsf{ela}} \eta : \star$, *and* $[\Delta]\eta = \eta$.

Proof. By induction on the derivation.

- The case for rules A-ELA-STAR, A-ELA-NAT, A-ELA-ARROW, A-ELA-FORALL, and A-ELA-FORALL-INFER is straightforward.
- The case for rules A-ELA-KUVAR, A-ELA-VAR, and A-ELA-TCON is similar. Consider

$$\textsc{a-ela-kuvar}\quad \frac{(\widehat{\alpha} : \omega) \in \Delta}{\Delta \Vdash^{\mathsf{ela}} \widehat{\alpha} : [\Delta]\omega}$$

  Given Δ ok, by inversion and weakening, we have $\Delta \Vdash^{\mathsf{ela}} \omega : \star$. By lemma F.17, we have $\Delta \Vdash^{\mathsf{ela}} [\Delta]\omega : \star$. And $[\Delta]([\Delta]\omega) = [\Delta]\omega$
- Case

$$\textsc{a-ela-app}\quad \frac{\Delta \Vdash^{\mathsf{ela}} \rho_1 : \omega_1 \to \omega_2 \qquad \Delta \Vdash^{\mathsf{ela}} \rho_2 : \omega_1}{\Delta \Vdash^{\mathsf{ela}} \rho_1\, \rho_2 : \omega_2}$$

  By I.H., we have $\Delta \Vdash^{\mathsf{ela}} \omega_1 \to \omega_2 : \star$. By inversion we have $\Delta \Vdash^{\mathsf{ela}} \omega_2 : \star$. Similarly $[\Delta]\omega_2 = \omega_2$.
- The case for rules A-ELA-KAPP and A-ELA-KAPP-INFER is similar. Consider

$$\textsc{a-ela-kapp}\quad \frac{\Delta \Vdash^{\mathsf{ela}} \rho_1 : \forall a : \omega.\eta \qquad \Delta \Vdash^{\mathsf{ela}} \rho_2 : \omega}{\Delta \Vdash^{\mathsf{ela}} \rho_1\, @\rho_2 : \eta[a \mapsto [\Delta]\rho_2]}$$

$$\begin{array}{l|l}
\Delta \Vdash^{\mathsf{ela}} \forall a : \omega.\eta : \star & \text{I.H.} \\
\Delta, a : \omega \Vdash^{\mathsf{ela}} \eta : \star & \text{By inversion} \\
\Delta \Vdash^{\mathsf{ela}} \rho_2 : \omega & \text{Given} \\
\Delta \Vdash^{\mathsf{ela}} [\Delta]\rho_2 : \omega & \text{lemma F.17} \\
\Sigma \vdash^{\mathsf{ela}} \eta[a \mapsto [\Delta]\rho_2] : \star & \text{By substiution} \\
[\Delta](\forall a : \omega.\eta) = \forall a : \omega.\eta & \text{by I.H.} \\
[\Delta]\eta = \eta & \text{Follows directly} \\
[\Delta](\eta[a \mapsto [\Delta]\rho_2]) & \\
= ([\Delta]\eta)[a \mapsto ([\Delta]([\Delta]\rho_2))] & \\
= \eta[a \mapsto [\Delta]\rho_2] &
\end{array}$$

□

Lemma F.12 (Well-formedness of Typing Data Constructor Decl.). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{dc}}_{\rho} \mathcal{D} \rightsquigarrow \mu \dashv \Theta$, *then* $\Delta \longrightarrow \Theta$, *and* $\Theta \Vdash^{\mathsf{ela}} \mu : \star$.

Proof. We have

$$\textsc{a-dc-tt}\quad \frac{\Delta, \blacktriangleright_D \Vdash^{\mathsf{k}} \forall \phi.(\overline{\tau_i}^{\,i} \to \rho) : \star \rightsquigarrow \mu \dashv \Theta_1, \blacktriangleright_D, \Theta_2 \qquad \widehat{\phi}^{\mathsf{c}} = \mathsf{unsolved}(\Theta_2)}{\Delta \Vdash^{\mathsf{dc}}_{\rho} \forall \phi. D\, \overline{\tau_i}^{\,i} \rightsquigarrow \forall \{\phi^{\mathsf{c}}\}.(([\Theta_2]\mu)[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}]) \dashv \Theta_1}$$

$$\begin{array}{l|l}
\Delta, \blacktriangleright_D \longrightarrow \Theta_1, \blacktriangleright_D, \Theta_2 \wedge \Theta_1, \blacktriangleright_D, \Theta_2 \Vdash^{\mathsf{ela}} \mu : \star & \text{lemma F.10} \\
\Delta \longrightarrow \Theta_1 \wedge \Theta_2\ \textbf{soft} & \text{By Lemma F.22}
\end{array}$$

$$\begin{array}{l|l}
\Theta_1, \blacktriangleright_D, \widehat{\phi}^{\mathsf{c}} \Vdash^{\mathsf{ela}} [\Theta_2]\mu : \star & \text{By Lemma F.18} \\
\Theta_1, \blacktriangleright_D, \phi^{\mathsf{c}} \Vdash^{\mathsf{ela}} ([\Theta_2]\mu)[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}] : \star & \text{By substitution} \\
\Theta_1, \blacktriangleright_D \Vdash^{\mathsf{ela}} \forall \{\phi^{\mathsf{c}}\}.([\Theta_2]\mu)[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}] : \star & \text{By rule A-ELA-FORALL-INFER} \\
\Theta_1 \Vdash^{\mathsf{ela}} \forall \{\phi^{\mathsf{c}}\}.([\Theta_2]\mu)[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}] : \star & \text{By strengthening}
\end{array}$$

□

Lemma F.13 (Well-formedness of Typing Datatype Declaration). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow \Gamma \dashv \Theta$, *then* $\Delta \longrightarrow \Theta$, *and* $\Theta \Vdash^{\mathsf{ectx}} \Gamma$.

Proof. We have

$$\textsc{a-dt-tt}\quad \frac{(T : \forall \{\phi^{\mathsf{c}}_1\}.\forall \phi^{\mathsf{c}}_2.\ \omega) \in \Delta \qquad \Delta, \phi^{\mathsf{c}}_1, \phi^{\mathsf{c}}_2, \overline{\widehat{\alpha}_i : \star}^{\,i} \Vdash^{\mathsf{u}} [\Delta]\omega \approx (\overline{\widehat{\alpha}_i}^{\,i} \to \star) \dashv \Theta_1, \phi^{\mathsf{c}}_1, \phi^{\mathsf{c}}_2, \overline{\widehat{\alpha}_i : \star = \omega_i}^{\,i} \qquad \overline{\Theta_j, \phi^{\mathsf{c}}_1, \phi^{\mathsf{c}}_2, \overline{a_i : \omega_i}^{\,i} \Vdash^{\mathsf{dc}}_{(T\ @\phi^{\mathsf{c}}_1\ @\phi^{\mathsf{c}}_2\ \overline{a_i}^{\,i})} \mathcal{D}_j \rightsquigarrow \mu_j \dashv \Theta_{j+1}, \phi^{\mathsf{c}}_1, \phi^{\mathsf{c}}_2, \overline{a_i : \omega_i}^{\,i}}^{\,j}}{\Delta \Vdash^{\mathsf{dt}} \mathbf{data}\ T\ \overline{a_i}^{\,i} = \overline{\mathcal{D}_j}^{\,j \in 1..n} \rightsquigarrow \overline{D_j : \forall \{\phi^{\mathsf{c}}_1\}.\forall \phi^{\mathsf{c}}_2.\ \forall \overline{a_i : \omega_i}^{\,i}.\mu_j}^{\,j} \dashv \Theta_{n+1}}$$

$$\begin{array}{l|l}
\Delta, \phi^{\mathsf{c}}_1, \phi^{\mathsf{c}}_2, \overline{\widehat{\alpha}_i : \star}^{\,i} \longrightarrow \Theta_1, \phi^{\mathsf{c}}_1, \phi^{\mathsf{c}}_2, \overline{\widehat{\alpha}_i : \star = \omega_i}^{\,i} & \text{Lemma F.7} \\
\overline{\Theta_j, \phi^{\mathsf{c}}_1, \phi^{\mathsf{c}}_2, \overline{a_i : \omega_i}^{\,i} \longrightarrow \Theta_{j+1}, \phi^{\mathsf{c}}_1, \phi^{\mathsf{c}}_2, \overline{a_i : \omega_i}^{\,i}}^{\,j} & \text{By Lemma F.12} \\
\overline{\Theta_{j+1}, \phi^{\mathsf{c}}_1, \phi^{\mathsf{c}}_2, \overline{a_i : \omega_i}^{\,i} \Vdash^{\mathsf{ela}} \mu_j : \star}^{\,j} & \text{By Lemma F.12} \\
\Delta \longrightarrow \Theta_{n+1} & \text{By Lemma F.22 and Lemma F.24} \\
\overline{\Theta_{n+1}, \phi^{\mathsf{c}}_1, \phi^{\mathsf{c}}_2, \overline{a_i : \omega_i}^{\,i} \Vdash^{\mathsf{ela}} \mu_j : \star}^{\,j} & \text{By Lemma F.15} \\
\overline{\Theta_{n+1} \Vdash^{\mathsf{ela}} \forall \{\phi^{\mathsf{c}}_1\}.\forall \phi^{\mathsf{c}}_2.\ \forall \overline{a_i : \omega_i}^{\,i}.\mu_j : \star}^{\,j} & \text{By rules A-ELA-FORALL and A-ELA-FORALL-INFER} \\
\Theta \Vdash^{\mathsf{ectx}} \overline{D_j : \forall \phi^{\mathsf{c}}.\ \mu_j}^{\,j} & \text{By rule A-ECTX-DCON-TT}
\end{array}$$

□

F.2.2 Properties of Context Extension.

Lemma F.14 (Declaration Preservation). *If* $\Delta \longrightarrow \Theta$, *if a type constructor or a type variable or a kind unification variable is declared in* Δ, *then it is declared in* Θ.

Proof. By a straightforward induction on $\Delta \longrightarrow \Theta$. □

Lemma F.15 (Extension Weakening). *Given* $\Delta \longrightarrow \Theta$, *if* $\Delta \Vdash^{\mathsf{ela}} \mu : \eta$, *then* $\Theta \Vdash^{\mathsf{ela}} \mu : [\Theta]\eta$.

Proof. By a straightforward induction on the elaborated kinding, making use of Lemma F.23.

□

Lemma F.17 (Substitution Kinding). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{ela}} \mu : \eta$, *then* $\Delta \Vdash^{\mathsf{ela}} [\Delta]\mu : \eta$.

Proof. Similar as the proof for Lemma D.12. □

Lemma F.18 (Soft Substitution Kinding). *If* Δ_1, Δ_2 ok, *and* Δ_2 **soft**, *and* $\Delta_1, \Delta_2 \Vdash^{\mathsf{ela}} \mu : \eta$, *then* $\Delta_1, \mathsf{unsolved}(\Delta_2) \Vdash^{\mathsf{ela}} [\Delta_2]\mu : \eta$.

Proof. Similar as the proof for Lemma D.12, making use of substitution properties.

Lemma F.19 (Reflexivity of Context Extension). *If* Δ ok, *then* $\Delta \longrightarrow \Delta$.

Proof. By straightforward induction on Δ ok. The conclusion follows directly from the definition, including the case for local scopes.

□

Lemma F.20 (Well-formedness of Context Extension). *If Δ ok, and $\Delta \longrightarrow \Theta$, then Θ ok.*

Proof. Similar as the proof for Lemma D.15.
For the case

$$\textsc{a-ctxe-lo}\quad \frac{\Delta \longrightarrow \Theta \qquad \Delta, \mathsf{topo}\,(\Delta_1) \longrightarrow \Theta, \Theta_1}{\Delta, \{\Delta_1\} \longrightarrow \Theta, \{\Theta_1\}}$$

As $\Delta, \mathsf{topo}\,(\Delta_1)$ ok preserves a well-formed ordering, by strengthening and weakening we can prove $\Delta, \mathsf{topo}\,(\Delta_1)$ ok. Then by I.H. we have Θ, Θ_1 ok.

□

Lemma F.22 (Extension Order).

(1) *If $\Delta_1, a : \omega, \Delta_2 \longrightarrow \Theta$, then $\Theta = \Theta_1, a : \omega, \Theta_2$, where $\Delta_1 \longrightarrow \Theta_1$. Moreover, if Δ_2* **soft**, *then Θ_2* **soft**.
(2) *If $\Delta_1, T : \eta, \Delta_2 \longrightarrow \Theta$, then $\Theta = \Theta_1, T : \eta, \Theta_2$, where $\Delta_1 \longrightarrow \Theta_1$. Moreover, if Δ_2* **soft**, *then Θ_2* **soft**.
(3) *If $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Theta$, then $\Theta = \Theta_1, \Theta', \Theta_2$, where $\Delta_1 \longrightarrow \Theta_1$, and Θ' is either $\widehat{\alpha} : \omega$ or $\widehat{\alpha} : \omega = \rho$ for some ρ. Moreover, if Δ_2* **soft**, *then Θ_2* **soft**.
(4) *If $\Delta_1, \widehat{\alpha} : \omega = \rho_1, \Delta_2 \longrightarrow \Theta$, then $\Theta = \Theta_1, \widehat{\alpha} : \omega = \rho_2, \Theta_2$, where $\Delta_1 \longrightarrow \Theta_1$, and $[\Theta_1]\rho_1 = [\Theta_1]\rho_2$. Moreover, if Δ_2* **soft**, *then Θ_2* **soft**.
(5) *If $\Delta_1, \{\Delta\}, \Delta_2 \longrightarrow \Theta$, then $\Theta = \Theta_1, \{\Theta\}, \Theta_2$, where $\Delta_1 \longrightarrow \Theta_1$, Moreover, if Δ_2* **soft**, *then Θ_2* **soft**.

Proof. Similar as the proof for Lemma D.17.

□

Lemma F.23 (Substitution Extension Invariance). *If Δ ok, and $\Delta \Vdash^{\mathsf{ela}} \mu : \eta$, and $\Delta \longrightarrow \Theta$, then $[\Theta]\kappa = [\Theta]([\Delta]\mu)$ and $[\Theta]\kappa = [\Delta]([\Theta]\mu)$.*

Proof. Similar as the proof for Lemma D.18.

□

Lemma F.24 (Transitivity of Context Extension). *If Δ' ok, and $\Delta' \longrightarrow \Delta$, and $\Delta \longrightarrow \Theta$, then $\Delta' \longrightarrow \Theta$.*

Proof. By induction on $\Delta \longrightarrow \Theta$. The proof is similar as the proof for Lemma D.19.
For the case

$$\textsc{a-ctxe-lo}\quad \frac{\Delta \longrightarrow \Theta \qquad \Delta, \mathsf{topo}\,(\Delta_1) \longrightarrow \Theta, \Theta_1}{\Delta, \{\Delta_1\} \longrightarrow \Theta, \{\Theta_1\}}$$

$$\begin{array}{l|l}
\Delta' \longrightarrow \Delta, \{\Delta_1\} & \text{Given} \\
\Delta' = \Delta_2, \{\Delta_3\} \land \Delta_2 \longrightarrow \Delta \land \Delta_2, \mathsf{topo}\,(\Delta_3) \longrightarrow \Delta, \Delta_1 & \text{By inversion} \\
\Delta_2, \mathsf{topo}\,(\Delta_3) \longrightarrow \Delta, \mathsf{topo}\,(\Delta_1) & \text{By reordering } \Delta_3 \text{ according to } \mathsf{topo}\,(\Delta_1) \\
\Delta_2, \mathsf{topo}\,(\Delta_3) \longrightarrow \Theta, \Theta_1 & \text{I.H.} \\
\Delta_2, \{\Delta_3\} \longrightarrow \Theta, \{\Theta_1\} & \text{By rule } \textsc{a-ctxe-lo}
\end{array}$$

□

Lemma F.25 (Solution Admissibility for Extension).

- *If* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2$ ok *and* $\Delta_1 \Vdash^{\mathsf{ela}} \rho : [\Delta_1]\omega$*, then* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha} : \omega = \rho, \Delta_2$.
- *If* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega, \Delta_4\}, \Delta_2$ ok *and* $\Delta_1, \Delta_3 \Vdash^{\mathsf{ela}} \rho : [\Delta_1, \Delta_3]\omega$*, then* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega, \Delta_4\}, \Delta_2 \longrightarrow \Delta_1, \{\Delta_3, \widehat{\alpha} : \omega = \rho, \Delta_4\}, \Delta_2$.

Proof. **Part 1** By induction on Δ_2. The proof is similar as the proof for Lemma D.20. For the case $\Delta_2 = \Delta_2', \{\Delta_3\}$. By I.H., we have $\Delta_1, \widehat{\alpha} : \omega \longrightarrow \Delta_1, \widehat{\alpha} : \omega = \rho$. Then by rule A-CTXE-LO we have $\Delta_1, \widehat{\alpha} : \omega, \{\Delta_2'\} \longrightarrow \Delta_1, \widehat{\alpha} : \omega = \rho, \{\Delta_2'\}$.

Part 2 By induction on Δ_2. Most cases are similar as Part 1. When Δ_2 is empty, we only need to prove $\Delta_1, \Delta_3, \widehat{\alpha} : \omega, \Delta_4 \longrightarrow \Delta_1, \Delta_3, \widehat{\alpha} : \omega = \rho, \Delta_4$. By referring Part 1 we are done.

□

Lemma F.26 (Solved Variable Addition for Extension).

- *If* Δ_1, Δ_2 ok *and* $\Delta_1 \Vdash^{\mathsf{ela}} \rho : [\Delta_1]\omega$*, then* $\Delta_1, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha} : \omega = \rho, \Delta_2$.
- *If* $\Delta_1, \{\Delta_2, \Delta_3\}, \Delta_4$ ok *and* $\Delta_1, \Delta_2 \Vdash^{\mathsf{ela}} \rho : [\Delta_1, \Delta_2]\omega$*, then* $\Delta_1, \{\Delta_2, \Delta_3\}, \Delta_4 \longrightarrow \Delta_1, \{\Delta_2, \widehat{\alpha} : \omega = \rho, \Delta_3\}, \Delta_4$.

Proof. The proof is exactly the same as the one for lemma F.25. Except for the case when Δ_2 is empty, we use rule A-CTXE-ADDSOLVED-TT.

□

Lemma F.27 (Unsolved Variable Addition).

