Inductive Data Types Based on Fibrations Theory in Programming

Decheng Miao¹, Jianqing Xi², Yubin Guo³ and Deyou Tang²

1 School of Information Science and Engineering, Shaoguan University, Shaoguan, China
2 School of Software, South China University of Technology, Guangzhou, China
3 School of Information, South China University of Agriculture, Guangzhou, China

Traditional methods including algebra and category theory have some deficiencies in analyzing semantics properties and describing inductive rules of inductive data types, we present a method based on Fibrations theory aiming at those questions above. We systematically analyze some basic logical structures of inductive data types about a fibration such as re-indexing functor, truth functor and comprehension functor, make semantics models of non-indexed fibration, single-sorted indexed fibration and many-sorted indexed fibration respectively. On this basis, we thoroughly discuss semantics properties of fibred, single-sorted indexed and many-sorted indexed inductive data types, and abstractly describe their inductive rules with universality. Furthermore, we briefly introduce applications of the three inductive data types for analyzing semantics properties and describing inductive rules based on Fibrations theory via some examples. Compared with traditional methods, our works have the following three advantages. Firstly, brief descriptions and flexible expansibility of Fibrations theory can analyze semantics properties of inductive data types accurately, whose semantics are computed automatically. Secondly, superior abstractness of Fibrations theory does not rely on particular computing environments to depict inductive rules of inductive data types with universality. Thirdly, its rigorousness and consistence provide sound basis for testing and maintenance of software development.

ACM CCS (2012) Classification: Theory of computation → Logic → Constraint and logic programming

Keywords: inductive data types, Fibrations theory, semantics properties, induction rules, adjoint functor

1. Introduction

Traditional methods of inductive data types are mainly algebra and categorical theory [1-3]. The former focuses on describing the finite syntax construction of inductive data types, e.g., Abstract Data Type Research Groups encapsulated inductive data types and their operations in Σ-algebras. The latter presents model of type theory in local Cartesian closed category, but both render inductive data types and predicates denoting their semantic properties to coexist in the same category, which leads to functors and their lifting are equivalent. Traditional methods are difficult to process the recursive computations of inductive data types effectively, and there are some limitation to analyze semantic properties and describe inductive rules.

Fibrations theory is a new field in computer science foundation. Recently, it has become a hot spot, especially in categorical methods. It is also applied widely by database system modeling [4-6], software specifications [7] and programming methods [8], etc., which provides universal theoretical tools, thinking means and research ways for depicting diverse relationships between structured systems effectively. The method of inductive data types based on Fibrations theory in programming integrates traditional methods, provides a mathematical framework of syntax construction and semantic computation for inductive data types. It also incorporates their formal semantics and program logic of programming, which enhances the abilities of program languages on analyzing semantic properties and describing inductive rules.

In the formal framework of Fibrations theory, the predicates representing semantics of inductive data types do not limit to functions or morphisms, but lift to objects in total category. Meanwhile, it is more important that inductive data types and their predicates do not coexist...
in the same category any more, but constructs functor lifting in total category to depict recursive computing and program logic of inductive data types abstractly. Hermida and Jacobs have done lots of foundational researching works for this in [9]. The main idea of inductive data types based on Fibration theory is that we take inductive data types in programming to be object-set in base category, semantic properties of inductive data types to be object-set in total category, establish the responsible relations in program logic directly between inductive data types and their semantic properties by semantic models of fibration, construct recursive operations of inductive data types to analyze semantic properties by the tools of endo-functors in base category and their lifting in total category as well as and abstractly describe inductive rules with universality by initial property of initial algebra.

Our primary works are researching inductive data types and their inductive rules by Fibrations theory. The rest of this paper is structured as follows. In Section 2, we firstly introduce some basic concepts for our research works, such as Cartesian arrow and fibration, then demonstrate adjunction properties of re-indexing functor and op-re-indexing functor. In Section 3, we make semantic model of non-indexed fibration to analyze semantics of fibred inductive data types and depict their inductive rule with universality abstractly. In Section 4, we extend non-indexed fibration to single-sorted indexed fibration on slice category, make its semantic model to analyze semantics of single-sorted indexed fibred inductive data types and depict their inductive rule. In Section 5, we develop single-sorted discrete object to indexed category, further extend single-sorted indexed fibration to many-sorted indexed fibration, make its semantic model to analyze many-sorted indexed inductive data types and depict their inductive rule. In Section 6, we study some related works in the research field of inductive data types. At last, we summarize our conclusions and discuss our future research.

2. Fibration and Opfibration

2.1. Fibration and Re-Indexing Functor

We assume readers have the categorical foundations, functor, adjunction, natural transformation, etc. Considering not making set theoretical models by mathematical logic, currently, some basic math literatures do not require all morphisms to be a set, but from the practical applications perspective of computer processing discrete objects, we deem it is reasonable to take all morphisms to be a set. If all objects and morphisms can form two sets respectively in a category, the category is called a small category, as stated by [10]. All research objects in this paper are based on small category, more details about Fibrations theory can be found in [10-12]. Let $\text{Obj} \ C$ be a set of objects for category $C$, and $\text{Mor} \ C$ a set of morphisms for category $C$. We introduce some basic concepts in this paper.

**Definition 1.** Let $P : \mathcal{T} \to \mathcal{B}$ be a functor between small categories $\mathcal{T}$ and $\mathcal{B}$, $f : C \to D \in \text{Mor} \ \mathcal{B}$, $u : X \to Y \in \text{Mor} \ \mathcal{T}$. We call morphism $u$ a Cartesian arrow of $f$ and $Y$ if $P(Y) = D$, $P(u) = f$, for $\forall Z \in \text{Obj} \ \mathcal{T}$, $v : Z \to Y \in \text{Mor} \ \mathcal{T}$ and $\forall h : P(Z) \to C \in \text{Mor} \ \mathcal{B}$, it satisfies $f \circ h = P(v)$, and there exists a unique $w : Z \to X \in \text{Mor} \ \mathcal{T}$ such that $u \circ w = v$ and $P(w) = h$.

For Cartesian arrow $u$ of $f$ and $Y$, we call $u$ lies above $f$; and similarly for $f$, $Y$ lies above $D$. If $u$ is a cone [10] in category $\mathcal{T}$, in definition 1 the Cartesian arrow $u$ is also a universal cone in $\mathcal{T}$ by the uniqueness of cones morphism $w$, namely, limit cone. Accordingly, the vertex $X$ of universal cone $u$ is the terminal object in [13] of $X$.

**Definition 2.** Let $P : \mathcal{T} \to \mathcal{B}$ be a functor between small categories $\mathcal{T}$ and $\mathcal{B}$, then if for $\forall Y \in \text{Obj} \ \mathcal{T}$ and $\forall f : C \to P(Y) \in \text{Mor} \ \mathcal{B}$ there exists a Cartesian arrow of $f$ and $Y$, we call $P$ a fibration.

By definition 2, we know that fibration is a functor that in fact ensures a large supply of Cartesian arrows. For a fibration $P : \mathcal{T} \to \mathcal{B}$, we call $\mathcal{B}$ the base category of $P$, and $\mathcal{T}$ its total category. If for an object $C$ in $\text{Obj} \ \mathcal{B}$, $\exists X \in \text{Obj} \ \mathcal{T}$, $k \in \text{Mor} \ \mathcal{T}$ is satisfied $P(X) = C$ and $P(k) = \text{id}_C$, then the subcategory $\mathcal{T}_C$ is called a fiber over $C$ in [10], and $k$ is vertical morphism.

Fiber $\mathcal{T}_C$ is actually a full subcategory of total category $\mathcal{T}$. Without loss of generality, we write $f^*_V$ for Cartesian arrow $u$ of $f$ and $Y$ in definition 1 if $f^*(Y)$ is domain of $f^*_V$, then $f^*(Y)$ lies above $C$, that is, $Y \in \text{Obj} \ \mathcal{T}_D, f^*(Y) \in \text{Obj} \ \mathcal{T}_C$. 