- *If* Δ_1, Δ_2 ok *and* $\Delta_1 \Vdash^{\mathsf{ela}} \omega : \star$ *then* $\Delta_1, \Delta_2 \longrightarrow \Delta_1, \widehat{\alpha} : \omega, \Delta_2$.
- *If* $\Delta_1, \{\Delta_2, \Delta_3\}, \Delta_4$ ok *and* $\Delta_1, \Delta_2 \Vdash^{\mathsf{ela}} \omega : \star$*, then* $\Delta_1, \{\Delta_2, \Delta_3\}, \Delta_4 \longrightarrow \Delta_1, \{\Delta_2, \widehat{\alpha} : \omega, \Delta_3\}, \Delta_4$.

Proof. The proof is exactly the same as the one for lemma F.25. Except for the case when Δ_2 is empty, we use rule A-CTXE-ADD-TT.

□

Lemma F.28 (Parallel Admissibility).

- *If* $\Delta_1 \longrightarrow \Theta_1$*, and* Δ_1, Δ_2 ok*, and* $\Delta_1, \Delta_2 \longrightarrow \Theta_1, \Theta_2$*, and* Δ_2 *is fresh w.r.t.* Θ_1*, then:*
  - *if* $\Delta_1 \Vdash^{\mathsf{ela}} \omega : \star$*, then* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega, \Theta_2$*;*
  - *if* $\Theta_1 \Vdash^{\mathsf{ela}} \rho : [\Theta_1]\omega$*, then* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \rho, \Theta_2$*;*
  - *if* $[\Theta_1]\rho_1 = [\Theta_1]\rho_2$*, then* $\Delta_1, \widehat{\alpha} : \omega = \rho_1, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \rho_2, \Theta_2$.
- *If* $\Delta_1, \{\Delta_3\} \longrightarrow \Theta_1, \{\Theta_3\}$*, and* $\Delta_1, \{\Delta_3, \Delta_4\}, \Delta_2$ ok*, and* $\Delta_1, \{\Delta_3, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \Theta_4\}, \Theta_2$*, and* Δ_2, Δ_4 *is fresh w.r.t.* Θ_1, Θ_3*, then:*
  - *if* $\Delta_1, \{\Delta_3\} \Vdash^{\mathsf{ela}} \omega : \star$*, then* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega, \Theta_4\}, \Theta_2$*;*
  - *if* $\Theta_1, \{\Theta_3\} \Vdash^{\mathsf{ela}} \rho : [\Theta_1, \Theta_3]\omega$*, then* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega = \rho, \Theta_4\}, \Theta_2$*;*
  - *if* $[\Theta_1, \Theta_3]\rho_1 = [\Theta_1, \Theta_3]\rho_2$*, then* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega = \rho_1, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega = \rho_2, \Theta_4\}, \Theta_2$.

Proof. **Part 1** By induction on the size of Θ_2. The proof is similar as in Lemma D.23. In the first subcase, when the case $\Theta_2 = \Theta_3, \{\Theta_4\}$. Then the derivation of $\Delta_1, \Delta_2 \longrightarrow \Theta_1, \Theta_2$ must conclude with rule A-CTXE-LO. It must be $\Delta_2 = \Delta_{21}, \{\Delta_{22}\}$.

$\Delta_1, \Delta_{21}, \{\Delta_{22}\} \longrightarrow \Theta_1, \Theta_3, \{\Theta_4\}$	Given
$\Delta_1, \Delta_{21}, \mathrm{topo}\,(\Delta_{22}) \longrightarrow \Theta_1, \Theta_3, \Theta_4$	By inversion
$\Delta_1, \widehat{\alpha} : \omega, \Delta_{21}, \mathrm{topo}\,(\Delta_2) \longrightarrow \Theta_1, \widehat{\alpha} : \omega, \Theta_3, \Theta_4$	I.H.
$\Delta_1, \widehat{\alpha} : \omega, \Delta_{21}, \{\Delta_2\} \longrightarrow \Theta_1, \widehat{\alpha} : \omega, \Theta_3, \{\Theta_4\}$	By rule A-CTXE-LO

Part 2 By induction on Θ_2. Most cases are similar to Part 1. For the first case, when Θ_2 is empty, we know Δ_2 is empty. We have $\Delta_1, \text{topo}\,(\Delta_3), \text{topo}\,(\Delta_4) \longrightarrow \Theta_1, \Theta_3, \Theta_4$. By Part 1 we know $\Delta_1, \text{topo}\,(\Delta_3), \widehat{\alpha} : \omega, \text{topo}\,(\Delta_4) \longrightarrow \Theta_1, \Theta_3, \widehat{\alpha} : \omega, \Theta_4$. By rule A-CTXE-LO we have $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega, \Delta_4\} \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega, \Theta_4\}$. The rest cases are similar. Rest cases are similar as the first case.

□

Lemma F.29 (Parallel Extension Solution).

- *If* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \rho_2, \Theta_2$, *and* $[\Theta_1]\rho_1 = [\Theta_1]\rho_2$, *then* $\Delta_1, \widehat{\alpha} : \omega = \rho_1, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \kappa_2, \Theta_2$.
- *If* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega = \rho_2, \Theta_4\}, \Theta_2$, *and* $[\Theta_1, \Theta_3]\rho_1 = [\Theta_1, \Theta_3]\rho_2$, *then* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega = \rho_1, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega = \rho_2, \Theta_4\}, \Theta_2$.

Proof. **Part 1** By induction on Θ_2. The proof is similar to Lemma D.24. For the case when $\Theta_2 = \Theta_3, \{\Theta_4\}$. Then $\Delta_2 = \Delta_3, \{\Delta_4\}$. And $\Delta_1, \widehat{\alpha} : \omega, \Delta_3, \{\Delta_4\} \longrightarrow \Theta_1, \widehat{\alpha} : \omega, \Theta_3, \{\Theta_4\}$. By inversion, we have $\Delta_1, \widehat{\alpha} : \omega, \Delta_3, \text{topo}\,(\Delta_4) \longrightarrow \Theta_1, \widehat{\alpha} : \omega, \Theta_3, \Theta_4$. By I.H., we have $\Delta_1, \widehat{\alpha} : \omega = \rho_1, \Delta_3, \text{topo}\,(\Delta_4) \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \rho_2, \Theta_3, \Theta_4$. By rule A-CTXE-LO we have $\Delta_1, \widehat{\alpha} : \omega = \rho_1, \Delta_3, \{\Delta_4\} \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \rho_2, \Theta_3, \{\Theta_4\}$.

Part 2 By induction on Θ_2. Most cases are similar to Part 1. We discuss when Θ_2 is empty. Then Δ_2 must to empty. From givens we know that $\Delta_1, \Delta_5, \widehat{\alpha} : \omega, \Delta_6 \longrightarrow \Theta_1, \Theta_3, \widehat{\alpha} : \omega = \rho_2, \Theta_4$ where $\Delta_5, \widehat{\alpha} : \omega, \Delta_6 = \text{topo}\,(\Delta_3, \widehat{\alpha} : \omega, \Delta_4)$. By Part 1 we have $\Delta_1, \Delta_5, \widehat{\alpha} : \omega = \rho_1, \Delta_6 \longrightarrow \Theta_1, \Theta_3, \widehat{\alpha} : \omega = \rho_2, \Theta_4$. Since $\Delta_5, \widehat{\alpha} : \omega = \rho_1, \Delta_6 = \text{topo}\,(\Delta_3, \widehat{\alpha} : \omega = \rho_1, \Delta_4)$, we have $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega = \rho_1, \Delta_4\} \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega = \rho_2, \Theta_4\}$.

Lemma F.30 (Parallel Variable Update).

- *If* $\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \rho, \Theta_2$, *and* $\Delta_1 \Vdash^{\text{ela}} \rho_1 : [\Delta_1]\omega$, *and* $\Theta_1 \Vdash^{\text{ela}} \rho_2 : [\Theta_1]\omega$, *and* $[\Theta_1]\rho = [\Theta_1]\rho_1 = [\Theta_1]\rho$, *then* $\Delta_1, \widehat{\alpha} : \omega = \rho_1, \Delta_2 \longrightarrow \Theta_1, \widehat{\alpha} : \omega = \rho_2, \Theta_2$
- *If* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega = \rho, \Theta_4\}, \Theta_2$, *and* $\Delta_1, \Delta_3 \Vdash^{\text{ela}} \rho_1 : [\Delta_1, \Delta_3]\omega$, *and* $\Theta_1, \Theta_3 \Vdash^{\text{ela}} \rho_2 : [\Theta_1, \Theta_3]\omega$, *and* $[\Theta_1]\rho = [\Theta_1]\rho_1 = [\Theta_1]\rho$, *then* $\Delta_1, \{\Delta_3, \widehat{\alpha} : \omega = \rho_1, \Delta_4\}, \Delta_2 \longrightarrow \Theta_1, \{\Theta_3, \widehat{\alpha} : \omega = \rho_2, \Theta_4\}, \Theta_2$

Proof. The proof is exactly the same as the one for lemma F.29. Except for the case when Θ_2 is empty, we use rule A-CTXE-SOLVE-TT.

□

F.2.3 Properties of Complete Context.

Lemma F.31 (Finishing Kinding). *If* Ω ok, *and* $\Omega \Vdash^{\text{ela}} \rho : \omega$, *and* $\Omega \longrightarrow \Omega'$, *then* $[\Omega]\rho = [\Omega']\rho$.

Proof. By lemma F.23 we know $[\Omega']\rho = [\Omega']([\Omega]\rho)$. Because $[\Omega]\rho$ contains no unsolved kind unification variable, we have $[\Omega']([\Omega]\rho) = [\Omega]\rho$. Therefore $[\Omega']\rho = [\Omega]\rho$.

□

Lemma F.32 (Finishing Term Contexts). *If* Ω ok, *and* $\Omega \Vdash^{\text{ectx}} \Gamma$, *and* $\Omega \longrightarrow \Omega'$, *then* $[\Omega']\Gamma = [\Omega]\Gamma$.

Proof. The proof is exactly the same as in Lemma D.27, making use of Lemma F.31.

□

Lemma F.33 (Stability of Complete Contexts). *If* $\Delta \longrightarrow \Omega$, *then* $[\Omega]\Delta = [\Omega]\Omega$.

Proof. By induction on $\Delta \longrightarrow \Omega$. Most cases are the same as Lemma D.28. For the case

A-CTXE-LO

$$\frac{\Delta \longrightarrow \Theta \qquad \Delta, \text{topo}\,(\Delta_1) \longrightarrow \Theta, \Theta_1}{\Delta, \{\Delta_1\} \longrightarrow \Theta, \{\Theta_1\}}$$

$$\begin{array}{l|l}
\Delta, \{\Delta_1\} \longrightarrow \Omega, \{\Omega_1\} & \text{Given} \\
\Delta, \mathsf{topo}\,(\Delta_1) \longrightarrow \Omega, \Omega_1 & \text{Given} \\
[\Omega, \{\Omega_1\}](\Omega, \{\Omega_1\}) & \\
= [\Omega, \Omega_1](\Omega, \Omega_1) & \text{By definition} \\
= [\Omega, \Omega_1](\Delta, \mathsf{topo}\,(\Delta_1)) & \text{I.H.} \\
= [\Omega, \Omega_1](\Delta, \{\Delta_1\}) & \text{By definition}
\end{array}$$

□

Lemma F.34 (Softness Goes Away). *If* $\Delta_1, \Delta_2 \longrightarrow \Omega_1, \Omega_2$ *where* $\Delta_1 \longrightarrow \Omega_1$, *and* Δ_2 **soft**, *then* $[\Omega_1, \Omega_2](\Delta_1, \Delta_2) = [\Omega_1]\Delta_1$.

Proof. By induction on Δ_2 and the goal follows directly from the definition of context application.

□

Lemma F.35 (Confluence of Completeness). *If* $\Delta_1 \longrightarrow \Omega$, *and* $\Delta_2 \longrightarrow \Omega$, *then* $[\Omega]\Delta_1 = [\Omega]\Delta_2$.

Proof. By lemma F.33 we have $[\Omega]\Delta_1 = [\Omega]\Omega$ and $[\Omega]\Delta_2 = [\Omega]\Omega$. Therefore $[\Omega]\Delta_1 = [\Omega]\Delta_2$.

□

Lemma F.36 (Substitution Stability). *If* Ω, Δ ok, *and* $\Omega \Vdash^{\mathsf{ela}} \rho : \omega$, *then* $[\Omega]\rho = [\Omega, \Delta]\rho$.

Proof. By straightforward induction on $\mid \Omega \vdash \rho \mid$.

□

Lemma F.37 (Type Constructor Preservation). *If* Δ ok, *then* $(T : \eta) \in \Delta$, *and* $\Delta \longrightarrow \Omega$, *then* $(T : [\Omega]\eta) \in [\Omega]\Delta$.

Proof. The proof is exactly the same as Lemma D.32, making use of Lemma F.36.

□

Lemma F.38 (Type Variable Preservation). *If* $(a : \omega) \in \Delta$, *and* $\Delta \longrightarrow \Omega$, *then* $(a : [\Omega]\omega) \in [\Omega]\Delta$.

Proof. This lemma is similar to lemma F.37.

□

Lemma F.39 (Finishing Completions). *If* Ω ok, *and* $\Omega \longrightarrow \Omega'$, *then* $[\Omega']\Omega'$ *is a well-formed permutation of* $[\Omega]\Omega$.

Proof. By induction on $\Omega \longrightarrow \Omega'$. Most cases are the same as Lemma D.34.
For the case

$$\textsc{a-ctxe-lo}\quad \frac{\Delta \longrightarrow \Theta \qquad \Delta, \mathsf{topo}\,(\Delta_1) \longrightarrow \Theta, \Theta_1}{\Delta, \{\Delta_1\} \longrightarrow \Theta, \{\Theta_1\}}$$

$$\begin{array}{l|l}
\Omega, \{\Omega_1\} \longrightarrow \Omega', \{\Omega'_1\} \wedge \Omega, \mathsf{topo}\,(\Omega_1) \longrightarrow \Omega', \Omega'_1 \wedge \Omega \longrightarrow \Omega' & \text{Given} \\
[\Omega', \{\Omega'_1\}](\Omega', \{\Omega'_1\}) & \\
= [\Omega', \Omega'_1](\Omega', \Omega'_1) & \text{By definition} \\
= [\Omega, \mathsf{topo}\,(\Omega_1)](\Omega, \mathsf{topo}\,(\Omega_1)) & \text{I.H.} \\
= \mathsf{topo}\,([\Omega, \Omega_1](\Omega, \Omega_1)) & \text{Follows}
\end{array}$$

□

F.2.4 Decidability.

Lemma F.40 (Promotion Preserves $\langle\Delta\rangle$)**.** *If* $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \omega_2 \dashv \Theta$*, then* $\langle\Delta\rangle = \langle\Theta\rangle$*.*

Proof. By a straightforward induction on the derivation. □

Lemma F.41 (Unification Makes Progress)**.** *If* $\Delta \Vdash^{\mathsf{u}} \omega_1 \approx \omega_2 \dashv \Theta$*, then either* $\Theta = \Delta$*, or* $\langle\Theta\rangle < \langle\Delta\rangle$*.*

Proof. By induction on the derivation.

- In rule A-U-REFL-TT, the goal holds trivially.
- Case

$$\frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta_1, \widehat{\alpha} : \omega_1, \Theta_2 \qquad \Theta_1 \Vdash^{\mathsf{ela}} \rho_2 : \omega_2 \qquad \Theta_1 \Vdash^{\mathsf{u}} [\Theta_1]\omega_1 \approx \omega_2 \dashv \Theta_3}{\Delta \Vdash^{\mathsf{u}} \widehat{\alpha} \approx \rho_1 \dashv \Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2}\ \textsc{a-u-kvarL-tt}$$

$$\begin{array}{l|l}
\langle\Theta_1, \widehat{\alpha} : \omega_1, \Theta_2\rangle = \langle\Delta\rangle & \text{lemma F.40} \\
\Theta_3 = \Theta_1 \cup \langle\Theta_3\rangle < \langle\Theta_1\rangle & \text{I.H.} \\
\langle\Theta_3\rangle \leqslant \langle\Theta_1\rangle & \text{Follows} \\
\langle\Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2\rangle < \langle\Theta_3, \widehat{\alpha} : \omega_1, \Theta_2\rangle & \\
\leqslant \langle\Theta_1, \widehat{\alpha} : \omega_1, \Theta_2\rangle & \\
= \langle\Delta\rangle &
\end{array}$$

- The case for rule A-U-KVARR-TT is similar as the previous case.
- Case

$$\frac{\begin{array}{cc}\Delta_1, \Delta_2 \mathrel{+\!\!+}^{\mathsf{mv}} \widehat{\alpha} : \omega_1 \leadsto \Theta & \Delta[\{\Theta\}] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4 \\ \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{ela}} \rho_2 : \omega_2 & \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{u}} [\Theta_1, \Theta_2]\omega_1 \approx \omega_2 \dashv \Theta_5, \{\Theta_6\}\end{array}}{\Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}] \Vdash^{\mathsf{u}} \widehat{\alpha} \approx \rho_1 \dashv \Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4}\ \textsc{a-u-kvarL-lo-tt}$$

$$\begin{array}{l|l}
\langle\Theta\rangle = \langle\Delta_1, \widehat{\alpha} : \omega, \Delta_2\rangle & \text{By moving} \\
\langle\Delta[\{\Theta\}]\rangle = \langle\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4\rangle & \text{By lemma F.40} \\
\Theta_5, \{\Theta_6\} = \Theta_1, \{\Theta_2\} \cup \Theta_5, \{\Theta_6\} < \langle\Theta_1, \{\Theta_2\}\rangle & \text{I.H.} \\
\langle\Theta_5, \{\Theta_6\}\rangle \leqslant \langle\Theta_1, \{\Theta_2\}\rangle & \text{Follows} \\
\langle\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4\rangle & \\
< \langle\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4\rangle & \\
\leqslant \langle\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4\rangle & \\
= \langle\Delta[\{\Theta\}]\rangle & \\
= \langle\Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}]\rangle. &
\end{array}$$

- The case for rule A-U-KVARR-LO-TT is similar as the previous case.
- Case

$$\frac{\Delta \Vdash^{\mathsf{u}} \rho_1 \approx \rho_3 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{u}} [\Delta_1]\rho_2 \approx [\Delta_1]\rho_4 \dashv \Theta}{\Delta \Vdash^{\mathsf{u}} \rho_1\, \rho_2 \approx \rho_3\, \rho_4 \dashv \Theta}\ \textsc{a-u-app}$$

$$\begin{array}{l|l}
\Delta_1 = \Delta \cup \langle\Delta_1\rangle < \langle\Delta\rangle & \text{I.H.} \\
\Delta_1 = \Theta \cup \langle\Theta\rangle < \langle\Delta_1\rangle & \text{I.H.} \\
\text{If } \Delta_1 = \Delta \text{ and } \Delta_1 = \Theta & \\
\Delta = \Theta & \text{Follows directly} \\
\text{Otherwise} & \\
\langle\Theta\rangle < \langle\Delta\rangle & \text{Follows directly}
\end{array}$$

- The case for rule A-U-KAPP is similar as the previous case.

□

Lemma F.42 (Promotion Preserves $|\rho|$). *Given a context* $\Delta[\widehat{\alpha}]$ ok, *if* $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \omega_2 \dashv \Theta$, *then for all* ρ, *we have* $|[\Delta]\rho| = |[\Theta]\rho|$.

Proof. By a straightforward induction on the promotion judgment.

- Most cases we have $\Delta = \Theta$. So the goal follows trivially.
- Case

$$\textsc{a-pr-app}\quad \frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \rho_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} [\Delta_1]\omega_2 \rightsquigarrow \rho_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1\, \omega_2 \rightsquigarrow T \dashv \Theta}$$

The goal follows directly from I.H..
- The case for rule A-PR-KAPP is the same as the previous case.
- Case

$$\textsc{a-pr-kuvarR-tt}\quad \frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} [\Delta]\rho \rightsquigarrow \rho_1 \dashv \Theta[\widehat{\alpha}][\widehat{\beta} : \rho]}{\Delta[\widehat{\alpha}][\widehat{\beta} : \rho] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \widehat{\beta} \rightsquigarrow \widehat{\beta_1} \dashv \Theta[\widehat{\beta_1} : \rho_1, \widehat{\alpha}][\widehat{\beta} : \rho = \widehat{\beta_1}]}$$

From I.H., we have $|[\Theta[\widehat{\alpha}][\widehat{\beta} : \rho]]\rho| = |[\Delta]\rho|$. If ρ contains no $\widehat{\beta}$, then $[\Theta[\widehat{\beta_1} : \rho_1, \widehat{\alpha}][\widehat{\beta} : \rho = \widehat{\beta_1}]]\rho = [\Theta[\widehat{\alpha}][\widehat{\beta} : \rho]]\rho$. Then we have $|[\Theta[\widehat{\beta_1} : \rho_1, \widehat{\alpha}][\widehat{\beta} : \rho = \widehat{\beta_1}]]\rho| = |[\Delta]\rho|$. Otherwise, all $\widehat{\beta}$ are replaced by $\widehat{\beta_1}$, which also preserves the size.

□

Theorem F.43 (Promotion is Decidable). *Given a context* $\Delta[\widehat{\alpha}]$ ok, *and a kind* ρ_1 *with* $[\Delta]\rho_1 = \rho_1$, *it is decidable whether there exists* Θ *such that* $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \omega_2 \dashv \Theta$.

Proof. Draw the dependency graph of the input context. We measure the promotion process $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \omega_2 \dashv \Theta$ by the lexicographic order of

(1) the maximal height of the being promoted types in the dependency graph;
(2) $|\omega_1|$.

We prove the measurement always get smaller from the conclusion to the hypothesis.

We first prove (1) gets no larger from the conclusion to the premises. This can be done via a straightforward induction on the promotion judgment.

Now we induction on the promotion judgment.

- Most cases do not have hypothesis.
- Case

$$\textsc{a-pr-app}\quad \frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \rho_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} [\Delta_1]\omega_2 \rightsquigarrow \rho_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1\, \omega_2 \rightsquigarrow T \dashv \Theta}$$

$\lvert\omega_1\rvert < \lvert\omega_1\, \omega_2\rvert$	Follows directly
The height of Δ_1 is no larger than Δ	Given
$\lvert[\Delta_1]\omega_2\rvert = \lvert[\Delta]\omega_2\rvert$	By Lemma F.42
$= \lvert\omega_2\rvert$	Given the equation
$< \lvert\omega_1\, \omega_2\rvert$	Follows

- The case for rule A-PR-KAPP is the same as the previous case.

- Case

$$\frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} [\Delta]\rho \rightsquigarrow \rho_1 \dashv \Theta[\widehat{\alpha}][\widehat{\beta} : \rho]}{\Delta[\widehat{\alpha}][\widehat{\beta} : \rho] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \widehat{\beta} \rightsquigarrow \widehat{\beta}_1 \dashv \Theta[\widehat{\beta}_1 : \rho_1, \widehat{\alpha}][\widehat{\beta} : \rho = \widehat{\beta}_1]}\ \textsc{a-pr-kuvarR-tt}$$

  In the dependency graph, there are edges from $\widehat{\beta}$ to $[\Delta]\rho$. So the height gets decreased from the conclusion to the hypothesis.

□

Theorem F.44 (Unification is Decidable). *Given a context* Δ ok*, and kinds* ρ_1 *and* ρ_2*, where* $[\Delta]\rho_1 = \rho_1$*, and* $[\Delta]\rho_2 = \rho_2$*, it is decidable whether there exists* Θ *such that* $\Delta \Vdash^{\mathsf{u}} \rho_1 \approx \rho_2 \dashv \Theta$.

Proof. We measure the unification derivation by the lexicographic order on:

(1) $\langle \Delta \rangle$

(2) $|\rho_1|$

We case analyze the derivation.

- The case for rule A-U-REFL-TT is decidable.
- Case

$$\frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta_1, \widehat{\alpha} : \omega_1, \Theta_2 \qquad \Theta_1 \Vdash^{\mathsf{ela}} \rho_2 : \omega_2 \qquad \Theta_1 \Vdash^{\mathsf{u}} [\Theta_1]\omega_1 \approx \omega_2 \dashv \Theta_3}{\Delta \Vdash^{\mathsf{u}} \widehat{\alpha} \approx \rho_1 \dashv \Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2}\ \textsc{a-u-kvarL-tt}$$

$$\begin{array}{l|l}
\langle \Theta_1, \widehat{\alpha} : \omega_1, \Theta_2 \rangle = \langle \Delta \rangle & \text{lemma F.40} \\
\langle \Theta_1 \rangle < \langle \Theta_1, \widehat{\alpha} : \omega_1, \Theta_2 \rangle = \langle \Delta \rangle & \text{Follows}
\end{array}$$

- The case for rule A-U-KVARR-TT is similar as the previous case.
- Case

$$\frac{\begin{array}{c}\Delta_1, \Delta_2 \mathrel{+\!\!+}^{\mathsf{mv}} \widehat{\alpha} : \omega_1 \leadsto \Theta \qquad \Delta[\{\Theta\}] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4 \\ \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{ela}} \rho_2 : \omega_2 \qquad \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{u}} [\Theta_1, \Theta_2]\omega_1 \approx \omega_2 \dashv \Theta_5, \{\Theta_6\}\end{array}}{\Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}] \Vdash^{\mathsf{u}} \widehat{\alpha} \approx \rho_1 \dashv \Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4}\ \textsc{a-u-kvarL-lo-tt}$$

$$\begin{array}{l|l}
\langle \Theta \rangle = \langle \Delta_1, \widehat{\alpha} : \omega_1, \Delta_2 \rangle & \text{By moving} \\
\langle \Delta[\{\Theta\}] \rangle = \langle \Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}] \rangle & \text{Follows} \\
\langle \Delta[\{\Theta\}] \rangle = \langle \Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4 \rangle & \text{Lemma F.40} \\
\langle \Theta_1, \{\Theta_2\} \rangle & \\
< \langle \Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4 \rangle & \\
= \langle \Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}] \rangle &
\end{array}$$

- The case for rule A-U-KVARR-LO-TT is similar as the previous case.
- Case

$$\frac{\Delta \Vdash^{\mathsf{u}} \rho_1 \approx \rho_3 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{u}} [\Delta_1]\rho_2 \approx [\Delta_1]\rho_4 \dashv \Theta}{\Delta \Vdash^{\mathsf{u}} \rho_1\, \rho_2 \approx \rho_3\, \rho_4 \dashv \Theta}\ \textsc{a-u-app}$$

  For the first condition, we know that $\langle \Delta \rangle = \langle \Delta \rangle$ and the size of the expression decreases. For the second condition, from lemma F.41, we know that either $\Delta_1 = \Delta$, or $\langle \Delta_1 \rangle < \langle \Delta \rangle$. In the former case, we know that $[\Delta_1]\rho_2 = \rho_2$. So the size of the expression decreases. In the latter case, we have $\langle \Delta_1 \rangle < \langle \Delta \rangle$ so we are done.
- The case for rule A-U-KAPP is similar as the previous one.

$\square$

F.2.5 Source of Unification Variables.

Lemma F.45 (Source of Unification Variables). *If* $\Delta \Vdash^{k} \sigma : \eta \leadsto \mu \dashv \Theta$, *then for any* $\widehat{\alpha} \in \mathsf{unsolved}(\Theta)$, *either* $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\mu)$, *or there exists* $\widehat{\beta} \in \mathsf{unsolved}(\Delta)$ *such that* $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\widehat{\beta})$.

Proof. This lemma depends on the similar lemma on many judgments, including kind checking, instantiation, and unification. We prove them one by one.

When the input context is the same as the output context, the lemma holds trivially, as all unsolved unification variables in Θ are in Δ. So we will skip the discussion of those cases.

Part 1: Kinding By induction on the judgment.

- Case

  A-KTT-FORALL
  $$\frac{\Delta \Vdash^{kc} \kappa \Leftarrow \star \leadsto \omega \dashv \Delta_1 \qquad \Delta_1, a : \omega \Vdash^{kc} \sigma \Leftarrow \star \leadsto \mu \dashv \Delta_2, a : \omega, \Delta_3 \qquad \Delta_3 \hookrightarrow a}{\Delta \Vdash^{k} \forall a : \kappa.\sigma : \star \leadsto \forall a : \omega.[\Delta_3]\mu \dashv \Delta_2, \mathsf{unsolved}(\Delta_3)}$$

  Given $\widehat{\alpha} \in \mathsf{unsolved}(\Delta_2, \mathsf{unsolved}(\Delta_3))$, we know that $\widehat{\alpha} \in \mathsf{unsolved}(\Delta_2, a : \omega, \Delta_3)$. Then by the lemma on kind checking. we have two cases.

  (1) $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2, a : \omega, \Delta_3]\mu)$. Then

  (a) $\widehat{\alpha} \in \mathsf{fkv}(\mu)$, and $\widehat{\alpha}$ is unsolved in $\Delta_2, a : \omega, \Delta_3$.

  Therefore $\widehat{\alpha} \in \mathsf{fkv}([\Delta_3]\mu)$.

  Since $\widehat{\alpha} \in \mathsf{unsolved}(\Delta_2, \mathsf{unsolved}(\Delta_3))$, we have $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2, \mathsf{unsolved}(\Delta_3)]([\Delta_3]\mu))$ so we are done.

  (b) there exists a $\widehat{\beta_2} \in \mathsf{fkv}(\mu)$, such that $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2, a : \omega, \Delta_3]\widehat{\beta_2})$.

  Now the goal is to prove $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2, \mathsf{unsolved}(\Delta_3)]([\Delta_3]\widehat{\beta_2}))$.

  Notice that $[\Delta_2, \mathsf{unsolved}(\Delta_3)]([\Delta_3]\widehat{\beta_2}) = [\Delta_2]([\Delta_3]\widehat{\beta_2}) = [\Delta_2, a : \omega, \Delta_3]\widehat{\beta_2}$.

  So we are done.

  (2) there exists $\widehat{\beta_1} \in \mathsf{unsolved}(\Delta_1, a : \omega)$ such that $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2, a : \omega, \Delta_3]\widehat{\beta_1})$.

  Because $\widehat{\beta_1}$ is in $\Delta_1, a : \omega$, then it must be $\widehat{\beta_1}$ in $\Delta_2, a : \omega$ by Lemma F.22 and Lemma F.14.

  Therefore $[\Delta_2, a : \omega, \Delta_3]\widehat{\beta_1} = [\Delta_2]\widehat{\beta_1}$.

  So we have $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2]\widehat{\beta_1})$.

  Also, it must be $\widehat{\beta_1} \in \mathsf{unsolved}(\Delta_1)$. Then by the lemma on kind checking. we have two subcases.

  (a) $\widehat{\beta_1} \in \mathsf{fkv}([\Delta_1]\omega)$.

  We know that $\Delta_1 \longrightarrow \Delta_2$ by lemma F.10 and Lemma F.22.

  So $[\Delta_2, \mathsf{unsolved}(\Delta_3)]\omega = [\Delta_2]\omega = [\Delta_2]([\Delta_1]\omega)$.

  We already know that $\widehat{\beta_1} \in \mathsf{fkv}([\Delta_1]\omega)$ and $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2]\widehat{\beta_1})$, so we know $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2]([\Delta_1]\omega))$ and we are done.

  (b) there exists $\widehat{\beta_3} \in \mathsf{unsolved}(\Delta)$ such that $\widehat{\beta_1} \in \mathsf{fkv}([\Delta_1]\widehat{\beta_3})$.

  Similar as the previous subcase, we have $[\Delta_2, \mathsf{unsolved}(\Delta_3)]\widehat{\beta_3} = [\Delta_2]\widehat{\beta_3} = [\Delta_2]([\Delta_1]\widehat{\beta_3})$.

  We already know that $\widehat{\beta_1} \in \mathsf{fkv}([\Delta_1]\widehat{\beta_3})$ and $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2]\widehat{\beta_1})$, so we know $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2]([\Delta_1]\widehat{\beta_3}))$ and we are done.

- The case for rule A-KTT-FORALLI is similar as the previous case.
- Case

  A-KTT-APP
  $$\frac{\Delta \Vdash^{k} \tau_1 : \eta_1 \leadsto \rho_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{kapp} (\rho_1 : [\Delta_1]\eta_1) \bullet \tau_2 : \omega \leadsto \rho \dashv \Theta}{\Delta \Vdash^{k} \tau_1\ \tau_2 : \omega \leadsto \rho \dashv \Theta}$$

  Given $\widehat{\alpha} \in \mathsf{unsolved}(\Theta)$, by the lemma on application kinding part we have two cases.

(1) $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\rho)$. Then the goal follows directly.

(2) there exists $\widehat{\beta}_1 \in \mathsf{unsolved}(\Delta_1)$ such that $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\widehat{\beta}_1)$.
Because $\widehat{\beta}_1 \in \mathsf{unsolved}(\Delta_1)$, by I.H., we have two subcases.

(a) $\widehat{\beta}_1 \in \mathsf{fkv}([\Delta_1]\rho_1)$.
Then by lemma F.23 we have $[\Theta]\rho_1 = [\Theta]([\Delta_1]\rho_1)$.
We already know that $\widehat{\beta}_1 \in \mathsf{fkv}([\Delta_1]\rho_1)$ and $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\widehat{\beta}_1)$ so we must have $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\rho_1)$.
By the lemma on application kinding, we have $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\rho)$, so we are done.

(b) there exists $\widehat{\beta}_2 \in \mathsf{unsolved}(\Delta_1)$, such that $\widehat{\beta}_1 \in \mathsf{fkv}([\Delta_1]\widehat{\beta}_2)$
By lemma F.23 we have $[\Theta]\widehat{\beta}_2 = [\Theta]([\Delta_1]\widehat{\beta}_2)$.
And we must have $\widehat{\alpha} \in \mathsf{fkv}([\Theta]([\Delta_1]\widehat{\beta}_2))$.

- The case for rules A-KTT-KAPP and A-KTT-KAPP-INFER is similar as the previous case.

Instantiation The statement for instantiation is: if $\Delta \Vdash^{\mathsf{inst}} \mu_1 : \eta_1 \sqsubseteq \eta_2 \rightsquigarrow \mu_2 \dashv \Theta$, then for any $\widehat{\alpha} \in \mathsf{unsolved}(\Theta)$, either $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\mu_2)$, or there exists $\widehat{\beta} \in \mathsf{unsolved}(\Delta)$, such that $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\widehat{\beta})$. Moreover, μ_2 contains all the unification variables in μ_1.
We prove it by induction on the derivation.

- Case

A-INST-REFL
$$\frac{\Delta \Vdash^{\mathsf{u}} \omega_1 \approx \omega_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{inst}} \mu : \omega_1 \sqsubseteq \omega_2 \rightsquigarrow \mu \dashv \Theta}$$

The first half of the goal follows directly from the lemma on unification part, and the second goal holds trivially.

- Case

A-INST-FORALL
$$\frac{\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{inst}} \mu_1\ @\widehat{\alpha} : \eta[a \mapsto \widehat{\alpha}] \sqsubseteq \omega_2 \rightsquigarrow \mu_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{inst}} \mu_1 : \forall a : \omega_1.\eta \sqsubseteq \omega_2 \rightsquigarrow \mu_2 \dashv \Theta}$$

The second half of the goal follows directly from I.H.. Given $\widehat{\alpha}_1 \in \mathsf{unsolved}(\Theta)$, by I.H., we have two cases.

(1) $\widehat{\alpha}_1 \in \mathsf{fkv}([\Theta]\mu_2)$. So the first half of the goal holds directly.

(2) there exists $\widehat{\beta} \in \mathsf{unsolved}(\Delta, \widehat{\alpha} : \omega_1)$, such that $\widehat{\alpha}_1 \in \mathsf{fkv}([\Theta]\widehat{\beta})$.
Then we have either $\widehat{\beta} = \widehat{\alpha}$, or $\widehat{\beta} \in \mathsf{unsolved}(\Delta)$. In the former case, as $\mu_1\ @\widehat{\alpha}$ contains $\widehat{\alpha}$, we have μ_2 contains $\widehat{\alpha}$. Therefore $\widehat{\alpha}_1 \in \mathsf{fkv}([\Theta]\mu_2)$ and we are done. In the latter case, the goal follows directly.

- The case for rule A-INST-FORALL-INFER is similar as the previous case.

Application Kinding The statement for application kinding is: if $\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \eta) \bullet \tau : \omega \rightsquigarrow \rho_2 \dashv \Theta$, then for any $\widehat{\alpha} \in \mathsf{unsolved}(\Theta)$, either $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\rho_2)$, or there exists $\widehat{\beta} \in \mathsf{unsolved}(\Delta)$, such that $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\widehat{\beta})$. Moreover, ρ_2 contains all the unification variables in ρ_1.
We prove it by induction on the derivation.