D. Miao et al.
Definition 3. A morphism \( f : C \to D \) in base category \( \mathcal{B} \) is extended to be a functor \( f^* : \mathcal{T}_D \to \mathcal{T}_C \) between fibers \( \mathcal{T}_D \) and \( \mathcal{T}_C \); we call \( f^* \) a re-indexing functor induced by \( f \).

\( f \) is the relationship between inductive data types in base category, and re-indexing functor \( f^* \) is a lifting of \( f \) in total category, which is related to their semantic properties. The dual concept to fibration, i.e., opfibration, is as follows.

2.2. Opfibration and Op-Re-Indexing Functor

Definition 4. Let \( P : \mathcal{T} \to \mathcal{B} \) be a functor between small categories \( \mathcal{T} \) and \( \mathcal{B} \), \( f : C \to D \in \text{Mor} \mathcal{B} \), \( u : X \to Y \in \text{Mor} \mathcal{T} \). We call morphism \( u \) an opposite Cartesian arrow of \( f \) and \( X \) if \( P(X) = C \), \( P(u) = f \), for \( \forall Z \in \text{Obj} \mathcal{T} \), \( v : X \to Z \in \text{Mor} \mathcal{T} \) and \( \forall h : D \to P(Z) \in \text{Mor} \mathcal{B} \), it satisfies \( h \circ f = P(v) \), and there exists a unique \( w : Y \to Z \in \text{Mor} \mathcal{T} \) such that \( w \circ u = v \) and \( P(w) = h \).

Similarly to definition 1, if \( u \) is a co-cone \([10]\) in category \( \mathcal{T} \), the opposite Cartesian arrow \( u \) in definition 4 is a universal co-cone \( \mathcal{T} \) by the uniqueness of co-cones morphism \( w \), namely, co-limit co-cone. Accordingly, the vertex \( Y \) of universal co-cone \( u \) is the initial object of \( u \) in \([13]\).

Definition 5. Let \( P : \mathcal{T} \to \mathcal{B} \) be a functor between small categories \( \mathcal{T} \) and \( \mathcal{B} \); if \( \forall X \in \text{Obj} \mathcal{T} \) and \( \forall f : P(X) \to D \in \text{Mor} \mathcal{B} \), there exists an opposite Cartesian arrow of \( f \) and \( X \), we call \( P \) an opfibration.

Definition 6. If the functor \( P : \mathcal{T} \to \mathcal{B} \) between small categories \( \mathcal{T} \) and \( \mathcal{B} \) is simultaneously a fibration and an opfibration, then it is a bifibration.

Without loss of generality, we write \( f^X \) for the opposite Cartesian arrow \( u \) of \( f \) and \( X \); let \( *f(X) \) be the codomain of \( f^X \), then we say \( *f(X) \) lies above \( D \), that is, \( X \in \text{Obj} \mathcal{T}_C \), \( *f(X) \in \text{Obj} \mathcal{T}_D \).

Definition 7. A morphism \( f : C \to D \) in base category \( \mathcal{B} \) is extended to be a functor \( f^* : \mathcal{T}_D \to \mathcal{T}_C \) between fibers \( \mathcal{T}_D \) and \( \mathcal{T}_C \); we call \( f^* \) an op-re-indexing functor induced by \( f \).

2.3. Adjunction Properties of Re-Indexing Functor and Op-Re-Indexing Functor

Definition 8. If \( F \downarrow G : \mathcal{C} \to \mathcal{D} \) is a pair of ad joint functors, \( \eta, \varepsilon \) is the unit and co-unit of this adjunction respectively, and for \( \forall X \in \text{Obj} \mathcal{C} \), \( \forall Y \in \text{Obj} \mathcal{D} \), \( \exists f : F(X) \to Y \in \text{Mor} \mathcal{D} \), \( \exists g : X \to G(Y) \in \text{Mor} \mathcal{C} \), then the transpose of \( f \) and \( g \) is \( G(f) \eta_X \) and \( \varepsilon_Y F(g) \) respectively.

Theorem 1. Let \( P : \mathcal{T} \to \mathcal{B} \) be a fibration between small categories \( \mathcal{T} \) and \( \mathcal{B} \); \( P \) is a bifibration iff \( \forall f : C \to D \in \text{Mor} \mathcal{B} \), the re-indexing functor \( f^* \) has a left ad joint functor \( f^\ast \) which is an op-re-indexing functor.

Proof. \( \Rightarrow \). Let \( \ast f \downarrow f^* : \mathcal{T}_C \to \mathcal{T}_D \) be a pair of ad joint functors, the unit is \( \eta \), the co-unit is \( \varepsilon \), and \( P : \mathcal{T} \to \mathcal{B} \) is a fibration between small categories \( \mathcal{T} \) and \( \mathcal{B} \). For \( \exists Y \in \text{Obj} \mathcal{T}_D \), we can construct a Cartesian arrow \( f^Y \downarrow (Y) \to Y \) whose codomain is \( Y \). \( \exists X \in \text{Obj} \mathcal{T}_C \), let \( l : X \to *f(X) \) be a morphism above \( f \), the proof that \( l \) is an opposite Cartesian arrow above \( f \) is as follows: it satisfies \( l = f^* f(X) \circ \eta_X \) by the adjunction properties of \( f \downarrow f^* \), seen from Figure 1. We write \( id \) to identify morphism, if \( g : X \to Y \) is another morphism above \( f \), let \( \phi : X \to f^* f(Y) \) be vertical morphism in \( \mathcal{T}_C \), we get \( P(\phi) = id_D \). By definition 1 we know \( g = f^Y \circ \phi \), Cartesian arrow \( f^Y \) is an universal cone, whose universal property ensures \( \phi \) is the unique morphism from \( g \) to \( f^Y \). We write \( \phi \) for the transpose of \( \phi \) under the adjunction \( \phi \downarrow \phi^* \), then \( \phi^Y \circ \phi^* f(X) \to Y \), \( f^* f(X) \circ \eta_X = \phi \). Universal property of universal cone \( f^Y \) ensures the unique existence of \( f^* f(X) \), and it satisfies \( \phi \circ f^Y = f^Y \circ f^* f(X) \). Above all, there exist two equalities, that are, \( \phi \circ l = \phi = f^Y \circ f^* f(X) \circ \eta_X = f^Y \circ f^* f(X) \circ \eta_X = f^Y \circ f(X) \), \( g = \phi \circ l \), then the transpose of \( \phi \) is the unique morphism from \( l \) to \( g \), and \( P(\phi) = id_D \). We thus prove that \( l \) is an opposite Cartesian arrow \( f^Y \) above \( f \) by definition 4.
We assume $g : X \to Y \in \text{Mor} \ T$ lies above $f$, write $T_C(X, f^\ast(Y))$ for set composed of morphisms above $C$ in fiber $T_C$, $T_D(\ast f(X), Y)$ for set composed of morphisms above $D$ in fiber $T_D$. For $\forall k : X' \to X \in \text{Mor} \ T_C$, $\forall h : Y \to Y' \in \text{Mor} \ T_D$, because $P : T \to B$ is a bifibration, this gives us one-to-one corresponding map $\phi_{XY} : T_D(\ast f(X), Y) \to T_C(X, f^\ast(Y))$. We write $k\op : X \to X' \in \text{Mor} \ T_C$ for an opposite morphism of $k$, it satisfies that $k\op \circ f_X^\op = f_X^\op \circ f_Y^\op$ and $f_X^\op \circ id_Y = f_Y^\op \circ id_Y$, so the left part of diagram in Figure 2 commutes. Similarly, it also satisfies $id_X \circ f_X^\op = f_X^\op \circ id_X$ and $f_X^\op \circ h = f_Y^\op \circ h$, i.e., the right part of diagram in Figure 2 commutes, so $\phi_{XY}$ is natural isomorphism. We thus prove $f \dagger f$ by definition of adjoint functors in [13].

\begin{align*}
\text{Figure 1. Proof of opposite Cartesian arrow.} \\
\text{Figure 2. Proof of adjunction properties.}
\end{align*}

Theorem 1 provides a convenient condition for judgment of bifibration. At the same time, it also amalgamates well adjunction property of re-indexing functor $f^\ast$ and op-re-indexing functor $f^\ast$ in the framework of bifibration.