- Case

A-KAPP-TT-ARROW
$$\frac{\Delta \Vdash^{\mathsf{kc}} \tau \Leftarrow \omega_1 \rightsquigarrow \rho_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \omega_1 \ \rightarrow\ \omega_2) \bullet \tau : \omega_2 \rightsquigarrow \rho_1\, \rho_2 \dashv \Theta}$$

The first half of the goal follows directly from the lemma on kind checking part.
The second half of the goal holds trivially.

- Case

$$\frac{\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{kapp}} (\rho_1 \ @\widehat{\alpha} : \eta[a \mapsto \widehat{\alpha}]) \bullet \tau : \omega \leadsto \rho \dashv \Theta}{\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \forall a : \omega_1 . \eta) \bullet \tau : \omega \leadsto \rho \dashv \Theta}\ \textsc{a-kapp-tt-forall}$$

The second half of the goal follows directly from I.H..
Given $\widehat{\alpha}_1 \in \mathsf{unsolved}(\Theta)$, by I.H., we have two cases.
(1) $\widehat{\alpha}_1 \in \mathsf{fkv}([\Theta]\rho)$. So the first half of the goal holds directly.
(2) there exists $\widehat{\beta} \in \mathsf{unsolved}(\Delta, \widehat{\alpha} : \omega_1)$, such that $\widehat{\alpha}_1 \in \mathsf{fkv}([\Theta]\widehat{\beta})$.
Then we have either $\widehat{\beta} = \widehat{\alpha}$, or $\widehat{\beta} \in \mathsf{unsolved}(\Delta)$. In the former case, as $\rho_1 \ @\widehat{\alpha}$ contains $\widehat{\alpha}$, we have ρ contains $\widehat{\alpha}$. Therefore $\widehat{\alpha}_1 \in \mathsf{fkv}([\Theta]\rho)$ and we are done. In the latter case, the goal follows directly.

- The case for rule A-KAPP-TT-FORALL-INFER is the same as previous case.
- Case

$$\frac{\Delta_1, \widehat{\alpha}_1 : \star, \widehat{\alpha}_2 : \star, \widehat{\alpha} : \omega = (\widehat{\alpha}_1 \ \to \ \widehat{\alpha}_2), \Delta_2 \Vdash^{\mathsf{kc}} \tau \Leftarrow \widehat{\alpha}_1 \leadsto \rho_2 \dashv \Theta}{\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \Vdash^{\mathsf{kapp}} (\rho_1 : \widehat{\alpha}) \bullet \tau : \widehat{\alpha}_2 \leadsto \rho_1\, \rho_2 \dashv \Theta}\ \textsc{a-kapp-tt-kuvar}$$

The second half of the goal follows trivially.
Given $\widehat{\alpha}_3 \in \mathsf{unsolved}(\Theta)$, by I.H., we have two cases.
(1) $\widehat{\alpha}_3 \in \mathsf{fkv}([\Theta]\rho_2)$. So the first half of the goal holds directly.
(2) there exists $\widehat{\beta} \in \mathsf{unsolved}(\Delta_1, \widehat{\alpha}_1 : \star, \widehat{\alpha}_2 : \star, \widehat{\alpha} : \omega = (\widehat{\alpha}_1 \ \to \ \widehat{\alpha}_2), \Delta_2)$, such that $\widehat{\alpha}_3 \in \mathsf{fkv}([\Theta]\widehat{\beta})$.
Then we have either $\widehat{\beta} = \widehat{\alpha}_1$, or $\widehat{\beta} = \widehat{\alpha}_2$, or $\widehat{\beta} \in \mathsf{unsolved}(\Delta_1, \widehat{\alpha} : \omega = \widehat{\alpha}_1 \ \to \ \widehat{\alpha}_2, \Delta_2)$. In the former two cases, we pick $\widehat{\alpha}$ from the input context. And $[\Theta]\widehat{\alpha} = [\Theta]([\Delta_1, \widehat{\alpha}_1 : \star, \widehat{\alpha}_2 : \star, \widehat{\alpha} : \kappa = (\widehat{\alpha}_1 \ \to \ \widehat{\alpha}_2), \Delta_2]\widehat{\alpha}) = [\Theta](\widehat{\alpha}_1 \ \to \ \widehat{\alpha}_2)$ by Lemma F.23. Therefore $\widehat{\alpha}_3 \in \mathsf{fkv}([\Theta]\widehat{\alpha})$.
In the later case, then it must be $\widehat{\beta} \in \mathsf{unsolved}(\Delta_1, \widehat{\alpha} : \omega, \Delta_2)$ So we are done.

Kind Checking The statement for kind checking is: if $\Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \eta \leadsto \mu \dashv \Theta$, then for any $\widehat{\alpha} \in \mathsf{unsolved}(\Theta)$, either $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\mu)$, or there exists $\widehat{\beta} \in \mathsf{unsolved}(\Delta)$, such that $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\widehat{\beta})$.
To prove the lemma, we have

$$\frac{\Delta \Vdash^{\mathsf{k}} \sigma : \eta \leadsto \mu_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{inst}} \mu_1 : [\Delta_1]\eta \sqsubseteq [\Delta_1]\omega \leadsto \mu_2 \dashv \Delta_2}{\Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \omega \leadsto \mu_2 \dashv \Delta_2}\ \textsc{a-kc-sub}$$

Given $\widehat{\alpha} \in \mathsf{unsolved}(\Delta_2)$, by the lemma on the instantiation part, we have two cases.
(1) $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2]\mu_2)$. Then the goal follows directly.
(2) there exists $\widehat{\beta} \in \mathsf{unsolved}(\Delta_1)$, such that $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2]\widehat{\beta})$. Then because $\widehat{\beta} \in \mathsf{unsolved}(\Delta_1)$, by the lemma on the kinding part, we have two subcases.
(a) $\widehat{\beta} \in \mathsf{fkv}([\Delta_1]\mu_1)$. Then by the lemma on the instantiation part, we know that $\widehat{\beta} \in \mathsf{fkv}([\Delta_1]\mu_2)$. By lemma F.23, we have $[\Delta_2]\mu_2 = [\Delta_2]([\Delta_1]\mu_2)$. So we have $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2]([\Delta_1]\mu_2))$.
(2) there exists $\widehat{\beta}_2 \in \mathsf{unsolved}(\Delta)$, such that $\widehat{\beta} \in \mathsf{fkv}([\Delta_1]\widehat{\beta}_2)$. By lemma F.23, we have $[\Delta_2]\widehat{\beta}_2 = [\Delta_2]([\Delta_1](\widehat{\beta}_2))$. So we have $\widehat{\alpha} \in \mathsf{fkv}([\Delta_2]([\Delta_1]\widehat{\beta}_2))$.

Promotion The statement for promotion is: if $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \leadsto \omega_2 \dashv \Theta$, then for any $\widehat{\alpha}' \in \mathsf{unsolved}(\Theta)$, there exists $\widehat{\beta} \in \mathsf{unsolved}(\Delta)$, such that $\widehat{\alpha}' \in \mathsf{fkv}([\Theta]\widehat{\beta})$.
The only interesting case here is

a-pr-kuvarR-tt

$$\frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} [\Delta]\rho \leadsto \rho_1 \dashv \Theta[\widehat{\alpha}][\widehat{\beta} : \rho]}{\Delta[\widehat{\alpha}][\widehat{\beta} : \rho] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \widehat{\beta} \leadsto \widehat{\beta}_1 \dashv \Theta[\widehat{\beta}_1 : \rho_1, \widehat{\alpha}][\widehat{\beta} : \rho = \widehat{\beta}_1]}$$

Given $\widehat{\alpha}' \in \mathsf{unsolved}(\Theta[\widehat{\beta}_1 : \rho_1, \widehat{\alpha}][\widehat{\beta} : \rho = \widehat{\beta}_1])$, we have two cases:

- $\widehat{\alpha}'$ is not $\widehat{\beta}_1$.
  Then we have $\widehat{\alpha}' \in \mathsf{unsolved}(\Theta[\widehat{\alpha}][\widehat{\beta} : \rho])$, and by I.H. we are done.
- $\widehat{\alpha}'$ is $\widehat{\beta}_1$.
  Then we pick $\widehat{\beta}$ from the input context, and we have that $[\Theta[\widehat{\beta}_1 : \rho_1, \widehat{\alpha}][\widehat{\beta} : \rho = \widehat{\beta}_1]]\widehat{\beta} = \widehat{\beta}_1$ so we are done.

Unification The statement for unification is: if $\Delta \Vdash^{\mathsf{u}} \omega_1 \approx \omega_2 \dashv \Theta$, then for any $\widehat{\alpha} \in \mathsf{unsolved}(\Theta)$, there exists $\widehat{\beta} \in \mathsf{unsolved}(\Delta)$, such that $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\widehat{\beta})$.

Here, all cases are essentially the same. We discuss two of them and the rest can be proved in a similar way.

- Case

  a-u-app

  $$\frac{\Delta \Vdash^{\mathsf{u}} \rho_1 \approx \rho_3 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{u}} [\Delta_1]\rho_2 \approx [\Delta_1]\rho_4 \dashv \Theta}{\Delta \Vdash^{\mathsf{u}} \rho_1\,\rho_2 \approx \rho_3\,\rho_4 \dashv \Theta}$$

  Given $\widehat{\alpha} \in \mathsf{unsolved}(\Theta)$, by I.H., we know that there exists $\widehat{\beta} \in \mathsf{unsolved}(\Delta_1)$, such that $\widehat{\alpha} \in \mathsf{fkv}([\Theta]\widehat{\beta})$.
  And because $\widehat{\beta} \in \mathsf{unsolved}(\Delta_1)$, by I.H., we know that there exists $\widehat{\beta}_2 \in \mathsf{unsolved}(\Delta)$, such that $\widehat{\beta} \in \mathsf{fkv}([\Delta_1]\widehat{\beta}_2)$.
  By lemma F.23 we know that $[\Theta]\widehat{\beta}_2 = [\Theta]([\Delta_1]\widehat{\beta}_2)$. So we must have $\widehat{\alpha} \in \mathsf{fkv}([\Theta]([\Delta_1]\widehat{\beta}_2))$.
- Case

  a-u-kvarL-lo-tt

  $$\frac{\begin{array}{cc}\Delta_1, \Delta_2 \mathrel{+\!\!\!+}^{\mathsf{mv}} \widehat{\alpha} : \omega_1 \rightsquigarrow \Theta & \Delta[\{\Theta\}] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_1 \leadsto \rho_2 \dashv \Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4 \\ \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{ela}} \rho_2 : \omega_2 & \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{u}} [\Theta_1, \Theta_2]\omega_1 \approx \omega_2 \dashv \Theta_5, \{\Theta_6\}\end{array}}{\Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}] \Vdash^{\mathsf{u}} \widehat{\alpha} \approx \rho_1 \dashv \Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4}$$

  Given $\widehat{\alpha}_1 \in \mathsf{unsolved}(\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4)$, we have two cases to discuss.
  (1) $\widehat{\alpha}_1 \in \mathsf{unsolved}(\Theta_5, \{\Theta_6\})$.
  Then by I.H., we know that there is $\widehat{\beta}_1 \in \mathsf{unsolved}(\Theta_1, \{\Theta_2\})$ such that $\widehat{\alpha}_1 \in \mathsf{fkv}([\Theta_5, \{\Theta_6\}]\widehat{\beta}_1)$.
  By the definition, we know that $\widehat{\beta}_1 \in \mathsf{unsolved}(\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4)$.
  Then by the lemma on the promotion part, we know that there exists a $\widehat{\beta}_2 \in \mathsf{unsolved}(\Delta[\{\Theta\}])$ such that $\widehat{\beta}_1 \in \mathsf{fkv}([\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4]\widehat{\beta}_2)$.
  By the definition of moving, we know that all unsolved unification in $\mathsf{unsolved}(\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2)$ are in $\mathsf{unsolved}(\Theta)$. Therefore we have $\widehat{\beta}_2 \in \mathsf{unsolved}(\Delta[\{\Theta\}])$.
  We have that $[\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4]\widehat{\beta}_2 = [\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4]([\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4]\widehat{\beta}_2)$.
  Then we must have $\widehat{\alpha}_1 \in \mathsf{fkv}([\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4]([\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4]\widehat{\beta}_2))$.
  (2) $\widehat{\alpha}_1$ is in the domain of Θ_3 and Θ_4.
  Then it must be in $\mathsf{unsolved}(\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4)$.
  Then by the lemma on the promotion part, we know that there exists a $\widehat{\beta} \in \mathsf{unsolved}(\Delta[\{\Theta\}])$ such that $\widehat{\alpha}_1 \in \mathsf{fkv}([\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4]\widehat{\beta})$.

By moving, we know that all unsolved unification in $\mathsf{unsolved}(\Delta[\{\Theta\}])$ are in $\mathsf{unsolved}(\Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}])$.
Therefore we have $\widehat{\beta} \in \mathsf{unsolved}(\Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}])$.
We have that $[\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4]\widehat{\beta} = [\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4]([\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4]\widehat{\beta})$.
Then we must have $\widehat{\alpha}_1 \in \mathsf{fkv}([\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4]([\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4]\widehat{\beta}))$.

□

F.2.6 Soundness of Algorithm.

Lemma F.46 (Soundness of Promotion). *If* Δ ok, *and* $[\Delta]\omega_1 = \omega_1$, *and* $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \omega_1 \rightsquigarrow \omega_2 \dashv \Theta$, *then* $[\Theta]\omega_1 = [\Theta]\omega_2 = \omega_2$. *If* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\omega_1 = [\Omega]\omega_2$.

Proof. The first half follows directly from a straightforward induction on promotion.
The second half of the goal follows directly from and lemma F.23.

□

Lemma F.47 (Soundness of Unification). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{u}} \omega_1 \approx \omega_2 \dashv \Theta$, *then* $[\Theta]\omega_1 = [\Theta]\omega_2$. *If* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\omega_1 = [\Omega]\omega_2$.

Proof. By lemma F.23, we only need to prove the first half of the lemma.
The case for rule a-u-refl-tt holds trivially. And the case for rule a-u-app and rule a-u-kapp follows from I.H. and lemma F.23. As rule a-u-kvarL-tt and rule a-u-kvarR-tt, rule a-u-kvarL-lo-tt and rule a-u-kvarR-lo-tt are symmetric, we only prove one of them.

- Case

a-u-kvarL-tt

$$\frac{\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta_1, \widehat{\alpha} : \omega_1, \Theta_2 \qquad \Theta_1 \Vdash^{\mathsf{ela}} \rho_2 : \omega_2 \qquad \Theta_1 \Vdash^{\mathsf{u}} [\Theta_1]\omega_1 \approx \omega_2 \dashv \Theta_3}{\Delta \Vdash^{\mathsf{u}} \widehat{\alpha} \approx \rho_1 \dashv \Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2}$$

$\Theta_1, \widehat{\alpha} : \omega_1, \Theta_2 \longrightarrow \Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2$	We have proved in lemma F.7
$[\Theta_1, \widehat{\alpha} : \omega_1, \Theta_2]\rho_1 = [\Theta_1, \widehat{\alpha} : \omega_1, \Theta_2]\rho_2$	By Lemma F.46
$[\Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2]\rho_1 = [\Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2]\rho_2$	By Lemma F.23
$[\Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2]\widehat{\alpha} = [\Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2]\rho_2$	By definition
$[\Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2]\widehat{\alpha} = [\Theta_3, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_2]\rho_1$	By equations

- Case

a-u-kvarL-lo-tt

$$\frac{\begin{array}{c}\Delta_1, \Delta_2 \mathrel{+\!\!\!+}^{\mathsf{mv}} \widehat{\alpha} : \omega_1 \rightsquigarrow \Theta \qquad \Delta[\{\Theta\}] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_1 \rightsquigarrow \rho_2 \dashv \Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4 \\ \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{ela}} \rho_2 : \omega_2 \qquad \Theta_1, \{\Theta_2\} \Vdash^{\mathsf{u}} [\Theta_1, \Theta_2]\omega_1 \approx \omega_2 \dashv \Theta_5, \{\Theta_6\}\end{array}}{\Delta[\{\Delta_1, \widehat{\alpha} : \omega_1, \Delta_2\}] \Vdash^{\mathsf{u}} \widehat{\alpha} \approx \rho_1 \dashv \Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4}$$

$\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4 \longrightarrow \Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4$	We have proved in lemma F.7
$[\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4]\rho_1 = [\Theta_1, \{\Theta_2, \widehat{\alpha} : \omega_1, \Theta_3\}, \Theta_4]\rho_2$	By Lemma F.46
$[\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4]\rho_1 = [\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4]\rho_2$	By lemma F.23
$[\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4]\widehat{\alpha} = [\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4]\rho_2$	By definition
$[\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4]\rho_1 = [\Theta_5, \{\Theta_6, \widehat{\alpha} : \omega_1 = \rho_2, \Theta_3\}, \Theta_4]\widehat{\alpha}$	By equations

□

Lemma F.48 (Soundness of Instantiation). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{ela}} \mu_1 : \eta$, *and* $\Delta \Vdash^{\mathsf{ela}} \omega : \star$, *and* $\Delta \Vdash^{\mathsf{inst}} \mu_1 : \eta \sqsubseteq \omega \rightsquigarrow \mu_2 \dashv \Theta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{\mathsf{inst}} [\Omega]\mu_1 : [\Omega]\eta \sqsubseteq [\Omega]\omega \rightsquigarrow [\Omega]\mu_2$.

Proof. By induction on the derivation. The case for rule A-INST-REFL follows from soundness of unification (lemma F.47) and rule INST-REFL. The rest cases follows from I.H..

□

Lemma F.49 (Soundness of Kinding). *If* Δ ok, *we have*

- *if* $\Delta \Vdash^{k} \sigma : \eta \rightsquigarrow \mu \dashv \Theta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{k} \sigma : [\Omega]\eta \rightsquigarrow [\Omega]\mu$;
- *if* $\Delta \Vdash^{kc} \sigma \Leftarrow \eta \rightsquigarrow \mu \dashv \Theta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{kc} \sigma \Leftarrow [\Omega]\eta \rightsquigarrow [\Omega]\mu$.
- *if* $\Delta \Vdash^{kapp} (\rho_1 : \eta) \bullet \tau : \omega \rightsquigarrow \rho_2 \dashv \Theta$, *and* $\Delta \Vdash^{ela} \rho_1 : \eta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{inst} [\Omega]\rho_1 : [\Omega]\eta \sqsubseteq (\omega_1 \rightarrow [\Omega]\omega) \rightsquigarrow \rho_3$, *and* $[\Omega]\Delta \vdash^{kc} \tau \Leftarrow \omega_1 \rightsquigarrow \rho_4$. *and* $[\Omega]\rho_2 = \rho_3\, \rho_4$.

Proof. By induction on the derivation.

Part 1 • The case for rules A-KTT-STAR, A-KTT-KUVAR, A-KTT-NAT, A-KTT-VAR, A-KTT-TCON, and A-KTT-ARROW are straightforward.

- Case

$$\frac{\Delta \Vdash^{k} \tau_1 : \eta_1 \rightsquigarrow \rho_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{kapp} (\rho_1 : [\Delta_1]\eta_1) \bullet \tau_2 : \omega \rightsquigarrow \rho \dashv \Theta}{\Delta \Vdash^{k} \tau_1\, \tau_2 : \omega \rightsquigarrow \rho \dashv \Theta}\ \text{A-KTT-APP}$$

By well-formedness of the judgments, we know every output context is an extension of the input context and by transitivity we have that output context is an extension of all the previous input contexts.

$$\begin{array}{l|l}
[\Omega]\Delta \vdash^{k} \tau_1 : [\Omega]\eta_1 \rightsquigarrow [\Omega]\rho_1 & \text{I.H.} \\
\Delta_1 \Vdash^{ela} \rho_1 : [\Delta_1]\eta_1 & \text{By Lemma F.10} \\
[\Omega]\Delta_1 \vdash^{inst} [\Omega]\rho_1 : [\Omega]([\Delta_1]\eta_1) \sqsubseteq \omega_1 \rightarrow [\Omega]\omega \rightsquigarrow \rho_2 & \text{By Part 3} \\
\wedge [\Omega]\Delta_1 \vdash^{kc} \tau_2 \Leftarrow \omega_1 \rightsquigarrow \rho_3 & \\
\wedge [\Omega]\rho = \rho_2\, \rho_3 & \\
[\Omega]\Delta = [\Omega]\Delta_1 & \text{By Lemma F.35} \\
[\Omega]\Delta \vdash^{inst} [\Omega]\rho_1 : [\Omega]\eta_1 \sqsubseteq \omega_1 \rightarrow [\Omega]\omega \rightsquigarrow \rho_2 & \text{By equations and Lemma F.23} \\
[\Omega]\Delta \vdash^{kc} \tau_2 \Leftarrow \omega_1 \rightsquigarrow \rho_3 & \text{By equations} \\
[\Omega]\Delta \vdash^{k} \tau_1\, \tau_2 : [\Omega]\omega \rightsquigarrow [\Omega]\rho & \text{By rule KTT-APP}
\end{array}$$

- Case

$$\frac{\Delta \Vdash^{k} \tau_1 : \eta \rightsquigarrow \rho_1 \dashv \Delta_1 \qquad [\Delta_1]\eta = \forall a : \omega.\eta_2 \qquad \Delta_1 \Vdash^{kc} \tau_2 \Leftarrow \omega \rightsquigarrow \rho_2 \dashv \Delta_2}{\Delta \Vdash^{k} \tau_1\, @\tau_2 : \eta_2[a \mapsto \rho_2] \rightsquigarrow \rho_1\, @\rho_2 \dashv \Delta_2}\ \text{A-KTT-KAPP}$$

By well-formedness of the judgments, we know every output context is an extension of the input context and by transitivity we have that output context is an extension of all the previous input contexts.

$$\begin{array}{l|l}
[\Omega]\Delta \vdash^{k} \tau_1 : [\Omega]\eta \rightsquigarrow [\Omega]\rho_1 & \text{I.H.} \\
[\Omega]\eta = [\Omega]([\Delta_1]\eta) = \forall a : [\Omega]\omega.[\Omega]\eta_2. & \text{By lemma F.23} \\
[\Omega]\Delta \vdash^{k} \tau_1 : \forall a : [\Omega]\omega.[\Omega]\eta_2 \rightsquigarrow [\Omega]\rho_1 & \text{By equations} \\
[\Omega]\Delta_1 \vdash^{kc} \tau_2 \Leftarrow [\Omega]\omega \rightsquigarrow [\Omega]\rho_2 & \text{By Part 2} \\
[\Omega]\Delta \vdash^{kc} \tau_2 \Leftarrow [\Omega]\omega \rightsquigarrow [\Omega]\rho_2 & \text{By lemma F.35} \\
[\Omega]\Delta \vdash^{k} \tau_1\, @\tau_2 : ([\Omega]\eta_2)[a \mapsto [\Omega]\omega] \rightsquigarrow [\Omega]\rho_1\, @[\Omega]\rho_2 & \text{By rule KTT-KAPP} \\
[\Omega]\Delta \vdash^{k} \tau_1\, @\tau_2 : [\Omega](\eta_2[a \mapsto [\Omega]\omega]) \rightsquigarrow [\Omega](\rho_1\, @\rho_2) & \text{By substitutions}
\end{array}$$

- The case for rule A-KTT-KAPP-INFER is similar to the previous case.

- Case

$$\frac{\Delta \Vdash^{\mathsf{kc}} \kappa \Leftarrow \star \rightsquigarrow \omega \dashv \Delta_1 \qquad \Delta_1, a : \omega \Vdash^{\mathsf{kc}} \sigma \Leftarrow \star \rightsquigarrow \mu \dashv \Delta_2, a : \omega, \Delta_3 \qquad \Delta_3 \hookrightarrow a}{\Delta \Vdash^{\mathsf{k}} \forall a : \kappa.\sigma : \star \rightsquigarrow \forall a : \omega.[\Delta_3]\mu \dashv \Delta_2, \mathsf{unsolved}(\Delta_3)}\ \textsc{a-ktt-forall}$$

$$\begin{array}{l|l}
\Delta_1, a : \omega \longrightarrow \Delta_2, a : \omega, \Delta_3 & \text{lemma F.10} \\
\Delta_1 \longrightarrow \Delta_2 & \text{By Lemma F.22} \\
\Delta_2 \longrightarrow \Delta_2, \mathsf{unsolved}(\Theta) & \text{rule } \textsc{a-ctxe-add} \\
\Delta_2, \mathsf{unsolved}(\Delta_3) \longrightarrow \Omega & \text{Given} \\
\Delta_1 \longrightarrow \Omega & \text{by lemma F.24} \\
[\Omega]\Delta \vdash^{\mathsf{kc}} \kappa \Leftarrow \star \rightsquigarrow [\Omega]\omega & \text{By Part 2} \\
\Delta_2 \longrightarrow \Omega_1 \wedge \Omega = \Omega_1, \Omega_2 \wedge \Omega_2 \textbf{ soft} & \text{By Lemma F.22} \\
\Delta_2, a : \omega \longrightarrow \Omega_1, a : \omega & \text{By rule } \textsc{a-ctxe-tvar-tt} \\
\text{Construct a } \Omega_3 \text{ such that} & \\
\Delta_2, a : \omega, \Delta_3 \longrightarrow \Omega_1, a : \omega, \Omega_3 & \\
\wedge \forall \widehat{\alpha} \in \mathsf{unsolved}(\Delta_3), [\Omega_3]\widehat{\alpha} = [\Omega_2]\widehat{\alpha} & \\
[\Omega_1, a : \omega, \Omega_3](\Delta_2, a : \omega, \Delta_3) \vdash^{\mathsf{kc}} \sigma \Leftarrow \star \rightsquigarrow [\Omega_1, a : \omega, \Omega_3]\mu & \text{Part 2} \\
[\Omega_1]\Delta_2, a : [\Omega_1]\omega \vdash^{\mathsf{kc}} \sigma \Leftarrow \star \rightsquigarrow [\Omega_1, a : \omega, \Omega_3]\mu & \text{Lemma F.34 and definition} \\
[\Omega_1]\Delta_2, a : [\Omega_1]\omega \vdash^{\mathsf{kc}} \sigma \Leftarrow \star \rightsquigarrow [\Omega_1, \Omega_2]([\Delta_3]\mu) & \\
[\Omega_1]\Delta, a : [\Omega_1]\omega \vdash^{\mathsf{kc}} \sigma \Leftarrow \star \rightsquigarrow [\Omega]([\Delta_3]\mu) & \text{Lemma F.35} \\
[\Omega]\Delta, a : [\Omega_1]\omega \vdash^{\mathsf{kc}} \sigma \Leftarrow \star \rightsquigarrow [\Omega]([\Delta_3]\mu) & \text{definition} \\
[\Omega]\Delta, a : [\Omega]\omega \vdash^{\mathsf{kc}} \sigma \Leftarrow \star \rightsquigarrow [\Omega]([\Delta_3]\mu) & \text{Lemma F.36} \\
[\Omega]\Delta \vdash^{\mathsf{k}} \forall a : \kappa.\sigma : \star \rightsquigarrow \forall a : [\Omega]\omega.[\Omega]([\Delta_3]\mu) & \text{By rule } \textsc{ktt-forall} \\
[\Omega]\Delta \vdash^{\mathsf{k}} \forall a : \kappa.\sigma : \star \rightsquigarrow [\Omega](\forall a : \omega.([\Delta_3]\mu)) & \text{By substitution}
\end{array}$$

- The case for rule A-KTT-FORALLI is similar to the previous case.
  The notable thing is that we use the solution of $\widehat{\alpha}$ (as in the rule A-KTT-FORALLI) in Ω as the ω in rule KTT-FORALL. rule KTT-FORALL.

Part 2 We have

$$\frac{\Delta \Vdash^{\mathsf{k}} \sigma : \eta \rightsquigarrow \mu_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{inst}} \mu_1 : [\Delta_1]\eta \sqsubseteq [\Delta_1]\omega \rightsquigarrow \mu_2 \dashv \Delta_2}{\Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \omega \rightsquigarrow \mu_2 \dashv \Delta_2}\ \textsc{a-kc-sub}$$

Follows directly from Part 1 and soundness of instantiation (lemma F.48).

Part 3 By induction on the judgment.

- Case

$$\frac{\Delta \Vdash^{\mathsf{kc}} \tau \Leftarrow \omega_1 \rightsquigarrow \rho_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \omega_1 \rightarrow \omega_2) \bullet \tau : \omega_2 \rightsquigarrow \rho_1\ \rho_2 \dashv \Theta}\ \textsc{a-kapp-tt-arrow}$$

$$\begin{array}{l|l}
[\Omega]\Delta \vdash^{\mathsf{inst}} [\Omega]\rho_1 : [\Omega]\omega_1 \rightarrow [\Omega]\omega_2 \sqsubseteq [\Omega]\omega_1 \rightarrow [\Omega]\omega_2 \rightsquigarrow [\Omega]\rho_1 & \text{By rule } \textsc{inst-refl} \\
[\Omega]\Delta \vdash^{\mathsf{kc}} \tau \Leftarrow [\Omega]\omega_1 \rightsquigarrow [\Omega]\rho_2 & \text{Part 2}
\end{array}$$

- Case

$$\frac{\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{kapp}} (\rho_1\ @\widehat{\alpha} : \eta[a \mapsto \widehat{\alpha}]) \bullet \tau : \omega \rightsquigarrow \rho \dashv \Theta}{\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \forall a : \omega_1.\eta) \bullet \tau : \omega \rightsquigarrow \rho \dashv \Theta}\ \textsc{a-kapp-tt-forall}$$

  Follows directly from I.H. and rule INST-FORALL.

- Case

$$\frac{\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{kapp}} (\rho_1 \ @\widehat{\alpha} : \eta[a \mapsto \widehat{\alpha}]) \bullet \tau : \omega \rightsquigarrow \rho \dashv \Theta}{\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \forall\{a : \omega_1\}.\eta) \bullet \tau : \omega \rightsquigarrow \rho \dashv \Theta}\ \textsc{A-kapp-tt-forall-infer}$$

  Follows directly from I.H. and rule INST-FORALL-INFER.

- Case

$$\frac{\Delta_1, \widehat{\alpha}_1 : \star, \widehat{\alpha}_2 : \star, \widehat{\alpha} : \omega = (\widehat{\alpha}_1 \ \rightarrow \ \widehat{\alpha}_2), \Delta_2 \Vdash^{\mathsf{kc}} \tau \Leftarrow \widehat{\alpha}_1 \rightsquigarrow \rho_2 \dashv \Theta}{\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \Vdash^{\mathsf{kapp}} (\rho_1 : \widehat{\alpha}) \bullet \tau : \widehat{\alpha}_2 \rightsquigarrow \rho_1 \ \rho_2 \dashv \Theta}\ \textsc{A-kapp-tt-kuvar}$$

$$\begin{array}{l|l} [\Omega]\Delta \vdash^{\mathsf{inst}} [\Omega]\rho_1 : [\Omega]\widehat{\alpha}_1 \ \rightarrow \ [\Omega]\widehat{\alpha}_2 \sqsubseteq [\Omega]\widehat{\alpha}_1 \ \rightarrow \ [\Omega]\widehat{\alpha}_2 \rightsquigarrow [\Omega]\rho_1 & \text{By rule INST-REFL} \\ [\Omega]\Delta \vdash^{\mathsf{kc}} \tau \Leftarrow [\Omega]\widehat{\alpha}_1 \rightsquigarrow [\Omega]\rho_2 & \text{Part 2} \end{array}$$

□

Lemma F.50 (Soundness of Elaborated Kinding). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{ela}} \mu : \eta$, *and* $\Delta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{\mathsf{ela}} [\Omega]\mu : [\Omega]\eta$.

Proof. By a straightforward induction on the derivation.

□

Lemma F.51 (Soundness of Typing Signature). *If* Δ ok, *and* $\Omega \Vdash^{\mathsf{sig}} \mathcal{S} \rightsquigarrow T : \eta$, *then* $[\Omega]\Omega \vdash^{\mathsf{sig}} \mathcal{S} \rightsquigarrow T : \eta$.

Proof. We have

$$\frac{\rceil\sigma\lceil \qquad \overline{a_i}^{\,i} = \mathsf{fkv}(\sigma) \qquad \Omega, \{\overline{\widehat{\alpha}_i : \star, a_i : \widehat{\alpha}_i}^{\,i}\} \Vdash^{\mathsf{k}} \sigma : \star \rightsquigarrow \eta \dashv \Delta \\ \phi^{\mathsf{c}}_1 = \mathsf{scoped\_sort}(\overline{a_i : [\Delta]\widehat{\alpha}_i}^{\,i}) \qquad \widehat{\phi}^{\mathsf{c}}_2 = \mathsf{unsolved}(\Delta) \qquad \Delta \hookrightarrow \overline{a_i}^{\,i}}{\Omega \Vdash^{\mathsf{sig}} \mathbf{data}\ T : \sigma \rightsquigarrow T : \forall\{\phi^{\mathsf{c}}_2\}.((\forall\{\phi^{\mathsf{c}}_1\}.[\Delta]\eta)[\widehat{\phi}^{\mathsf{c}}_2 \mapsto \phi^{\mathsf{c}}_2])}\ \textsc{A-sig-tt}$$

From $\Delta \hookrightarrow \overline{a_i}^{\,i}$ we know that all unsolved unification variables in Θ do not depend on $\overline{a_i}^{\,i}$.

Given $\widehat{\phi}^{\mathsf{c}}_2 = \mathsf{unsolved}(\Delta)$, we further know that $\widehat{\phi}^{\mathsf{c}}_2$ only contains unsolved unification variable that do not depend on $\overline{a_i}^{\,i}$.

So ϕ^{c}_2 only contains type variables that do not depend on any unification variable or $\overline{a_i}^{\,i}$.

By weakening, we can add ϕ^{c}_2 into the kinding judgment, so we get $\Omega, \phi^{\mathsf{c}}_2, \{\overline{\widehat{\alpha}_i : \star, a_i : \widehat{\alpha}_i}^{\,i}\} \Vdash^{\mathsf{k}} \sigma : \star \rightsquigarrow \eta \dashv \Delta_1$, where Δ_1 is identical to Δ except for the presence of ϕ^{c}_2.

Now, we solve all unsolved unification variable in Δ_1 (i.e., the domain of $\widehat{\phi}^{\mathsf{c}}$) to its corresponding type variable in ϕ^{c}_2. We get a complete context Ω_1 and $\Delta_1 \longrightarrow \Omega_1$.

By by lemma F.10, we have $\Omega, \phi^{\mathsf{c}}_2, \{\overline{\widehat{\alpha}_i : \star, a_i : \widehat{\alpha}_i}^{\,i}\} \longrightarrow \Delta_1$.

So by Lemma F.24 we have $\Omega, \phi^{\mathsf{c}}_2, \{\overline{\widehat{\alpha}_i : \star, a_i : \widehat{\alpha}_i}^{\,i}\} \longrightarrow \Omega_1$.

By soundness of kinding (lemma F.49), we know that $[\Omega_1](\Omega, \phi^{\mathsf{c}}_2, \{\overline{\widehat{\alpha}_i : \star, a_i : \widehat{\alpha}_i}^{\,i}\}) \vdash^{\mathsf{k}} \sigma : \star \rightsquigarrow [\Omega_1]\eta$.

$[\Omega_1](\Omega, \phi^{\mathsf{c}}_2, \{\overline{\widehat{\alpha}_i : \star, a_i : \widehat{\alpha}_i}^{\,i}\}) = [\Omega]\Omega, \phi^{\mathsf{c}}_2, \phi^{\mathsf{c}}_3[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}_2]$, where ϕ^{c}_3 is a well-formed order of ϕ^{c}_1.

And $[\Omega_1]\eta = ([\Delta]\eta)[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}_2]$, because Δ contains all the solved unification variable in Ω_1 except for $\widehat{\phi}^{\mathsf{c}}$.

Namely, $[\Omega]\Omega, \phi^{\mathsf{c}}_2, \phi^{\mathsf{c}}_3[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}_2] \vdash^{\mathsf{k}} \sigma : \star \rightsquigarrow ([\Theta]\eta)[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}_2]$. By reordering the context while preserving well-formedness, we have $[\Omega]\Omega, \phi^{\mathsf{c}}_2, \phi^{\mathsf{c}}_1[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}_2] \vdash^{\mathsf{k}} \sigma : \star \rightsquigarrow ([\Theta]\eta)[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}_2]$.

By the kinding rule we can get $[\Omega]\Omega, \phi^{\mathsf{c}}_2 \vdash^{\mathsf{k}} \forall\{\phi^{\mathsf{c}}_1[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}_2]\}.\sigma : \star \rightsquigarrow \forall\{\phi^{\mathsf{c}}_1[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}_2]\}.([\Theta]\eta)[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}_2]$. By distribution of substitution we get $[\Omega]\Omega, \phi^{\mathsf{c}}_2 \vdash^{\mathsf{k}} \forall\{\phi^{\mathsf{c}}_1[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}_2]\}.\sigma : \star \rightsquigarrow (\forall\{\phi^{\mathsf{c}}_1\}.[\Theta]\eta)[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}_2]$.