### 3. Semantic Properties and Inductive Rule of Fibered Inductive Data Types

Fibered inductive data types, such as natural numbers and finite partial order sets, are usual inductive data types with fibered structures from the view of Fibrations theory. This section presents a semantic model of non-indexed bifibration by Fibrations theory, which analyzes semantic properties and describes inductive rule of fibered inductive data types.

#### 3.1. Semantic Model of Non-Indexed Fibration

**Definition 9.** Let $P : T \to B$ and $P' : T' \to B$ be two fibrations between small categories, a fibered functor $F : T \to T'$ from $P$ to $P'$ above base category $B$ satisfies diagram commutes, that is, $P = P' \circ F$, and we call $F$ preserves Cartesian arrow.

**Definition 10.** Let $F : T \to T'$ and $G : T' \to T$ be two fibered functors above base category $B$, we call $G$ a right fibered adjoint functor to $F$, and $F \dashv G$ is a pair of fibrad adjunction above $B$, iff $G$ is a right adjoint functor to $F$, the unit or co-unit of $F \dashv G$ is vertical.

Definition 9 and definition 10 lift standard category structures to fibered structures. It is easy to process many practical problems with discrete structures in computer science effectively, that is, semantic properties of inductive data types in programming are mapped to their corresponding fibers in total category, which is closely related to inductive data types to their semantic properties further. At the same time, what is more important is that describing inductive rules with universality and program logic abstractly by the tools of fibered functor and fibered adjunction does not depend on particular computing environments, which improves the cohesion of inductive data types, and further enhances independence of program languages.

**Definition 11.** Let $P : T \to B$ be a functor between small categories $T$ and $B$, $F : B \to B$ is an endo-functor in base category $B$, the lifting of $F$ with respect to $P$ is an endo-functor $F^\perp : T \to T$ in total category $T$. If there exists diagram commute, i.e., $PF^\perp = FP$, then we call $P$ a non-indexed fibration.

Definition 11 makes the semantic model of non-indexed fibration. For $\forall D \in \text{Obj} B$, if $\exists ! D \in \text{Obj} T_D$, which is the terminal object of fiber $T_D$, and $\forall f : C \to D \in \text{Mor} B$, $f^\ast(1_D)$ is then the terminal object of fiber $T_C$, namely, re-indexing functor $f^\ast$ preserves terminal objects, non-indexed fibration $P$ is called having fibered terminal objects.
3.2. Semantic Properties of Fibered Inductive Data Types

**Definition 12.** Let \( P : \mathcal{T} \to \mathcal{B} \) be a non-indexed fibration, functor \( T : \mathcal{B} \to \mathcal{T} \) maps \( \forall C \in \text{Obj} \mathcal{B} \) to the terminal object in fiber \( \mathcal{T}_C \), \( T \) is called a truth functor of \( P \). If \( TF \cong F^\perp T \), we call \( F^\perp \) a lifting of \( F \) with respect to \( P \) preserving truth.

We write that \( 1_\mathcal{B} \) and \( 1_P \) are terminal objects of base category \( \mathcal{B} \) and total category \( \mathcal{T} \) respectively, we thus get \( P(1_P) = 1_\mathcal{B} \). For \( \forall C \in \text{Obj} \mathcal{B} \) there exists a unique morphism \( u : C \to 1_\mathcal{B} \) in base category \( \mathcal{B} \), so we have an isomorphism expression, that is, \( T(C) \cong u^* \). For \( \forall f : C \to D \in \text{Mor} \mathcal{B} \), we have an isomorphism expression \( f^*(T(D)) \cong T(C) \), and the truth functor \( T \) maps \( f \) to a Cartesian arrow \( f^\perp \) in total category \( \mathcal{T} \). If the truth functor \( T \) is full and faithful [13], then it is a fibred right adjoint functor to non-indexed fibration \( P \) [8].

For each object \( C \) in category \( \mathcal{B} \), a \( F \)-algebra \( (C, i : F(C) \to C) \) is constructed by the endofunctor \( F \), we call \( C \) the carrier of this \( F \)-algebra. The morphism between \( (C, i) \) and another \( F \)-algebra \( (D, j : F(D) \to D) \) is a morphism \( f : C \to D \) between their carriers, which satisfies diagram commutes, that is, \( f \circ i = j \circ F(f) \). \( F \)-algebra category is constituted of \( F \)-algebras and their morphisms, we write \( \text{Alg}_F \) for it. If the initial \( F \)-algebra \( (\mu F, in : F(\mu F) \to \mu F) \) exists, it is up to a unique isomorphism, whose properties of this unique isomorphism determined by its initial universal property is our main tool for researching semantics and inductive rules of inductive data types. An inductive data type \( \mu F \) as the carrier of initial \( F \)-algebra, is least fixed point of functor \( F \). The functor \( F \) denotes syntax construction of \( \mu F \), and its morphism \( in \) gives a kind of semantic interception of \( \mu F \) under this syntax construction.

A \( F \)-algebra \( (C, i : F(C) \to C) \) is mapped to a \( F^\perp \)-algebra \( (T(C), T(i) : T(F(C)) \cong F^\perp(T(C)) \to T(C)) \) by the truth functor \( T \). Accordingly, \( T(\mu F) \) is the carrier of initial \( F^\perp \)-algebra, i.e., truth functor preserves initial objects. We write \( \text{Alg}_T \) for the functor from \( F \)-algebra category \( \text{Alg}_F \) to \( F^\perp \)-algebra category \( \text{Alg}_F^\perp \), and define \( \text{Alg}_T(\mu F) \) \( \cong T \). Objects and morphisms in base category \( \mathcal{B} \) with respect to non-indexed fibration \( P \) are mapped to those responded in total category \( \mathcal{T} \) by the truth functor \( T \), which further makes connections from \( \text{Alg}_F \) to \( \text{Alg}_F^\perp \) via \( \text{Alg}_T \) and isomorphism property of composed functors of definition 12. The initial \( F^\perp \)-algebra \( (T(\mu F), in^\perp : F^\perp(T(\mu F)) \to T(\mu F)) \) in total category \( \mathcal{T} \) with respect to the non-indexed fibration \( P \), is the homomorphism image of \( in \) which is the morphism of initial \( F \)-algebra \( (\mu F, in) \) by the action of \( \text{Alg}_T \), that is, \( \text{Alg}_T(in) = in^\perp \). Initial property of initial \( F \)-algebra ensures that \( in^\perp \) is up to unique isomorphism, whose existence provides extreme convenience for analyzing semantic properties and depicting inductive rule of fibered inductive data types.

**Definition 13.** Let \( P : \mathcal{T} \to \mathcal{B} \) be a non-indexed fibration between small categories \( \mathcal{T} \) and \( \mathcal{B} \). \( T : \mathcal{B} \to \mathcal{T} \) is a truth functor of \( P \). If \( \{ \} \) is a right adjoint functor to \( T \), namely, \( T \downarrow \{ \} \), then \( \{ \} \) is called a comprehension functor of \( P \).

Let \( \sigma : \{ \} \to P \) be a natural transformation, \( F \sigma \) is also a natural transformation by composed theorem of natural transformations, as stated in [13], then for \( \forall X \in \text{Obj} \mathcal{T}, \{ (F \sigma_X) : T(F(X)) \to TFP(X) \} \) is an op-re-indexing functor induced by \( F \sigma_X \), and \( F^\perp(X) = \{ (F \sigma_X)(T(F(X))) \in \text{Obj} \mathcal{T}, X \} \) is the extension of \( X \) over the comprehension functor \( \{ \} \). For \( \forall X : X \to X' \in \text{Mor} \mathcal{T}, F^\perp(k) = F^\perp(X) = F^\perp(X') = \{ (F \sigma_X)(T(F(X'))) \} = \{ (F \sigma_X)(T(F(X))) \} \), that is, \( F^\perp(k) \in \{ F(P(k)) \} \), then \( F^\perp(k) \) is an element of re-indexing functor morphism \( \{ F(P(k)) \} \).