To prove the rule, our goal is to prove all preconditions in

$$\textsc{sig-tt}\quad \frac{\rceil\sigma\lceil \qquad \phi \in \mathcal{Q}(\sigma) \qquad \phi^{\mathsf{c}} \in \mathcal{Q}(\eta) \qquad \Sigma, \phi^{\mathsf{c}} \vdash^{\mathsf{k}} \forall\{\phi\}.\sigma : \star \rightsquigarrow \eta}{\Sigma \vdash^{\mathsf{sig}} \mathbf{data}\ T : \sigma \rightsquigarrow T : \forall\{\phi^{\mathsf{c}}\}.\eta}$$

We have $\rceil\sigma\lceil$ as given. We claim that ϕ^{c}_1 fits ϕ, and ϕ^{c}_2 fits ϕ^{c}.

We first prove ϕ^{c}_1 fits ϕ. Because $\phi^{\mathsf{c}}_1 = \text{scoped\_sort}(\overline{a_i : [\Theta]\widehat{\alpha}_i}^{\,i})$, obviously ϕ^{c}_1 is one of the well-formed permutation of $\overline{a_i}^{\,i}$, namely the free kind binder of σ.

We then prove ϕ^{c}_2 fits ϕ^{c}. That requires us to prove that ϕ^{c}_2 is the free kind binder of $(\forall\{\phi^{\mathsf{c}}_1\}.[\Theta]\eta)[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}_2]$. Because $\widehat{\phi}^{\mathsf{c}} = \text{unsolved}(\Theta)$, by lemma F.45, we know every unsolved unification variable in $\widehat{\phi}^{\mathsf{c}}$ either appears in $[\Theta]\eta$, or appears in ϕ^{c}_1. For sure $[\Theta]\eta$ and ϕ^{c}_1 cannot contain more unsolved unification variable than $\widehat{\phi}^{\mathsf{c}}$ or otherwise it would be ill-formed. Namely, $\widehat{\phi}^{\mathsf{c}}$ *are* the free unification variables of $[\Theta]\eta$ and ϕ^{c}_1. By substituting $\widehat{\phi}^{\mathsf{c}}$ with ϕ^{c}_2, we know that ϕ^{c}_2 are the free kind binder in $(\forall\{\phi^{\mathsf{c}}_1\}.[\Theta]\eta)[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}_2]$.

By now we have proved all the preconditions and we conclude that $[\Omega]\Omega \vdash^{\mathsf{sig}} \mathcal{S} \rightsquigarrow T : \eta$.

□

Lemma F.52 (Soundness of Typing Data Constructor Decl.). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{dc}}_{\rho} \mathcal{D} \rightsquigarrow \mu \dashv \Theta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{\mathsf{dc}}_{([\Omega]\rho)} \mathcal{D} \rightsquigarrow [\Omega]\mu$.

Proof. We have

$$\textsc{a-dc-tt}\quad \frac{\Delta, \blacktriangleright_D \Vdash^{\mathsf{k}} \forall\phi.(\overline{\tau_i}^{\,i} \rightarrow \rho) : \star \rightsquigarrow \mu \dashv \Theta_1, \blacktriangleright_D, \Theta_2 \qquad \widehat{\phi}^{\mathsf{c}} = \text{unsolved}(\Theta_2)}{\Delta \Vdash^{\mathsf{dc}}_{\rho} \forall\phi.D\,\overline{\tau_i}^{\,i} \rightsquigarrow \forall\{\phi^{\mathsf{c}}\}.(([\Theta_2]\mu)[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}]) \dashv \Theta_1}$$

To prove our goal, we claim that ϕ^{c} fits the ϕ^{c} in

$$\textsc{dc-tt}\quad \frac{\phi^{\mathsf{c}} \in \mathcal{Q}(\mu\backslash_{\Sigma,\overline{\tau_i}^{\,i}}) \qquad \Sigma, \phi^{\mathsf{c}} \vdash^{\mathsf{k}} \forall\phi.\overline{\tau_i}^{\,i} \rightarrow \rho : \star \rightsquigarrow \mu}{\Sigma \vdash^{\mathsf{dc}}_{\rho} \forall\phi.D\,\overline{\tau_i}^{\,i} \rightsquigarrow \forall\{\phi^{\mathsf{c}}\}.\mu}$$

We prove this by lemma F.45 and the similar reasoning as in lemma F.51.

The important thing to note is $\widehat{\phi}^{\mathsf{c}}$ only contains unsolved unification variables in μ.

Note that μ might contain unsolved unification variables in Θ_1. Then they must be the dependency of unsolved unification variables in Δ. And those are not unification variables that we should generalize over.

□

Lemma F.53 (Soundness of Typing Datatype Decl.). *If* Δ ok, *and* $\Delta \Vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow \Gamma \dashv \Theta$, *and* $\Theta \longrightarrow \Omega$, *then* $[\Omega]\Delta \vdash^{\mathsf{dt}} \mathcal{T} \rightsquigarrow [\Omega]\Gamma$.

Proof. We have

$$\frac{\begin{array}{c}(T : \forall\{\phi_1^{\mathsf{c}}\}.\forall\phi_2^{\mathsf{c}}.\ \omega) \in \Delta \qquad \Delta, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{\widehat{\alpha}_i : \star}^{\,i} \Vdash^{\mathsf{u}} [\Delta]\omega \approx (\overline{\widehat{\alpha}_i}^{\,i} \to \star) \dashv \Theta_1, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{\widehat{\alpha}_i : \star = \omega_i}^{\,i} \\ \overline{\Theta_j, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{a_i : \omega_i}^{\,i} \Vdash^{\mathsf{dc}}_{(T\ @\phi_1^{\mathsf{c}}\ @\phi_2^{\mathsf{c}}\ \overline{a_i}^{\,i})} \mathcal{D}_j \rightsquigarrow \mu_j \dashv \Theta_{j+1}, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{a_i : \omega_i}^{\,i}}^{\,j}\end{array}}{\Delta \Vdash^{\mathsf{dt}} \mathbf{data}\ T\ \overline{a_i}^{\,i} = \overline{\mathcal{D}_j}^{\,j\in 1..n} \rightsquigarrow \overline{D_j : \forall\{\phi_1^{\mathsf{c}}\}.\forall\phi_2^{\mathsf{c}}.\ \forall\overline{a_i : \omega_i}^{\,i}.\mu_j}^{\,j} \dashv \Theta_{n+1}}\ \textsc{a-dt-tt}$$

$\Delta, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{\widehat{\alpha}_i : \star}^{\,i} \longrightarrow \Theta_1, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{\widehat{\alpha}_i : \star = \omega_i}^{\,i}$	By Lemma F.7
$\Delta \longrightarrow \Theta_1$	By Lemma F.22
$\overline{\Theta_j, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{a_i : \omega_i}^{\,i} \longrightarrow \Theta_{j+1}, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{a_i : \omega_i}^{\,i}}^{\,j}$	By lemma F.12
$\overline{\Theta_j \longrightarrow \Theta_{j+1}}^{\,j}$	By Lemma F.22
$\Theta_{n+1} \longrightarrow \Omega$	Given
$\Theta_1 \longrightarrow \Omega$	By lemma F.24
$\Delta \longrightarrow \Omega$	By lemma F.24
$\Theta_1, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{\widehat{\alpha}_i : \star = \omega_i}^{\,i} \longrightarrow \Omega, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{\widehat{\alpha}_i : \star = \omega_i}^{\,i}$	By definition
$[\Omega, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{\widehat{\alpha}_i : \star = \omega_i}^{\,i}]([\Delta]\omega) = [\Omega, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{\widehat{\alpha}_i : \star = \omega_i}^{\,i}](\overline{\widehat{\alpha}_i}^{\,i} \to \star)$	By Lemma F.47
$[\Omega]([\Delta]\omega) = \overline{[\Omega]\omega_i}^{\,i} \to \star$	By definition and freshness
$[\Omega]\omega = \overline{[\Omega]\omega_i}^{\,i} \to \star$	By Lemma F.23
$(T : \forall\{\phi_1^{\mathsf{c}}\}.\forall\phi_2^{\mathsf{c}}.\ \overline{[\Omega]\omega_i}^{\,i} \to \star) \in \Omega$	By Lemma F.37
$\overline{\Theta_{j+1} \longrightarrow \Omega}^{\,j}$	By lemma F.24
$\overline{\Theta_{j+1}, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{a_i : \omega_i}^{\,i} \longrightarrow \Omega, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{a_i : \omega_i}^{\,i}}^{\,j}$	By definition
$\overline{[\Omega]\Delta, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{a_i : [\Omega]\omega_i}^{\,i} \vdash^{\mathsf{dc}}_{(T\ @\phi_1^{\mathsf{c}}\ @\phi_2^{\mathsf{c}}\ \overline{a_i}^{\,i})} \mathcal{D}_j \rightsquigarrow [\Omega]\mu_j}^{\,j}$	By Lemma F.52, and Lemma F.35
$[\Omega]\Delta \vdash^{\mathsf{dt}} \mathbf{data}\ T\ \overline{a_i}^{\,i} = \overline{\mathcal{D}_j}^{\,j} \rightsquigarrow \overline{D_j : [\Omega](\forall\{\phi_1^{\mathsf{c}}\}.\forall\phi_2^{\mathsf{c}}.\ \forall\overline{a_i : \omega_i}^{\,i}.\mu_j)}^{\,j}$	By rule DT-TT

□

Lemma F.54 (Soundness of Typing Program). *If* $\Omega; \Gamma \Vdash^{\mathsf{pgm}} pgm : \mu$*, then* $[\Omega]\Omega; [\Omega]\Gamma \vdash^{\mathsf{pgm}} pgm : [\Omega]\mu$*.*

Proof. By induction on the derivation.

- Case

$$\frac{[\Omega]\Omega; [\Omega]\Gamma \vdash e : \sigma}{\Omega; \Gamma \Vdash^{\mathsf{pgm}} e : \sigma}\ \textsc{a-pgm-expr}$$

  The goal holds directly.

- Case

$$\frac{\Omega \Vdash^{\mathsf{sig}} \mathcal{S} \rightsquigarrow T : \eta \qquad \Omega, T : \eta; \Gamma \Vdash^{\mathsf{pgm}} pgm : \mu}{\Omega; \Gamma \Vdash^{\mathsf{pgm}} \mathbf{sig}\,\mathcal{S}; pgm : \mu}\ \textsc{a-pgm-sig}$$

  The goal holds directly from soundness of typing signature (lemma F.51) and I.H..

- Case

A-PGM-DT-TT

$$\dfrac{\begin{array}{c}\Theta_1 = \Omega, \overline{\widehat{\alpha}_i : \star}^{\,i}, \overline{T_i : \widehat{\alpha}_i}^{\,i} \qquad \overline{\Theta_i \Vdash^{\mathsf{dt}} \mathcal{T}_i \rightsquigarrow \Gamma_i \dashv \Theta_{i+1}}^{\,i} \qquad \overline{\widehat{\phi}^{\mathsf{c}}_i = \mathsf{unsolved}([\Theta_{n+1}]\widehat{\alpha}_i)}^{\,i} \\ \overline{\Omega, T_i : \forall \{\phi^{\mathsf{c}}_i\}.(([\Theta_{n+1}]\widehat{\alpha}_i)[\overline{\widehat{\phi}^{\mathsf{c}}_i \mapsto \phi^{\mathsf{c}}_i}^{\,i}])}^{\,i} ; \overline{\Gamma, \forall \{\phi^{\mathsf{c}}_i\}.(([\Theta_{n+1}]\Gamma_i)[\overline{\widehat{\phi}^{\mathsf{c}}_i \mapsto \phi^{\mathsf{c}}_i}^{\,i}][\overline{T_i \mapsto T_i \mathbin{@} \phi^{\mathsf{c}}_i}^{\,i}])}^{\,i} \Vdash^{\mathsf{pgm}} pgm : \mu\end{array}}{\Omega ; \Gamma \Vdash^{\mathsf{pgm}} \mathbf{rec}\, \overline{\mathcal{T}_i}^{\,i \in 1..n} ; pgm : \mu}$$

The key is to prove that ϕ^{c}_i corresponds to the ϕ^{c}_i in rule PGM-DT-TT. The reasoning is similar to the one in lemma F.51.

The key observation here is that, in typing datatype decl (rule A-DT-TT), the result context does not have new unification variables at the end. Therefore, all unsolved unification variable in Θ_{n+1} is in one of the free kind variable in $[\Theta_{n+1}]\widehat{\alpha}_i$. Once we have all the ϕ^{c}_i, the rest of preconditions follow straightforwardly.

□

F.2.7 Principality.

Lemma F.55 (Completeness of Promotion). *Given Δ ok, and $\Delta \longrightarrow \Omega$, and $\widehat{\alpha} \in \Delta$, and $\Delta \Vdash^{\mathsf{ela}} \rho : \omega$, and $[\Delta]\widehat{\alpha} = \widehat{\alpha}$, and $[\Delta]\rho = \rho$, if κ is free of $\widehat{\alpha}$, then there exists ρ_2, Θ and Ω' such that $\Theta \longrightarrow \Omega'$, and $\Omega \longrightarrow \Omega'$, and $\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho \rightsquigarrow \rho_2 \dashv \Theta$.*

Proof. By induction on the lexicographic order indicated in the proof of Theorem F.43.

The proof is essentially the same as Lemma D.45.

For case $\rho = \widehat{\beta}$, and the context $\Delta[\widehat{\alpha}][\widehat{\beta} : \rho_1]$.

$\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} [\Delta]\rho_1 \rightsquigarrow \rho_2 \dashv \Delta_1[\widehat{\alpha}][\widehat{\beta} : \rho_1] \land \Delta_1 \longrightarrow \Omega_1 \land \Omega \longrightarrow \Omega_1$	I.H.
$\Delta[\widehat{\alpha}][\widehat{\beta} : \rho_1] \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \widehat{\beta} \rightsquigarrow \widehat{\beta}_1 \dashv \Delta_1[\widehat{\beta}_2 : \rho_2, \widehat{\alpha}][\widehat{\beta} = \widehat{\beta}_2]$	By rule A-PR-KUVARR-TT
$\Omega_1 = \Omega_1[\widehat{\alpha} = \rho_3][\widehat{\beta} : \rho_1 = \rho_4]$	By Lemma F.22
$\Delta_1[\widehat{\alpha}][\widehat{\beta} : \rho_1] \longrightarrow \Omega_1$	Given
$\Delta_1[\widehat{\beta}_2 : \rho_2, \widehat{\alpha}][\widehat{\beta} : \rho_1] \longrightarrow \Omega_1[\widehat{\beta}_2 : \rho_2 = [\Omega]\rho_4, \widehat{\alpha} = \rho_3][\widehat{\beta} : \rho_1 = \rho_4]$	By Lemma F.28
$\Delta_1[\widehat{\beta}_2 : \rho_2, \widehat{\alpha}][\widehat{\beta} = \widehat{\beta}_2] \longrightarrow \Omega_1[\widehat{\beta}_2 : \rho_2 = [\Omega]\rho_4, \widehat{\alpha} = \rho_3][\widehat{\beta} : \rho_1 = \rho_4]$	By Lemma F.29
$\Omega' = \Omega_1[\widehat{\beta}_2 : \rho_2 = [\Omega]\rho_4, \widehat{\alpha} = \rho_3][\widehat{\beta} : \rho_1 = \rho_4]$	Let
$\Omega \longrightarrow \Omega'$	By Lemma F.26 and Lemma F.24

□

Lemma F.56 (Completeness of Unification). *Given Δ ok, and $\Delta \longrightarrow \Omega$, and $\Delta \Vdash^{\mathsf{ela}} \rho_1 : \omega$ and $\Delta \Vdash^{\mathsf{ela}} \rho_2 : \omega$, and $[\Delta]\rho_1 = \rho_1$ and $[\Delta]\rho_2 = \rho_2$, if $[\Omega]\rho_1 = [\Omega]\rho_2$, then there exists Θ and Ω' such that $\Theta \longrightarrow \Omega'$, and $\Omega \longrightarrow \Omega'$ and $\Delta \Vdash^{\mathsf{u}} \rho_1 \approx \rho_2 \dashv \Theta$.*

Proof. By induction on the lexicographic order indicated in the proof of Theorem F.44. Then case analysis on ρ_1 and ρ_2.

The proof is essentially the same as Lemma D.46.

For case $\rho_1 = \widehat{\alpha}$, and ρ_2 is free of $\widehat{\alpha}$.

$\Delta \Vdash^{\mathsf{pr}}_{\widehat{\alpha}} \rho_2 \rightsquigarrow \rho_3 \dashv \Theta_1 \land \Theta_1 \longrightarrow \Omega_1 \land \Omega \longrightarrow \Omega_1$	By Lemma F.55
$\Theta_1 = \Theta_{11}, \widehat{\alpha} : \omega_3, \Theta_{12} \land \Theta_{11} \Vdash^{\mathsf{ela}} \rho_3 : [\Theta_{11}]\omega$	By Lemma F.6
$\Delta \Vdash^{\mathsf{ela}} \widehat{\alpha} : \omega$	Given
$[\Delta]\omega_3 = \omega$	By rule A-ELA-VAR
$\Omega_1 = \Omega_{11}, \widehat{\alpha} : \omega_3 = \rho_4, \Omega_{12} \land \Theta_{11} \longrightarrow \Omega_{11}$	Lemma F.22

$\Theta_{11} \Vdash^{\mathsf{u}} [\Theta_{11}]\omega_3 \approx [\Theta_{11}]\omega \dashv \Theta_3 \wedge \Theta_3 \longrightarrow \Omega_2 \wedge \Omega_{11} \longrightarrow \Omega_2$	I.H.
$\Delta \Vdash^{\mathsf{u}} \widehat{\alpha} \approx \rho_2 \dashv \Theta_3, \widehat{\alpha} : \omega_3 = \rho_4, \Theta_{12}$	By rule A-U-KVARL-TT
$\Theta_3, \widehat{\alpha} : \omega_3 = \rho_4, \Theta_{12} \longrightarrow \Omega_2, \widehat{\alpha} : \omega_3 = \rho_4, \Omega_{12}$	By Lemma F.28, Lemma F.30, Lemma F.29
$\Omega \longrightarrow \Omega_2, \widehat{\alpha} : \omega_3 = \rho_4, \Omega_{12}$	Similarly

The case when the variable in a local scope is similar.

□

Lemma F.57 (Completeness of Instantiation). *Given* $\Delta \longrightarrow \Omega$*, and* $\Delta \Vdash^{\mathsf{ela}} \rho : \eta$ *and* $\Delta \Vdash^{\mathsf{ela}} \omega : \star$*, and* $[\Delta]\eta = \eta$ *and* $[\Delta]\omega = \omega$*, if* $[\Omega]\Delta \vdash^{\mathsf{inst}} [\Omega]\rho_1 : [\Omega]\eta \sqsubseteq [\Omega]\omega \rightsquigarrow \rho_2$*, then there exists* ρ_2'*,* Θ *and* Ω' *such that* $\Theta \longrightarrow \Omega'$*, and* $\Omega \longrightarrow \Omega'$ *and* $\Delta \Vdash^{\mathsf{inst}} \rho_1 : \eta \sqsubseteq \omega \rightsquigarrow \rho_2' \dashv \Theta$*, and* $[\Omega']\rho_2' = \rho_2$*.*

Proof. By induction on the declarative instantiation.

- Case

$$\frac{}{\Sigma \vdash^{\mathsf{inst}} \mu : \omega \sqsubseteq \omega \rightsquigarrow \mu}\ \textsc{inst-refl}$$

  Follows directly from rule A-INST-REFL and Lemma F.56.
- Case

$$\frac{\Sigma \vdash^{\mathsf{ela}} \rho : \omega_1 \qquad \Sigma \vdash^{\mathsf{inst}} \mu_1 \mathbin{@}\rho : \eta[a \mapsto \rho] \sqsubseteq \omega_2 \rightsquigarrow \mu_2}{\Sigma \vdash^{\mathsf{inst}} \mu_1 : \forall a : \omega_1 . \eta \sqsubseteq \omega_2 \rightsquigarrow \mu_2}\ \textsc{inst-forall}$$

  We case analyze η, and it can only be of the shape $\forall a : \omega_2 . \eta_2$, and $[\Omega]\omega_2 = \omega_1$ and $[\Omega]\eta_2 = \eta$.
  From hypothesis we get $[\Omega]\Delta \vdash^{\mathsf{inst}} ([\Omega]\mu_1 \mathbin{@}[\Omega]\rho) : [\Omega]\mu[a \mapsto [\Omega]\rho] \sqsubseteq [\Omega]\omega_2 \rightsquigarrow [\Omega]\mu_2$.
  By substitution, $[\Omega]\Delta \vdash^{\mathsf{inst}} [\Omega](\mu_1 \mathbin{@}\rho) : [\Omega](\mu[a \mapsto \rho]) \sqsubseteq [\Omega]\omega_2 \rightsquigarrow [\Omega]\mu_2$.
  By definition, $[\Omega, \widehat{\alpha} : \omega_1 = \rho]\Delta \vdash^{\mathsf{inst}} [\Omega, \widehat{\alpha} : \omega_1 = \rho](\mu_1 \mathbin{@}\widehat{\alpha}) : [\Omega, \widehat{\alpha} : \omega_1 = \rho](\mu[a \mapsto \widehat{\alpha}]) \sqsubseteq [\Omega, \widehat{\alpha} : \omega_1 = \rho]\omega_2 \rightsquigarrow [\Omega, \widehat{\alpha} : \omega_1 = \rho]\mu_2$.
  The goal follows directly from I.H., and rule A-INST-FORALL.
- The case for rule INST-FORALL-INFER is similar to the previous case

□

Lemma F.58 (Principality of Kinding).

- *Given* $\Delta \longrightarrow \Omega$*, if* $[\Omega]\Delta \vdash^{\mathsf{k}} \sigma : \eta \rightsquigarrow \mu$*, and* $\Delta \Vdash^{\mathsf{k}} \sigma : \eta' \rightsquigarrow \mu' \dashv \Theta$*, then there exists* Ω' *such that* $\Theta \longrightarrow \Omega'$*, and* $\Omega \longrightarrow \Omega'$*. Moreover,* $[\Omega']\eta' = \eta$*. Furthermore, if* μ *and* μ' *are monotypes, then* $[\Omega']\mu' = \mu$*.*
- *Given* $\Delta \longrightarrow \Omega$*, if* $[\Omega]\Delta \vdash^{\mathsf{kc}} \sigma \Leftarrow [\Omega]\eta \rightsquigarrow \mu$*, and* $\Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \eta \rightsquigarrow \mu' \dashv \Theta$*, then there exists* Ω' *such that* $\Theta \longrightarrow \Omega'$*, and* $\Omega \longrightarrow \Omega'$*. Furthermore, if* μ *and* μ' *are monotypes, then* $[\Omega']\mu' = \mu$*.*
- *Given* $\Delta \longrightarrow \Omega$*, if* $[\Omega]\Delta \vdash^{\mathsf{inst}} [\Omega]\rho_1 : [\Omega]\eta \sqsubseteq (\omega_1 \ \rightarrow \ \omega_2) \rightsquigarrow \rho_3$*, and* $[\Omega]\Delta \vdash^{\mathsf{kc}} \tau \Leftarrow \omega_1 \rightsquigarrow \rho_4$ *and* $\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \eta) \bullet \tau : \omega \rightsquigarrow \rho_2 \dashv \Theta$*, then there exists* Ω' *such that* $\Theta \longrightarrow \Omega'$*, and* $\Omega \longrightarrow \Omega'$*. Moreover,* $[\Omega']\omega = \omega_2$*. Further,* $[\Omega']\rho_2 = \rho_3\, \rho_4$*.*

Proof. From this lemma, we make use of *Any* to ensure every algorithmic context can be extended to a complete context. The existence of *Any* does not affect at all how this lemma is used.

By induction on the algorithmic kinding.

Part 1 • The case for rules A-KTT-STAR, A-KTT-KUVAR, A-KTT-NAT, A-KTT-VAR, A-KTT-TCON, and A-KTT-ARROW follows trivially by picking $\Theta = \Delta$, and $\Omega' = \Omega$.

- Case

$$\frac{\Delta \Vdash^{\mathsf{kc}} \kappa \Leftarrow \star \rightsquigarrow \omega \dashv \Delta_1 \qquad \Delta_1, a : \omega \Vdash^{\mathsf{kc}} \sigma \Leftarrow \star \rightsquigarrow \mu \dashv \Delta_2, a : \omega, \Delta_3 \qquad \Delta_3 \hookrightarrow a}{\Delta \Vdash^{\mathsf{k}} \forall a : \kappa . \sigma : \star \rightsquigarrow \forall a : \omega . [\Delta_3]\mu \dashv \Delta_2, \mathrm{unsolved}(\Delta_3)}\ \textsc{a-ktt-forall}$$

$[\Omega]\Delta \vdash^{k} \forall a : \kappa.\sigma : \star \rightsquigarrow \forall a : \omega_1.\mu_1$	Given
$[\Omega]\Delta \vdash^{kc} \kappa \Leftarrow \star \rightsquigarrow \omega_1 \wedge [\Omega]\Delta, a : \omega_1 \vdash^{kc} \sigma \Leftarrow \star \rightsquigarrow \mu_1$	By inversion
$\Delta_1 \longrightarrow \Omega_1 \wedge \Omega \longrightarrow \Omega_1 \wedge [\Omega_1]\omega = \omega_1$	I.H.
$[\Omega_1, a : \omega](\Delta_1, a : \omega)$	
$= [\Omega_1]\Delta_1, a : \omega_1$	By definition
is a well-formed permutation of $[\Omega]\Delta$	By Lemma F.35 and Lemma F.39
$[\Omega_1, a : \omega](\Delta_1, a : \omega) \vdash^{kc} \sigma \Leftarrow \star \rightsquigarrow \mu_1$	Follows
$\Delta_2, a : \omega, \Delta_3 \longrightarrow \Omega_2 \wedge \Omega_1, a : \omega \longrightarrow \Omega_2$	I.H.
$\Delta_1, a : \omega \longrightarrow \Delta_2, a : \omega, \Delta_3$	By Lemma F.10
$\Delta_1 \longrightarrow \Delta_2 \wedge \Delta_3$ **soft**	By Lemma F.22
$\Omega_2 = \Omega_{21}, a : \omega, \Omega_{22} \wedge \Delta_2 \longrightarrow \Omega_{21} \wedge \Omega_1 \longrightarrow \Omega_{21} \wedge \Omega_{22}$ **soft**	By Lemma F.22
construct Ω_{23} which contain same domain of unsolved(Δ_3)	
$\Omega' = \Omega_{21}, \Omega_{23}$	Let
$\Delta_2, \text{unsolved}(\Delta_3) \longrightarrow \Omega'$	By rule A-CTXE-SOLVE-TT
$\Omega_1 \longrightarrow \Omega'$	By rule A-CTXE-ADDSOLVED-TT

- The case for rule A-KTT-FORALLI is similar as the previous case.
- Case

A-KTT-APP
$$\frac{\Delta \Vdash^{k} \tau_1 : \eta_1 \rightsquigarrow \rho_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{kapp} (\rho_1 : [\Delta_1]\eta_1) \bullet \tau_2 : \omega \rightsquigarrow \rho \dashv \Theta}{\Delta \Vdash^{k} \tau_1\,\tau_2 : \omega \rightsquigarrow \rho \dashv \Theta}$$

$[\Omega]\Delta \vdash^{k} \tau_1\,\tau_2 : \omega_2 \rightsquigarrow \rho_2\,\rho_3$	Given
$[\Omega]\Delta \vdash^{k} \tau_1 : \eta_2 \rightsquigarrow \rho_4 \wedge [\Omega]\Delta \vdash^{inst} \rho_4 : \eta_2 \sqsubseteq \omega_1 \to \omega_2 \rightsquigarrow \rho_2$	
$\wedge [\Omega]\Delta \vdash^{kc} \tau_2 \Leftarrow \omega_1 \rightsquigarrow \rho_3$	By inversion
$\Delta_1 \longrightarrow \Omega_1 \wedge \Omega \longrightarrow \Omega_1 \wedge [\Omega_1]\rho_1 = \rho_4 \wedge [\Omega_1]\eta_1 = \eta_2$	I.H.
$[\Omega_1]\Delta$ is a well-formed permutation of $[\Omega]\Delta$	By Lemma F.33 and Lemma F.39
$[\Omega_1]\Delta \vdash^{inst} [\Omega_1]\rho_1 : [\Omega]\eta_1 \sqsubseteq \omega_1 \to \omega_2 \rightsquigarrow \rho_2$	By equations
$[\Omega_1]\Delta \vdash^{kc} \tau_2 \Leftarrow \omega_1 \rightsquigarrow \rho_3$	By equations
$\Theta \longrightarrow \Omega' \wedge \Omega_1 \longrightarrow \Omega' \wedge [\Omega']\rho = \rho_2\,\rho_3 \wedge [\Omega']\omega = \omega_2$	By Part 3
$\Omega \longrightarrow \Omega'$	By Lemma F.24

- Case

A-KTT-KAPP
$$\frac{\Delta \Vdash^{k} \tau_1 : \eta \rightsquigarrow \rho_1 \dashv \Delta_1 \qquad [\Delta_1]\eta = \forall a : \omega.\eta_2 \qquad \Delta_1 \Vdash^{kc} \tau_2 \Leftarrow \omega \rightsquigarrow \rho_2 \dashv \Delta_2}{\Delta \Vdash^{k} \tau_1\,@\tau_2 : \eta_2[a \mapsto \rho_2] \rightsquigarrow \rho_1\,@\rho_2 \dashv \Delta_2}$$

The goal follows directly from I.H., Part 2, and Lemma F.24.

$[\Omega]\Delta \vdash^{k} \tau_1\,@\tau_2 : \mu_3[a \mapsto \rho_3] \rightsquigarrow \rho_4\,@\rho_3$	Given
$[\Omega]\Delta \vdash^{k} \tau_1 : \forall a : \omega_2.\mu_3 \rightsquigarrow \rho_4 \wedge [\Omega]\Delta \vdash^{kc} \tau_2 \Leftarrow \omega_2 \rightsquigarrow \rho_3$	By inversion
$\Delta_1 \longrightarrow \Omega_1 \wedge \Omega \longrightarrow \Omega_1 \wedge [\Omega_1]\eta = \forall a : \omega_2.\mu_3 \wedge [\Omega_1]\rho_1 = \rho_4$	I.H.
$[\Omega_1]\Delta$ is a well-formed permutation of $[\Omega]\Delta$	By Lemma F.33 and Lemma F.39
$[\Omega_1]\eta = [\Omega_1]([\Delta_1]\eta)$	By Lemma F.23
$= \forall a : [\Omega_1]\omega.[\Omega_1]\eta_2$	By definition
$= [\Omega_1]\eta = \forall a : \omega_2.\mu_3$	Given
$[\Omega_1]\omega = \omega_2$	Follows
$[\Omega_1]\eta_2 = \mu_3$	Follows
$[\Omega_1]\Delta \vdash^{kc} \tau_2 \Leftarrow [\Omega_1]\omega \rightsquigarrow \rho_3$	By equations

$$\begin{array}{l|l}
\Delta_2 \longrightarrow \Omega' \wedge \Omega_1 \longrightarrow \Omega' \wedge [\Omega']\rho_2 = \rho_3 & \text{By Part 2} \\
\Omega \longrightarrow \Omega' & \text{By Lemma F.24} \\
[\Omega'](\eta_2[a \mapsto \rho_2]) & \\
= ([\Omega']\eta_2)[a \mapsto [\Omega']\rho_2] & \text{By substitution} \\
= (\mu_3)[a \mapsto \rho_3] & \text{By Lemma F.31}
\end{array}$$

- The case for rule A-KTT-KAPP-INFER is similar as the previous case.

Part 2 We have

A-KC-SUB

$$\frac{\Delta \Vdash^{\mathsf{k}} \sigma : \eta \rightsquigarrow \mu_1 \dashv \Delta_1 \qquad \Delta_1 \Vdash^{\mathsf{inst}} \mu_1 : [\Delta_1]\eta \sqsubseteq [\Delta_1]\omega \rightsquigarrow \mu_2 \dashv \Delta_2}{\Delta \Vdash^{\mathsf{kc}} \sigma \Leftarrow \omega \rightsquigarrow \mu_2 \dashv \Delta_2}$$

$$\begin{array}{l|l}
[\Omega]\Delta \vdash^{\mathsf{kc}} \sigma \Leftarrow [\Omega]\omega \rightsquigarrow \mu_4 & \text{Given} \\
[\Omega]\Delta \vdash^{\mathsf{k}} \sigma : \eta_3 \rightsquigarrow \mu_3 & \text{By inversion} \\
[\Omega]\Delta \vdash^{\mathsf{inst}} \mu_3 : \eta_3 \sqsubseteq [\Omega]\omega \rightsquigarrow \mu_4 & \text{By inversion} \\
\Delta_1 \longrightarrow \Omega_1 \wedge \Omega \longrightarrow \Omega_1 \wedge [\Omega_1]\eta = \eta_3 & \text{I.H.} \\
\text{If } \mu_1 \text{ and } \mu_3 \text{ are monotypes, then } [\Omega_1]\mu_1 = \mu_3 & \text{I.H.} \\
[\Omega_1]\Delta \text{ is a well-formed permutation of } [\Omega]\Delta & \text{By Lemma F.33 and Lemma F.39} \\
[\Omega_1]\omega = [\Omega]\omega & \text{By Lemma F.31} \\
[\Omega_1]\Delta_1 \vdash^{\mathsf{inst}} \mu_3 : [\Omega_1]\eta \sqsubseteq [\Omega_1]\omega \rightsquigarrow \mu_4 & \text{Follows} \\
\text{If } \mu \text{ and } \mu_3 \text{ are monotypes} & \\
[\Omega_1]\Delta_1 \vdash^{\mathsf{inst}} [\Omega_1]\mu_1 : [\Omega_1]\eta \sqsubseteq [\Omega_1]\omega \rightsquigarrow \mu_4 & \text{Follows} \\
[\Omega_1]\Delta_1 \vdash^{\mathsf{inst}} [\Omega_1]\mu_1 : [\Omega_1]([\Delta_1]\eta) \sqsubseteq [\Omega_1]([\Delta_1]\omega) \rightsquigarrow \mu_4 & \text{By Lemma F.23} \\
\Omega_1 \longrightarrow \Omega' \wedge \Delta_2 \longrightarrow \Omega' \wedge [\Omega']\mu_2 = \mu_4 & \text{By Lemma F.57} \\
\text{If } \mu_1 \text{ and } \mu_3 \text{ are polytypes} & \\
\text{then only rule INST-REFL and rule A-INST-REFL can apply} & \\
\Delta_2 = \Delta_1 & \text{Follows} \\
\Omega' = \Omega_1 & \text{Let}
\end{array}$$

Part 3 • Case

A-KAPP-TT-ARROW

$$\frac{\Delta \Vdash^{\mathsf{kc}} \tau \Leftarrow \omega_1 \rightsquigarrow \rho_2 \dashv \Theta}{\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \omega_1 \ \rightarrow \ \omega_2) \bullet \tau : \omega_2 \rightsquigarrow \rho_1\, \rho_2 \dashv \Theta}$$

$$\begin{array}{l|l}
[\Omega]\Delta \vdash^{\mathsf{inst}} [\Omega]\rho_1 : [\Omega]\omega_1 \ \rightarrow \ [\Omega]\omega_2 \sqsubseteq [\Omega]\omega_1 \ \rightarrow \ [\Omega]\omega_2 \rightsquigarrow [\Omega]\rho_1 & \text{Given} \\
[\Omega]\Delta \vdash^{\mathsf{kc}} \tau \Leftarrow [\Omega]\omega_1 \rightsquigarrow \rho_4 & \text{Given} \\
\text{The goal follows directly from Part 2} &
\end{array}$$

- Case

A-KAPP-TT-FORALL

$$\frac{\Delta, \widehat{\alpha} : \omega_1 \Vdash^{\mathsf{kapp}} (\rho_1 \ @\widehat{\alpha} : \eta[a \mapsto \widehat{\alpha}]) \bullet \tau : \omega \rightsquigarrow \rho \dashv \Theta}{\Delta \Vdash^{\mathsf{kapp}} (\rho_1 : \forall a : \omega_1 . \eta) \bullet \tau : \omega \rightsquigarrow \rho \dashv \Theta}$$

$$\begin{array}{l|l}
[\Omega]\Delta \vdash^{\mathsf{inst}} [\Omega]\rho_1 : \forall a : [\Omega]\omega_1 . [\Omega]\eta \sqsubseteq \omega_3 \ \rightarrow \ \omega_4 \rightsquigarrow \rho_3 & \text{Given} \\
[\Omega]\Delta \vdash^{\mathsf{kc}} \tau \Leftarrow \omega_3 \rightsquigarrow \rho_4 & \text{Given} \\
[\Omega]\Delta \vdash^{\mathsf{ela}} \rho_5 : [\Omega]\omega_1 & \text{By inversion} \\
[\Omega]\Delta \vdash^{\mathsf{inst}} [\Omega]\rho_1 \ @\rho_5 : ([\Omega]\eta)[a \mapsto \rho_5] \sqsubseteq \omega_3 \ \rightarrow \ \omega_4 \rightsquigarrow \rho_3 & \text{By inversion} \\
[\Omega, \widehat{\alpha} : \omega_1 = \rho_5](\Delta, \widehat{\alpha} : \omega_1) \vdash^{\mathsf{inst}} & \\
\quad [\Omega, \widehat{\alpha} : \omega_1 = \rho_5](\rho_1 \ @\widehat{\alpha}) : [\Omega, \widehat{\alpha} : \omega_1 = \rho_5](\eta[a \mapsto \widehat{\alpha}]) \sqsubseteq \omega_3 \ \rightarrow \ \omega_4 \rightsquigarrow \rho_3 & \text{By definition}
\end{array}$$

The goal follows from I.H.