Similar to \( \text{Alg}_T \), we write \( \text{Alg} \{ \} \) for the functor from \( F^\perp \)-algebra category \( \text{Alg}_F^\perp \) to \( F \)-algebra category \( \text{Alg}_F \), define \( \text{Alg} \{ \} \overset{\text{def}}{=} \{ \} \), it satisfies \( \text{Alg}_T(\mu F) + \text{Alg} \{ \} \overset{\text{def}}{=} T \) by adjunction property of \( T \downarrow \{ \} \) from definition 13. For each \( F^\perp \)-algebra \( (X, j : F^\perp(X) \to X), \text{Alg} \{ \}(j) = F(X) \to \{ X \} \), that is, \( \text{Alg} \{ \}(j) = \{ j \} \), seen from Figure 3. By previous analysis we also know that \( \{ j \} \) is an isomorphism image of \( j \) by the action of functor \( \text{Alg} \{ \} \).

If \( g : X \to T(C) \) is a \( F^\perp \)-algebra morphism
from $j$ to $\text{Alg}(T)(i)$, then a $F$-algebra morphism $h : \{X\} \to C$ from $\text{Alg}\{\{\}\}(j)$ to $i$ is a $F$-algebras homomorphism above $g$. Accordingly, $g$ is a $F^\perp$-algebras homomorphism above $h$. The right ad joint functor $\text{Alg}\{\{\}\}$ to $\text{Alg}(T)$ establishes a kind of intuitively mutual induction relations between $F^\perp$-algebra whose carrier is $X$ and $F$-algebras whose carrier is $\{X\}$, which further supplies a succinct and coherent modeling method for describing inductive rules of inductive data types formally, whose initial algebra carrier is $\mu F$.

\[
\begin{align*}
\text{Alg}(T) & \xrightarrow{\mu} \text{Alg}(\{\{\}\}) \xrightarrow{i} F \text{Alg}(\{\{\}\}) \\
&T \text{Alg}(\{\{\}\}) \xrightarrow{j} \text{Alg}(T) \\
F(\text{Alg}(\{\{\}\})) & \xrightarrow{F(i)} F(\text{Alg}(T)) \\
T(F(C)) & \xrightarrow{T(j)} F^\perp(T(C)) \\
F(X) & \xrightarrow{F(j)} F(\text{Alg}(T)) \\
\end{align*}
\]

Figure 3. Adjunction properties of $\text{Alg}(T) \vdash \text{Alg}\{\{\}\}$.

3.3. Inductive Rule of Fibered Inductive Data Types

For semantic model of non-indexed fibration with comprehension functor, formal description of inductive rules and semantic analysis of fibered inductive data types are coherent. A non-indexed fibration $P : \mathcal{T} \rightarrow \mathcal{B}$, its comprehension functor $\{\{\}\}$ is right ad joint to its truth functor $T$, i.e., $P \dashv T \vdash \{\{\}\}$. Let $F$ be an endo-functor in base category $\mathcal{B}$, and $\mu F$ is the carrier of initial $F$-algebra, then each preserving truth lifting $F^\perp$ of $F$ with respect to $P$ has an inductive rule [9], which provides a sound foundation for judging on validity of inductive rules generated from fibered inductive data types for $F^\perp$ applied by initial $F$-algebra. Accordingly, if semantic model of non-indexed fibration $P$ defines and utilizes comprehension functor to compute recursively on fibered inductive data types, then its inductive rule based on $F$-algebra is valid when processing semantic logic analysis in programming.

In the framework of Fibrations theory, describing inductive rule of fibered inductive data types with universality is as follows: we consider recursive computation of fibered inductive data types firstly, which arises from initial algebra semantics categorically in [2]. Let inductive data types $\mu F$ be the carrier of initial $F$-algebra, we utilize endo-functor $F$ in base category $\mathcal{B}$ to construct a recursive operation on fibered inductive data types $\text{fold} : (F(C) \rightarrow C) \rightarrow \mu F \rightarrow C$. For each $F$-algebra $\langle C, i : F(C) \rightarrow C \rangle$, by the action of recursive operation $\text{fold}$, $\text{fold} i$ is the map sending $i$ to the unique $F$-algebra morphism $\text{fold} i : \mu F \rightarrow C$ from initial $F$-algebra morphism $i$ to $\mu F$, seen from Figure 4. The essence of $\text{fold}$ stems from initial algebra semantics as a parameterized recursive operation on inductive data types, which possesses lots of good properties such as correct semantics, flexible extension and succinct expression.

\[
\begin{align*}
\mu F \xrightarrow{\text{fold } i} C \\
\end{align*}
\]

Figure 4. $F$-algebra morphism.

Henceforth, the following isomorphism equations hold: $TF(C) \cong F^\perp T(C), TF(\mu F) \cong F^\perp T(\mu F)$, and by the property of truth functor $T$ preserving initial objects, $T(\mu F)$ is the carrier of initial $F^\perp$-algebra, we write $\mu F^\perp = T(\mu F)$, $X = T(C) \in \text{Obj } \mathcal{T}$. Similarly, we can construct a recursive operation $\text{fold} : (F^\perp(X) \rightarrow X) \rightarrow \mu F^\perp \rightarrow X$, which depicts semantics of fibered inductive data types in total category $\mathcal{T}$ by the tool of preserving truth lifting $F^\perp$ of $F$, see Figure 5. So for $\forall C \in \text{Obj } \mathcal{B}, X \in \text{Obj } \mathcal{T} C$, we have the inductive rule of fibered inductive data types with universality, that is,

\[
\text{IndGen} : (F^\perp(X) \rightarrow X) \rightarrow T(\mu F) \rightarrow X.
\]

If $(X, j : F^\perp(X) \rightarrow X)$ is a $F^\perp$-algebra over $F$-algebra $(C, i : F(C) \rightarrow C)$, then $\text{IndGen } X j : T(\mu F) \rightarrow X$ is a $F^\perp$ algebra homomorphism over $\text{fold } i$.

\[
\begin{align*}
F^\perp(\mu F^+) \xrightarrow{F(\text{fold } i)} F^\perp(X) \\
\mu F^+ \xrightarrow{\text{fold } j} X
\end{align*}
\]

Figure 5. $F^\perp$-algebra morphism.
3.4. Example Analysis of Fibered Inductive Data Types

**Example 1.** Let natural number type Nat be the carrier μF of initial F-algebra in base category B, we write 1 for the terminal object. For ∀N ∈ Obj B, it is to be said F : N → 1+N, then its inject in1(1) = 0 is the minimum natural number, its inject in2(N) = N + 1 is a successful function. For any natural number property X ∈ Obj T N in total category T with respect to the non-indexed fibration P, e.g., transitivity, compatibility and completeness, an induction (X(in1(N)) → (X(N) → X(in2(N)))) → X(N) holds for property X of Nat. For each F-algebra (N, h : F(N) → N), it is lifted to be a F⊥-algebra (X, k : F⊥(X) → X) via the non-indexed fibration P, which satisfies diagram commutes, that is, FP(X) = PF⊥(X). A recursive operation fold h is defined by the initial property of initial F-algebra, it executes to be judgment of fibered inductive data type Nat: And another recursive operation defined by the initial property of initial F⊥-algebra describes semantics of Nat. If k lies over h, then IndGen X k is a F⊥-algebra homomorphism over fold h, and when it iterates each object in total category T, we can get the semantics set describing properties of Nat, that is {X(N)|∀N ∈ Obj B}.

Compared with traditional methods including algebras and category theory, inductive rule IndGen depicting recursive computing of Nat is established based on semantic model of non-indexed fibration P given by example 1, which presents a succinct descriptive pattern for semantic properties and program logic of fibered inductive data type Nat, especially in functional program languages, such as ML and Haskell, inductive rule IndGen makes code fragments easier to read, write and understand.