- The case for rule A-KAPP-TT-FORALL-INFER is similar as the previous case.
- Case

$$\dfrac{\Delta_1, \widehat{\alpha}_1 : \star, \widehat{\alpha}_2 : \star, \widehat{\alpha} : \omega = (\widehat{\alpha}_1 \rightarrow \widehat{\alpha}_2), \Delta_2 \Vdash^{\mathsf{kc}} \tau \Leftarrow \widehat{\alpha}_1 \leadsto \rho_2 \dashv \Theta}{\Delta_1, \widehat{\alpha} : \omega, \Delta_2 \Vdash^{\mathsf{kapp}} (\rho_1 : \widehat{\alpha}) \bullet \tau : \widehat{\alpha}_2 \leadsto \rho_1\, \rho_2 \dashv \Theta}\ \textsc{a-kapp-tt-kuvar}$$

As $\widehat{\alpha}$ can only be instituted with monotypes, obviously the declarative instantiation judgment must be rule INST-REFL. Then the goal follows directly from Part 2.

□

Lemma F.59 (Principality of Typing Data Constructor Declaration). *Given $\Delta \longrightarrow \Omega$, if $[\Omega]\Delta \vdash^{\mathsf{dc}}_{\rho} \mathcal{D} \leadsto \mu_1$, and $\Delta \Vdash^{\mathsf{dc}}_{\rho} \mathcal{D} \leadsto \mu_2 \dashv \Theta$, then there exists Ω' such that $\Theta \longrightarrow \Omega'$, and $\Omega \longrightarrow \Omega'$.*

Proof. We have

$$\dfrac{\Delta, \blacktriangleright_D \Vdash^{\mathsf{k}} \forall \phi.(\overline{\tau_i}^{\,i} \rightarrow \rho) : \star \leadsto \mu \dashv \Theta_1, \blacktriangleright_D, \Theta_2 \qquad \widehat{\phi}^{\mathsf{c}} = \mathsf{unsolved}(\Theta_2)}{\Delta \Vdash^{\mathsf{dc}}_{\rho} \forall \phi. D\, \overline{\tau_i}^{\,i} \leadsto \forall \{\phi^{\mathsf{c}}\}.(([\Theta_2]\mu)[\widehat{\phi}^{\mathsf{c}} \mapsto \phi^{\mathsf{c}}]) \dashv \Theta_1}\ \textsc{a-dc-tt}$$

$$\begin{array}{l|l}
[\Omega]\Delta \vdash^{\mathsf{dc}}_{\rho} \mathcal{D} \leadsto \mu_1 & \text{Given} \\
[\Omega]\Delta, \phi^{\mathsf{c}} \vdash^{\mathsf{k}} \forall \phi.\overline{\tau_i}^{\,i} \rightarrow \rho : \star \leadsto \mu_1 & \text{By inversion} \\
\Delta, \blacktriangleright_D \Vdash^{\mathsf{k}} \forall \phi.(\overline{\tau_i}^{\,i} \rightarrow \rho) : \star \leadsto \mu \dashv \Theta_1, \blacktriangleright_D, \Theta_2 & \text{Given} \\
\Delta, \blacktriangleright_D, \phi^{\mathsf{c}} \Vdash^{\mathsf{k}} \forall \phi.(\overline{\tau_i}^{\,i} \rightarrow \rho) : \star \leadsto \mu \dashv \Theta_1, \blacktriangleright_D, \phi^{\mathsf{c}}, \Theta_2 & \text{By weakening} \\
\Delta \longrightarrow \Omega & \text{Given} \\
\Delta, \blacktriangleright_D, \phi^{\mathsf{c}} \longrightarrow \Omega, \blacktriangleright_D, \phi^{\mathsf{c}} & \text{By definition} \\
[\Omega, \blacktriangleright_D, \phi^{\mathsf{c}}](\Delta, \blacktriangleright_D, \phi^{\mathsf{c}}) \vdash^{\mathsf{k}} \forall \phi.\overline{\tau_i}^{\,i} \rightarrow \rho : \star \leadsto \mu_1 & \text{By definition} \\
\Theta_1, \blacktriangleright_D, \phi^{\mathsf{c}}, \Theta_2 \longrightarrow \Omega_1 \wedge \Omega, \blacktriangleright_D, \phi^{\mathsf{c}} \longrightarrow \Omega_1 & \text{By Lemma F.58} \\
\Omega_1 = \Omega', \blacktriangleright_D, \Omega_{12} \wedge \Theta_1 \longrightarrow \Omega_1 \wedge \Theta_1 \longrightarrow \Omega' \wedge \Omega \longrightarrow \Omega' & \text{By Lemma F.22}
\end{array}$$

□

Lemma F.60 (Principality of Typing Datatype Declaration). *Given $\Delta \longrightarrow \Omega$, if $[\Omega]\Delta \vdash^{\mathsf{dt}} \mathcal{T} \leadsto \Psi$, and $\Delta \Vdash^{\mathsf{dt}} \mathcal{T} \leadsto \Gamma \dashv \Theta$, then there exists Ω' such that $\Theta \longrightarrow \Omega'$, and $\Omega \longrightarrow \Omega'$.*

Proof. We have

$$\dfrac{\begin{array}{c}(T : \forall \{\phi_1^{\mathsf{c}}\}.\forall \phi_2^{\mathsf{c}}.\, \omega) \in \Delta \qquad \Delta, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{\widehat{\alpha}_i : \star}^{\,i} \Vdash^{\mathsf{u}} [\Delta]\omega \approx (\overline{\widehat{\alpha}_i}^{\,i} \rightarrow \star) \dashv \Theta_1, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{\widehat{\alpha}_i : \star = \omega_i}^{\,i} \\ \overline{\Theta_j, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{a_i : \omega_i}^{\,i} \Vdash^{\mathsf{dc}}_{(T\ @\phi_1^{\mathsf{c}}\ @\phi_2^{\mathsf{c}}\, \overline{a_i}^{\,i})} \mathcal{D}_j \leadsto \mu_j \dashv \Theta_{j+1}, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{a_i : \omega_i}^{\,i}}^{\,j}\end{array}}{\Delta \Vdash^{\mathsf{dt}} \mathbf{data}\ T\, \overline{a_i}^{\,i} = \overline{\mathcal{D}_j}^{\,j \in 1..n} \leadsto \overline{D_j : \forall \{\phi_1^{\mathsf{c}}\}.\forall \phi_2^{\mathsf{c}}.\, \forall \overline{a_i : \omega_i}^{\,i}.\mu_j}^{\,j} \dashv \Theta_{n+1}}\ \textsc{a-dt-tt}$$

$$\begin{array}{l|l}
[\Omega]\Delta \vdash^{\mathsf{dc}}_{\rho} \mathcal{D} \leadsto \mu_1 & \text{Given} \\
(T : (\forall \{\phi_1^{\mathsf{c}}\}.\forall \phi_2^{\mathsf{c}}.\, [\Omega]\omega)) \in [\Omega]\Delta & \text{Inversion} \\
[\Omega]\omega = \overline{\omega_i'}^{\,i} \rightarrow \star & \text{Inversion} \\
[\Omega]\Delta, \phi_1^{\mathsf{c}}, \phi_2^{\mathsf{c}}, \overline{a_i : \omega_i'}^{\,i} \vdash^{\mathsf{dc}}_{(T\ @\phi_1^{\mathsf{c}}\ @\phi_2^{\mathsf{c}}\, \overline{a_i}^{\,i})} \mathcal{D}_j \leadsto \mu_j & \text{Inversion} \\
\Delta \longrightarrow \Omega & \text{Given}
\end{array}$$

$$
\begin{array}{l|l}
\Delta, \phi_1^c, \phi_2^c, \overline{\widehat{\alpha}_i : \star}^{\,i} \longrightarrow \Omega, \phi_1^c, \phi_2^c, \overline{\widehat{\alpha}_i : \star = \omega_i'}^{\,i} & \text{By definition} \\
[\Omega, \phi_1^c, \phi_2^c, \overline{\widehat{\alpha}_i : \star = \omega_i'}^{\,i}]\omega = [\Omega, \phi_1^c, \phi_2^c, \overline{\widehat{\alpha}_i : \star = \omega_i'}^{\,i}](\overline{\widehat{\alpha}_i}^{\,i} \to \star) & \text{By substitution} \\
\Omega, \phi_1^c, \phi_2^c, \overline{\widehat{\alpha}_i : \star = \omega_i'}^{\,i} \longrightarrow \Omega_1 & \text{By Lemma F.56} \\
\Theta_1, \phi_1^c, \phi_2^c, \overline{\widehat{\alpha}_i : \star = \omega_i}^{\,i} \longrightarrow \Omega_1 & \text{By Lemma F.56} \\
\Omega_1 = \Omega_2, \Omega_3 \land \Omega \longrightarrow \Omega_2 \land \Theta_1 \longrightarrow \Omega_2 & \text{By Lemma F.22} \\
\Theta_1, \phi_1^c, \phi_2^c, \overline{a_i : \omega_i}^{\,i} \longrightarrow \Omega_2, \phi_1^c, \phi_2^c, \overline{a_i : \omega_i}^{\,i} & \text{By definition} \\
[\Omega_2, \phi_1^c, \phi_2^c, \overline{a_i : \omega_i}^{\,i}](\Theta_1, \phi_1^c, \phi_2^c, \overline{a_i : \omega_i}^{\,i}) & \\
= [\Omega_2]\Theta_1, \phi_1^c, \phi_2^c, \overline{a_i : [\Omega_2]\omega_i}^{\,i} & \text{By definition} \\
= [\Omega_2]\Theta_1, \phi_1^c, \phi_2^c, \overline{a_i : \omega_i'}^{\,i} & \text{By Lemma F.23} \\
\text{is a well-formed permutation of } [\Omega]\Delta, \phi_1^c, \phi_2^c, \overline{a_i : \omega_i'}^{\,i} & \text{By Lemma F.39} \\
[\Omega]\Delta, \phi_1^c, \phi_2^c, \overline{a_i : \omega_i'}^{\,i} \vdash^{\mathsf{dc}}_{(T \; @\phi_1^c \; @\phi_2^c \, \overline{a_i}^{\,i})} \mathcal{D}_j \rightsquigarrow \mu_j & \text{Given} \\
[\Omega_2, \phi_1^c, \phi_2^c, \overline{a_i : \omega_i}^{\,i}](\Theta_1, \phi_1^c, \phi_2^c, \overline{a_i : \omega_i}^{\,i}) \vdash^{\mathsf{dc}}_{(T \; @\phi_1^c \; @\phi_2^c \, \overline{a_i}^{\,i})} \mathcal{D}_j \rightsquigarrow \mu_j & \text{Follows} \\
(\Theta_2, \phi_1^c, \phi_2^c, \overline{a_i : \omega_i}^{\,i}) \longrightarrow \Omega_3 & \text{By Lemma F.59} \\
(\Omega_2, \phi_1^c, \phi_2^c, \overline{a_i : \omega_i}^{\,i}) \longrightarrow \Omega_3 & \text{By Lemma F.59}
\end{array}
$$

Repeating the process for each j, we can finally get $(\Theta_{n+1}, \phi_1^c, \phi_2^c, \overline{a_i : \omega_i}^{\,i}) \longrightarrow \Omega''$, and $(\Omega_{n+1}, \phi_1^c, \phi_2^c, \overline{a_i : \omega_i}^{\,i}) \longrightarrow \Omega''$.

$$
\begin{array}{l|l}
\Omega'' = \Omega', \Omega_0 \land \Theta_{n+1} \longrightarrow \Omega' \land \Omega_{n+1} \longrightarrow \Omega' & \text{By Lemma F.22} \\
\Omega \longrightarrow \Omega' & \text{By Lemma F.24}
\end{array}
$$

□

Theorem F.61 (Principality of Typing a Datatype Declaration Group). *If* $\Omega \Vdash^{\mathsf{grp}} \textbf{rec}\, \overline{\mathcal{T}_i}^{\,i} \rightsquigarrow \overline{\eta_i}^{\,i} ; \overline{\Gamma_i}^{\,i}$, *then whenever* $[\Omega]\Omega \vdash^{\mathsf{grp}} \textbf{rec}\, \overline{\mathcal{T}_i}^{\,i} \rightsquigarrow \overline{\eta_i'}^{\,i} ; \overline{\Psi_i}^{\,i}$ *holds, we have* $[\Omega]\Omega \vdash [\Omega]\eta_i \le \eta_i'$.

Proof. Given

PGM-DT-TT

$$
\frac{\overline{\Sigma, \cup\overline{\phi_i^c}^{\,i}, \overline{T_i : \omega_i}^{\,i} \vdash^{\mathsf{dt}} \mathcal{T}_i \rightsquigarrow \Psi_i}^{\,i} \qquad \overline{\dfrac{\overline{\Sigma, \phi_i^c \vdash^{\mathsf{ela}} \omega_i : \star}^{\,i} \qquad \overline{\phi_i^c \in \mathcal{Q}(\omega_i)}^{\,i}}{\Sigma, \overline{T_i : \forall\{\phi_i^c\}.\omega_i}^{\,i} ; \Psi, \forall\{\phi_i^c\}.\Psi_i[\overline{T_i \mapsto T_i \; @\phi_i^c}^{\,i}] \vdash^{\mathsf{pgm}} pgm : \sigma}}^{\,i}}{\Sigma; \Psi \vdash^{\mathsf{pgm}} \textbf{rec}\, \overline{\mathcal{T}_i}^{\,i} ; pgm : \sigma}
$$

A-PGM-DT-TT

$$
\frac{\overline{\dfrac{\Theta_1 = \Omega, \overline{\widehat{\alpha}_i : \star}^{\,i}, \overline{T_i : \widehat{\alpha}_i}^{\,i}}{\Omega, T_i : \forall\{\phi_i^c\}.(([\Theta_{n+1}]\widehat{\alpha}_i)[\overline{\widehat{\phi}_i^c \mapsto \phi_i^c}^{\,i}])}}^{\,i} \; \overline{\dfrac{\overline{\Theta_i \Vdash^{\mathsf{dt}} \mathcal{T}_i \rightsquigarrow \Gamma_i \dashv \Theta_{i+1}}^{\,i} \qquad \overline{\widehat{\phi}_i^c = \mathsf{unsolved}([\Theta_{n+1}]\widehat{\alpha}_i)}^{\,i}}{; \Gamma, \forall\{\phi_i^c\}.(([\Theta_{n+1}]\Gamma_i)[\overline{\widehat{\phi}_i^c \mapsto \phi_i^c}^{\,i}][\overline{T_i \mapsto T_i \; @\phi_i^c}^{\,i}]) \Vdash^{\mathsf{pgm}} pgm : \mu}}^{\,i}}{\Omega; \Gamma \Vdash^{\mathsf{pgm}} \textbf{rec}\, \overline{\mathcal{T}_i}^{\,i \in 1..n} ; pgm : \mu}
$$

Our goal is to prove that $[\Omega]\Delta \vdash \forall\{\phi_i^c\}.(([\Theta_{n+1}]\widehat{\alpha}_i)[\overline{\widehat{\phi}_i^c \mapsto \phi_i^c}^{\,i}]) \le \forall\{\phi_i^c\}.\omega_i$.

Similar as the proof in Lemma F.60, we can weaken the context Θ_i by adding $\cup\overline{\phi_i^c}^{\,i}$. By weakening we can get Θ_{n+1}', which is exactly the same as Θ_{n+1}, except for the addition of $\cup\overline{\phi_i^c}^{\,i}$.

Let Ω_1 be $\Omega, \cup\overline{\phi_i^c}^{\,i}, \overline{\widehat{\alpha}_i : \star = \omega_i}^{\,i}, \overline{T_i : \widehat{\alpha}_i}^{\,i}$.

According to the definition, our goal is equivalent to prove that for some Ω', we have $\Theta'_{n+1} \longrightarrow \Omega'$, and $[\Omega']\widehat{\alpha}_i = \omega_i$. According to Lemma F.60, we can prove there is indeed a Ω', such that $\Omega_1 \longrightarrow \Omega'$ and $\Theta'_{n+1} \longrightarrow \Omega'$. Moreover $[\Omega']\widehat{\alpha}_i = [\Omega_1]\widehat{\alpha}_i = \omega_i$ by Lemma F.31, so we are done.

□

REFERENCES

Richard A Eisenberg. 2016. *Dependent types in haskell: Theory and practice.* Ph.D. Dissertation. University of Pennsylvania.

Fritz Henglein. 1993. Type Inference with Polymorphic Recursion. *ACM Trans. Program. Lang. Syst.* 15, 2 (April 1993), 253–289. https://doi.org/10.1145/169701.169692

François Pottier and Didier Rémy. 2005. The essence of ML type inference.

Dimitrios Vytiniotis, Simon Peyton Jones, Tom Schrijvers, and Martin Sulzmann. 2011. OutsideIn (X) Modular type inference with local assumptions. *Journal of functional programming* 21, 4-5 (2011), 333–412.

Thomas Winant, Dominique Devriese, Frank Piessens, and Tom Schrijvers. 2014. Partial type signatures for haskell. In *International Symposium on Practical Aspects of Declarative Languages*. Springer, 17–32.

Brent A Yorgey, Stephanie Weirich, Julien Cretin, Simon Peyton Jones, Dimitrios Vytiniotis, and José Pedro Magalhães. 2012. Giving Haskell a promotion. In *Proceedings of the 8th ACM SIGPLAN workshop on Types in language design and implementation.* ACM, 53–66.