4. Semantic Properties and Inductive Rule of Single-Sorted Indexed Inductive Data Types

As a simple inductive data type, fibered inductive data type is limited to analyze semantics and depict inductive rule, but indexed inductive data types are a kind of inductive data types whose syntax construction and semantic computation are stronger than fibered inductive data types, and can process more complex data structure. Dybjer in [14] and Morris in [15] obtained some prominent achievements in the field of initial algebra semantics, but there has been few works about inductive rules of indexed inductive data types nowadays. This section presents a semantic model of single-sorted indexed fibration by fibrations theory, analyzes semantic properties of some classical single-sorted indexed inductive data types, such as streams, lists, trees, and so on. On the basis of [15], and borrowing from research production of Ghani et al. [16], we present an inductive rule of single-sorted indexed inductive data types with universality.

4.1. Semantic Model of Single-Sorted Indexed Fibration

**Theorem 2.** Let P : T → B be a bifibration or bifibration between small categories T and B, T : B → T is the truth functor of P. ∃I ∈ Obj B is a discrete indexed object in base category B, let single-sorted indexing functor P/I : T/T(I) → B/I be P/I(u) = P(u) : P(Y) → I ∈ Obj B/I for ∀u : Y → T(I) ∈ Obj T/T(I), then single-sorted indexing functor P/I is also a bifibration or bifibration.

**Proofs.** For ∀f : C → D ∈ Mor B, there exists a Cartesian arrow f ⊥ : f*(X) → X above f with respect to bifibration P such that P(X) = D, and exists an unique morphism w : T P(I) → f*(X) such that v = f ⊥ w and P(v) = f  h, as seen from Figure 6. Let α : D → I ∈ Obj B/I, β : C → I ∈ Obj B/I, we thus have two morpthisms in the slice category B/I, i.e., γ : P(u) → α = P(Y) → D ∈ Mor B/I and δ : P(u) → β = P(Y) → C ∈ Mor B/I, which satisfy diagram commutes, that is γ = f δ. In total category T/T(I) of functor P/I, s : X → T(I) ∈ Obj T/T(I), t : f ⊥(X) → T(I) ∈ Obj T/T(I), so g : u → s = Y → X ∈ Mor T/T(I), there exists a unique morphism k : u → t = Y → f ⊥(X) such that g = f ⊥ k, then f ⊥ is a Cartesian arrow of f with respect to functor P/I by definition 1, namely, if P is a bifibration, single-sorted indexed functor P/I is also a bifibration.
We assume that \( m : Z \to T(I) \in \text{Obj } \mathcal{T}/T(I) \), then \( P/I(m) = \alpha \) by the definition of single-sorted indexing functor \( P/I \). Let \( f^p_f : Z \to *f(Z) \) be an opposite Cartesian arrow with respect to \( P \) above \( f \), see from Figure 7. In slice category \( \mathcal{B}/I \) the diagram commutes \( \alpha = \beta \circ f \), there exists a unique morphism \( n : *f(Z) \to T_P(I) \) in total category \( \mathcal{T}/T(I) \) with respect to the functor \( P/I \) such that \( m = n \circ f^p \), by definition \( \mathcal{B}/I \) is a single-sorted indexed fibration, then \( P/I \) is also an opfibration.

Above all, if \( P \) is a bifibration, then the single-sorted indexed functor \( P/I \) is also a bifibration.

Theorem 2 demonstrates single-sorted functor \( P/I \) and \( P \) have the same properties of fibrations or bifibrations, for \( \forall \alpha : C \to I \in \text{Obj } \mathcal{B}/I \), fiber \( \mathcal{T}_C \) above \( C \) is up to isomorphism with indexed fiber \( (\mathcal{T}/T(I))_\alpha \) above \( \alpha \) in [16], i.e., \( \mathcal{T}_C \cong (\mathcal{T}/T(I))_\alpha \). We construct domain functor \( \text{dom} : \mathcal{B}/I \to \mathcal{B}, \text{dom}(\alpha : C \to I) = C \in \text{Obj } \mathcal{B} \). By the property of pullback preserving structures, namely, the pullback of fibration along any functor is also a fibration in [10], and the pullback of fibration \( \mathcal{T} \to \mathcal{B} \) along \( \text{dom} \) forms a single-sorted indexed fibration \( P/I : \mathcal{T}/T(I) \to \mathcal{B}/I \), which preserves fibered terminal objects [15], so it has property of preserving truth functor, that is, if fibration \( P \) has truth functor, then single-sorted indexed fibration \( P/I \) constructed by its pullback also has truth functor. The semantic model of single-sorted indexed fibration is as follows:

**Definition 14.** Let \( P : \mathcal{T} \to \mathcal{B} \) be a bifibration with truth functor \( T \) and comprehension functor \( \{-\}, P/I : \mathcal{T}/T(I) \to \mathcal{B}/I \) is a single-sorted indexed fibration of \( P \). We write \( T_{P/I} \) for truth functor of \( P/I \), a right adjoint functor \( \{-\}_{P/I} \) to \( T_{P/I} \) called to be a comprehension functor of \( P/I \). Let \( F \) be an endo-functor in base category \( \mathcal{B}/I \), \( F^\top \) is called a preserving truth lifting of \( F \) with respect to \( P/I \) in total category \( \mathcal{T}/T(I) \), which satisfies diagram commutes, that is, \( (P/I)F^\top = F(P/I) \) such that \( T_{P/I}F \cong F^\top T_{P/I} \) and \( F\{-\}_{P/I} \cong \{-\}_{P/I}F^\top \).

### 4.2. Semantic Properties of Single-Sorted Indexed Inductive Data Types

For \( \forall \alpha : C \to I \in \text{Obj } \mathcal{B}/I \), a \( F \)-algebra \( (\alpha, \phi : F(\alpha) \to \alpha) \) is constructed by the action of endo-functor \( F \), and the truth functor \( T_{P/I} \) of single-sorted indexed fibration \( P/I \) maps \( (\alpha, \phi) \) to \( \alpha \)-algebra \( (T_{P/I}(\alpha), T_{P/I}(\phi) : T_{P/I}(F(\alpha)) \cong F^\top T_{P/I}(\alpha)) \rightarrow T_{P/I}(\alpha) \). Let \( \mu F \) be the carrier of initial \( F \)-algebra, by the property of truth functor \( T_{P/I} \) preserves initial objects we know that \( T_{P/I} \mu F \) is the carrier of initial \( F^\top \)-algebra \( (T_{P/I}(\mu F), \text{in}^\top : F^\top(T_{P/I}(\mu F)) \rightarrow T_{P/I}(\mu F)) \). Similarly for subsection 2.2, we write \( \text{Alg}(T_{P/I}) \) for the functor from \( F \)-algebra category \( \text{Alg}_F \) to \( F^\top \)-algebra category \( \text{Alg}_{F^\top} \), define \( \text{Alg}(T_{P/I}) \cong T_{P/I} \). We thus have \( \text{Alg}(T_{P/I}) \cong T_{P/I} \), then \( \text{in}^\top \) is the homorphism image of \( \text{in} \), which is the morphism of initial \( F \)-algebra \( \mu F/(in) \) by the action of the functor \( \text{Alg}(T_{P/I}) \).

For each \( F^\top \)-algebra \( (Y, \phi : F^\top(Y) \to Y) \), the comprehension functor \( \{-\}_{P/I} \) of single-sorted indexed fibration \( P/I \) maps \( (Y, \phi) \) to a \( F \)-algebra \( \{Y \}_{P/I}, \phi \}_{P/I} : \{F^\top(Y)\}_{P/I} \cong F\{Y \}_{P/I} \rightarrow \{Y \}_{P/I} \), seen from Figure 8. If \( n : Y \to T_{P/I}(\alpha) \) is a \( F^\top \)-algebra morphism from \( \phi \) to \( T_{P/I}(\phi) \), we then have an \( F \)-algebra homomorphism \( m : \{Y \}_{P/I} \rightarrow \alpha \) over \( n \) sending \( \{\phi \}_{P/I} \) to \( \phi \). Similarly, \( n \) is a \( F^\top \)-algebra morphism.
homomorphism over \( m \). We define the functor \( Alg\{-\}_{P/I} \) from \( Alg_{F} \) to \( Alg_{G} \) and \( Alg\{-\}_{P/I} \) presents an intuitional mutual derivation relationship between \( F^\top \)-algebra \( Y \) as its carrier and \( F \)-algebra \( \{Y\}_{P/I} \) as its carrier, which provides a succinct and consistent modeling method for describing inductive rule of single-sorted indexed inductive data types and \( \mu F \) as the carrier of initial \( F \)-algebra. That is, if the functor \( Alg(T_{P/I}) \) preserves initial objects, then the preserving truth lifting \( F^\top \) of \( F \) with respect to \( P/I \) generates a sound inductive rule.

\[
\begin{align*}
&\begin{array}{c}
\text{Alg}_{G} \\
\text{Alg}_{G}^r \\
\text{Alg}_{G}^l \\
\end{array} \\
&\begin{array}{c}
\text{Alg}_{F} \\
\text{Alg}_{F}^r \\
\text{Alg}_{F}^l \\
\end{array}
\end{align*}
\]

\[\begin{align*}
F(\alpha) &\rightarrow \alpha \\
Y &\rightarrow Y
\end{align*}\]

By definition 14, we get the following two isomorphism expressions: \( T_{P/I}(F(\alpha)) \cong F^\top(T_{P/I}(\alpha)) \), \( T_{P/I}(F(\mu F)) \cong F^\top(T_{P/I}(\mu F)) \). The truth functor \( T_{P/I} \) preserves initial objects, then \( T_{P/I}(\mu F) \) is the carrier of initial \( F^\top \)-algebra. We write \( \mu F^\top = T_{P/I}(\mu F) \), and \( Y = T_{P/I}(\alpha) \). Applying endo-functor \( F^\top \) to construct a recursive operation \( fold : (F^\top(Y) \rightarrow Y) \rightarrow \mu F^\top \rightarrow Y \) on single-sorted indexed inductive data types in total category \( \mathcal{I}/T(I) \), for each \( F^\top \)-algebra \( q : F^\top(Y) \rightarrow Y \), \( fold \ q \ maps \ q \ to \ a \ unique \ F^\top -algebra \ morphism \ fold q : \mu F^\top \rightarrow Y \) from initial \( F^\top \)-algebra \( (\mu F^\top, in : F(\mu F^\top) \rightarrow \mu F^\top) \) to \( (Y, q) \). For \( \forall \alpha \in \mathcal{I}/\mathcal{I}, Y \in \mathcal{I}/T(I) \), inductive rule of single-sorted indexed inductive data types with universality is as follows:

\[\text{Ind}_{Gen} : (F^\top(Y) \rightarrow Y) \rightarrow T_{P/I}(\mu F) \rightarrow Y.\]

If \( (Y, q : F^\top(Y) \rightarrow Y) \) is a \( F^\top \)-algebra over the \( F \)-algebra \( (\alpha, r : F(\alpha) \rightarrow \alpha) \), then \( \text{Ind}_{Gen} Y q \) is a \( F^\top \)-algebra homomorphism over \( fold \ r \).

### 4.3. Inductive Rule of Single-Sorted Indexed Inductive Data Types

Let \( P : \mathcal{I} \rightarrow \mathcal{B} \) be a bifibration with truth functor \( T \) and comprehension functor \( \{-\} \), \( \forall I \in \mathcal{I} \), \( \forall I \in \mathcal{B} \) is a discrete indexed object in base category \( \mathcal{B} \). \( F \) is an endo-functor in base category \( \mathcal{B}/I \), and the carrier of initial \( F \)-algebra is \( \mu F \), then each preserving truth lifting \( F^\top \) of \( F \) with respect to single-sorted indexed fibration \( P/I \) of \( P \) has an inductive rule [16], which further ensures the validity of inductive rules generated on single-sorted indexed inductive data types by single-sorted indexed fibration. The following is the inductive rule of single-sorted indexed inductive data types with universality presented by us in the framework of Fibrations theory.

\[\forall \alpha : C \rightarrow I \in \mathcal{B}/I, \mu F \in \mathcal{B}/I, \text{a recursive operation} \ fold : (F(\alpha) \rightarrow \alpha) \rightarrow \mu F \rightarrow \alpha \text{ is constructed on single-sorted indexed inductive data types in base category} \mathcal{B}/I \text{ by } F. \text{ For each } F \text{-algebra } r : F(\alpha) \rightarrow \alpha, \text{ fold } r \text{ maps } r \text{ to unique } F \text{-algebra morphism } fold r : \mu F \rightarrow \alpha \text{ from initial } F \text{-algebra } (\mu F, in : F(\mu F) \rightarrow \mu F) \text{ to } (\alpha, r).\]

### 4.4. Example Analysis of Single-Sorted Indexed Inductive Data Types

#### Example 2

Element types of single-sorted in-dexed inductive data types including streams and infinite lists are designated by indexed object \( I \), such as natural number \( Nat \), integer \( Int \), character \( Char \) and so on, \( \forall I \in \mathcal{B} \). For any stream \( Stream(I) \) be the carrier \( \mu F \) of initial \( F \)-algebra in base category \( \mathcal{B}/I \), for each stream property \( Y \in \mathcal{B}/T(I) \) in total category \( \mathcal{B}/T(I) \) on single-sorted indexed fibration \( P/I \), such as merging and reversing, there exists an induction for property of stream \( Stream(I) : (Y(head(\alpha)) \rightarrow (Y(\alpha) \rightarrow Y(tail(\alpha)))) \rightarrow Y(\alpha) \). For any \( F \)-algebra \( (\alpha, r : F(\alpha) \rightarrow \alpha) \), it is lifted to be an \( F^\top \)-algebra \( (Y, q : F^\top(Y) \rightarrow Y) \) with respect to \( P/I \) such that \( F(P/I)(Y) = (P/I)F^\top(Y) \). Initial property of initial \( F \)-algebra defines a recursive operation \( fold \ r \) on \( Stream(I) \), which executes judgment...
of data type of $\text{Stream}(I)$; a recursive operation by initial property of initial $F^\top$-algebra describes semantics of $\text{Stream}(I)$. If $q$ lies above $r$, then $\text{Ind}^q_{\text{Gen}}Yq$ is an $F^\top$-algebra homomorphism above $\text{fold} r$, and when it iterates each object in total category $\mathbb{T}/T(I)$ with respect to single-sorted indexed fibration $P/I$, finally we obtain the semantics set $\{Y(\alpha)|\forall \alpha \in Obj \mathbb{B}/I\}$ depicting properties of $\text{Stream}(I)$.

In the research of semantic computing and program logic for programming, codes describing procedure of input/output based on streams or infinite lists are a dynamic executing process. Traditional methods including algebras and domain theory are difficult to effectively manage formal semantics of streams whose complex single-sorted indexed inductive data types are dynamic process of input/output. Example 2 establishes a semantic model of single-sorted indexed inductive data types by Fibrations theory, analyzes semantic properties of stream deeply, and depicts inductive rule of stream abstractly, which lays strong mathematical foundations for semantic computing and program logic in programming.

5. Semantic Properties and Inductive Rule of Many-Sorted Indexed Inductive Data Types

Modeling by slice category $\mathbb{B}/I$ processes to analyze semantic properties and depict inductive rule of single-sorted indexed inductive data types indexed by $I$ well, but $I$ only aims at single-sorted indexed inductive data types, it is difficult to manage effectively more complex many-sorted indexed inductive data types, e.g., mutual recursive. On the basis of ahead works, we extend discrete indexed object $I$ to indexed category $C$, present semantic model of many-sorted indexed fibration, describe many-sorted indexed inductive data types in base category $\mathbb{B}$ indexed by $Obj C$, make semantic logic model of many-sorted indexed inductive data types in indexed category $C$ by the fibration $G: \mathbb{B} \rightarrow C$, choose different program logics pointing to different indexes.

5.1. Semantic Model of Many-Sorted Indexed Fibration

Let $P: \mathbb{T} \rightarrow \mathbb{B}$ and $G: \mathbb{B} \rightarrow C$ are two fibrations between small categories, by the composed property that composing of two fibrations is also a fibration in $[10]$, $GP$ is a fibration. For $\forall a \in Obj C$, $T_a$ is a fiber in total category $\mathbb{T}$ on fibration $GP$ over $a$. The restriction $P_a: T_a \rightarrow B_a$ of $P$ at $a$ is a pullback of $P$ along including functor $\text{Inc}: B_a \rightarrow \mathbb{B}$, and $B_a$ is a fiber in total category $\mathbb{B}$ on fibration $G$ over $a$, then by the property of pullbacks preserving structure, $P_a$ is also a fibration. Different fibrations $P_a$ processes different indexed objects $a$, if $P$ is an opfibration or bifibration, then its restriction $P_a$ is also an opfibration or bifibration. Henceforth, if $P$ has truth functor, then $P_a$ also has truth functor, write $T_a$ for it. In fact, $P_a$ is a subfibration of $P$ in $[15]$, i.e., $P_a$ and $P$ have the same fibration structures and logical properties.

For a bifibration $P$, by the right adjunction property that re-indexing functor preserves terminal objects, when $a$ iterates each indexed object in indexed category $C$ the set of $T_a$ constructs the truth functor $T$ of fibration $P$, namely, $T = \{T_a|\forall a \in Obj C\}$. But $F: \mathbb{B}_a \rightarrow \mathbb{B}_a$ is indeed an endo-functor in fiber $\mathbb{B}_a$, in base category $\mathbb{B}$, so whether its lifting $F_\mathbb{B}_a^C$ is also an endo-functor in total category $\mathbb{T}$ is not decidable. Similar to undesirability of $F_\mathbb{B}_a^C$, that each restriction $P_a$ of $P$ at $a$ has truth functor and comprehension functor cannot decide $P$ itself has a truth functor and a comprehension functor; on the contrary, what $P$ has truth functor and comprehension functor cannot also decide its each restriction $P_a$ at $a$ has truth functor and comprehension functor. In the following works we introduce the definition of fibered fibration and demonstrate the decidability of $P$ and its restriction $P_a$ on the existences of truth functor and comprehension functor based on $[16]$.

Definition 15. Let $P: \mathbb{T} \rightarrow \mathbb{B}$ and $G: \mathbb{B} \rightarrow C$ are two fibrations between small categories, $T: \mathbb{B} \rightarrow \mathbb{T}$ is a truth functor of $P$. If $T$ has a fibered right adjoint functor $\{-\}: GP \rightarrow G$, then $P$ is called a fibered fibration with truth functor $T$ and comprehension functor $\{-\}$ over $G$.

From definition 10 and definition 9 we know that truth functor $T: G \rightarrow GP$ of $P$ is a fibered fibration, so we can judge that it is equivalent.
to what $P$ is a fibered fibration over $G$ and what $P$ is a fibration with truth functor and comprehension functor. Then, by the demonstration of theorem 3 below we research deeply the decidability of fibered fibration $P$ and its restriction $P_a$ at $a$ on the existence of truth functor and comprehension functor.

**Theorem 3.** Let $P : \mathcal{T} \to \mathcal{B}$ and $G : \mathcal{B} \to \mathcal{C}$ be two fibrations between small categories, $P$ is a fibered fibration over $G$, then for $\forall a \in \text{Obj} \mathcal{C}$, a restriction $P_a : \mathcal{T}_a \to \mathcal{B}_a$ of $P$ at $a$ is also a fibered fibration.

**Proofs.** Let fibered adjunction $T \vdash \{ - \}$ be truth functor and comprehension functor of the fibration $P$ respectively, for $\forall a \in \text{Obj} \mathcal{C}$, $T_a$ and $\{ - \}_a$ are the restriction of $T$ and $\{ - \}$ at $a$ respectively. For any $f : a \to b \in \text{Mor} \mathcal{C}$, $f_T : f^*(Y) \to Y \in \text{Mor} \mathcal{B}_a$ is a Cartesian arrow of $f$ on fibration $G$, now we prove that $T_a(f_Y)$ is also a Cartesian arrow of $f$ on fibration $GP$, i.e., truth functor $T_a$ preserves Cartesian arrow. For any $g : c \to a \in \text{Mor} \mathcal{C}$, let $l : X \to T_a(Y) \in \text{Mor} \mathcal{B}_a$ lies above $fg$, we can see it from Figure 9.

![Figure 9. Truth functor $T_a$ preserves Cartesian arrows.](image)

Let $\eta : 1_{\mathcal{T}_a} \to T_a\{ - \}_a$ and $\varepsilon : \{ - \}_a T_a \to 1_{\mathcal{B}_a}$ be two natural transformations, the transpose $f^\perp : \varepsilon_Y \{ \}_a$ of $l$ lies above $fg$, then in fiber $\mathcal{B}_a$ there exists a unique morphism $v : \{ X \}_a \to f^* (Y) \in \text{Mor} \mathcal{B}_a$ over $g$ such that diagram commutes $f_Y v^\perp = l$. Henceforth, in fiber $\mathcal{T}_a$ we obtain an unique morphism $(T_a(v)) \eta_X : X \to T_a(f^*(Y)) \in \text{Mor} \mathcal{T}_a$ over $g$ such that $T_a(f_Y)(T_a(v)) \eta_X = l$, so $T_a(f_Y)$ is a Cartesian arrow of $f$ on the fibration $GP$, namely, the truth functor $T_a$ preserves Cartesian arrows. Similarly, we also can prove the comprehension functor $\{ - \}_a$ preserves opposite Cartesian arrows by dual principles. We omit this proof by length of this paper.

Above all, we prove $T_a \vdash \{ - \}_a$, $\eta$ and $\varepsilon$ is unit and co-unit of this adjunction, and $\eta$ is vertical morphism, that is, the restriction $P_a : \mathcal{T}_a \to \mathcal{B}_a$ of $P$ at $a$ is also a fibered fibration.

Fibration $G : \mathcal{B} \to \mathcal{C}$ depicts indexed types, and theorem 3 ensures if $P : \mathcal{T} \to \mathcal{B}$ is a fibered fibration over $G$. Then for $\forall a \in \text{Obj} \mathcal{C}$, the restriction $P_a : \mathcal{T}_a \to \mathcal{B}_a$ of $P$ at $a$ is also a fibered fibration with truth functor $T_a$ and comprehension functor $\{ - \}_a$, and $T_a \vdash \{ - \}_a$. The following is semantic model of many-sorted indexed fibration $P_a$.

**Definition 16.** Let $G : \mathcal{B} \to \mathcal{C}$ be a fibration in indexed category $\mathcal{C}$, $P : \mathcal{T} \to \mathcal{B}$ is a fibered fibration over $G$ with truth functor $T$ and comprehension functor $\{ - \}$. $\forall a \in \text{Obj} \mathcal{C}$ is an indexed object, then the restriction $P_a : \mathcal{T}_a \to \mathcal{B}_a$ of $P$ at $a$ is a many-sorted indexed fibration constructed by the pullback of $P$. $F_G^a$ is called to be a lifting of $F$ on $P_a$, which preserves truth such that $P_a F_G^a = F a$ and there exists two isomorphism expressions, namely, $T_a F \cong F_G^a T_a$ and $\{ - \}_a F_G^a \cong F(\{ - \}_a)$.

### 5.2. Semantic Properties of Many-Sorted Indexed Inductive Data Types

For $\forall D \in \text{Obj} \mathcal{B}_a$, we can construct a $F$-algebra $(D, \sigma : F(D) \to D)$ by the action of endofunctor $F$. The restriction $P_a$ of fibration $P$ at indexed object $a$ is a many-sorted indexed fibration, whose truth functor, that is, $T_a$ maps the $(D, \sigma)$ to a $F_G^a$-algebra $(T_a(D), T_a(\sigma) : T_a(F(D)) \cong F_G^a(T_a(D)))$. We let $\mu F$ be the carrier of initial $F$-algebra, since truth functor $T_a$ preserves initial objects $T_a(\mu F)$ is the carrier of $F_G^a$-algebra $(T_a(\mu F), in_G^a : T_a(F(\mu F)) \cong F_G^a(T_a(\mu F)) \to T_a(\mu F))$. Similarly, for subsection 3.2, we write $\text{Alg}(T_a)$ for the functor from $F$-algebra category $\text{Alg}_F$ to $F_G^a$-algebra category $\text{Alg}_{F_G^a}$, and define $\text{Alg}(T_a)^{\text{def}} = T_a$, we then have $\text{Alg}(T_a)(in) = in_G^a$, that is, $in_G^a$ is the isomorphism image of $in$ which is the morphism of initial $F$-algebra $(\mu F, in)$ by the action of functor $\text{Alg}(T_a)$.

For any $F_G^a$-algebra $(Z, \tau : F_G^a(Z) \to Z)$, the comprehension functor $\{ - \}_a$ of many-sorted indexed fibration $P_a$ maps the $F$-algebra $(Z, \tau)$
to a $F^I_G$-algebra $\{\{\}\}_a \{\tau\}_a : \{F^I_G(Z)_a \cong F\{Z\}_a \rightarrow \{Z\}_a\}$, as it can be seen from Figure 10. If $t : Z \rightarrow T_a(D)$ is a $F^I_G$-algebra morphism from $t$ to $T_a(\sigma)$, then the $F$-algebra morphism $s : \{\{\}\}_a \rightarrow D$ from $\{\tau\}_a$ to $\sigma$ is an $F$-algebra isomorphism over $t$. Similarly, $t$ is a $F^I_G$-algebra isomorphism over $s$. We define $\text{Alg}\{-\}_a \overset{\text{def}}{=} \{-\}_a$, and $\text{Alg}\{-\}_a$ is the functor from $\text{Alg}_{F^I_G} \rightarrow \text{Alg}_F$. $\text{Alg}\{-\}_a$ establishes a kind of inductive predicate rule of many-sorted indexed inductive data types and $\mu F$ as the carrier of initial $F$-algebra. That is, if the functor $\text{Alg}(T_a)$ preserves initial object, then the preserving truth lifting $F^I_G$ of $F$, with respect to $P_a$, generates a sound inductive rule.

\begin{equation}
\text{Alg}(T_a) \rightarrow \text{Alg}(\{\{\}\}_a),
\end{equation}

\begin{equation}
\text{Alg}(\{\{\}\}_a) \rightarrow \text{Alg}(\{\{\}\}_a).
\end{equation}

**Figure 10.** Adjunction properties of $\text{Alg}(T_a)$ and $\text{Alg}\{-\}_a$.

### 5.3. Inductive Rule of Many-Sorted Indexed Inductive Data Types

Let $P : T \rightarrow \mathcal{B}$ be a fibered fibration over the fibration $G : \mathcal{C} \rightarrow \mathcal{C}$, for $\forall a \in \text{Obj} \mathcal{C}$, $F : \mathcal{B}_a \rightarrow \mathcal{B}_a$ is an endo-functor in fiber $\mathcal{B}_a$, and $\mu F$ is the carrier of initial $F$-algebra. Each preserving truth lifting $F^I_G : T_a \rightarrow T_a$ of $F$ has an inductive rule in [16], so it ensures the validity of inductive rules generated on many-sorted indexed inductive data types by many-sorted indexed fibrations. Then we present and describe inductive rule of many-sorted indexed inductive data types with universality in the framework of Fibrations theory.

For $\forall D \in \text{Obj} \mathcal{B}_a$, $(D, m : F(D) \rightarrow D)$ is a $F$-algebra in fiber $\mathcal{B}_a$. A recursive operation $\text{fold} : (F(D) \rightarrow D) \rightarrow \mu F \rightarrow D$ is constructed by fibration $P_a$ in base category $\mathcal{B}_a$ on many-sorted indexed inductive data types, and $\text{fold} m$ maps $m$ to a unique morphism $\text{fold} m : \mu F \rightarrow D$ from the initial $F$-algebra $(\mu F, in : F(\mu F) \rightarrow \mu F)$ to $m$. By the property that truth functor $T_a$ preserves initial objects, $T_a(\mu F)$ is the carrier of initial $F^I_G$-algebra. We write $\mu F^I_G = T_a(\mu F)$, and the two isomorphism expressions are as follows:

$T_a(F(D)) \cong F^I_G(T_a(D))$, $T_a(F(\mu F)) \cong F^I_G(T_a(\mu F)) = F^I_G(\mu F^I_G)$. A recursive operation $\text{fold} : (F^I_G(Z) \rightarrow Z) \rightarrow \mu F^I_G \rightarrow Z$ is constructed by $F^I_G$ in total category $T_a$ on fibration $P_a$, for any $F^I_G$-algebra $(Z, n : F^I_G(Z) \rightarrow Z)$, $\text{fold} n$ maps $n$ to unique $F^I_G$-algebra morphism $\text{fold} n : \mu F^I_G \rightarrow Z$ from initial $F^I_G$-algebra $(\mu F^I_G, in^I_G : F^I_G(\mu F^I_G) \rightarrow F^I_G(\mu F^I_G))$ to $n$. For $\forall D \in \text{Obj} \mathcal{B}_a$, $\forall Z \in \text{Obj} \mathcal{B}_a$, an inductive rule of many-sorted indexed inductive data types with universality is as follows:

$\text{Ind}^I_{\text{Gen}} : (F^I_G(Z) \rightarrow Z) \rightarrow T_a(\mu F) \rightarrow Z$.

If $(Z, n : F^I_G(Z) \rightarrow Z)$ is a $F^I_G$-algebra over the $F$-algebra $(D, m : F(D) \rightarrow D)$, then $\text{Ind}^I_{\text{Gen}} Zm$ is a $F^I_G$-algebra homomorphism over $\text{fold} n$.

### 5.4. Example Analysis of Many-Sorted Indexed Inductive Data Types

**Example 3.** Even number $\text{EVEN}$ and odd number $\text{ODD}$ are mutually recursive many-sorted indexed inductive data types, we let $a, b$ be only two indexed objects and $\text{succ}_a, \text{pre}_a : a \rightarrow b$, $\text{succ}_b, \text{pre}_b : b \rightarrow a$ be two pairs of morphisms in indexed category $\mathcal{C}$, $a$ is indexed object of $\text{EVEN}$, and $b$ is indexed object of $\text{ODD}$. We define a functor $F : \mathcal{B} \times \mathcal{B} \rightarrow \mathcal{B} \times \mathcal{B}$ on binary production $\mathcal{B} \times \mathcal{B}$ in base category, for $\forall E \in \text{EVEN}, \forall O \in \text{ODD}, F(E, O) = (O, E)$. For any pair of properties $(Z, Z')$ of even number type and odd number type in total category $(T_a, T_b)$ with respect to many-sorted indexed fibrations $(P_a, P_b)$, such as $Z \in \text{Obj} T_a$, $Z' \in \text{Obj} T_b$, the former is exactly divisible by 2 and the latter is indivisible by 2, then there exists an induction principle for $\text{EVEN}$ and $\text{ODD}:

\begin{align*}
(Z'\langle 0 \rangle) & \rightarrow ((Z(E) \rightarrow (Z'(\text{succ}_E(E)))) \\
\land Z'(\text{pre}_E(E)) & \land (Z'(O) \rightarrow (Z(\text{succ}_b(O)))) \\
\land Z(\text{pre}_a(O))) & \rightarrow (Z'(E) \times Z(O)).
\end{align*}
Let \((EVEN, ODD)\) be carrier \(\mu F = (\mu F_E, \mu F_O)\) of initial \(F\)-algebra over binary production in base category \((\mathcal{B}_a, \mathcal{B}_b)\) with respect to many-sorted indexed fibration \((P_a, P_b)\), for any \(F\)-algebra \(m : F(E, O) \to (E, O)\), it is lifted to be a \(F^G_{\mu^G}\)-algebra \(n : F^G_{\mu^G}(Z, Z') \to (Z, Z')\) by many-sorted indexed fibration \((P_a, P_b)\), which satisfies diagram commutes \(F((P_a, P_b))(Z, Z') = (\mu F_{\mu^G})(Z, Z')\). A recursive operation \textit{fold} \(m\) is defined by initial properties of initial \(F\)-algebra on \((EVEN, ODD)\), it executes judgment of data type \((EVEN, ODD)\); another recursive operation defined by initial properties of initial \(F^G_{\mu^G}\)-algebra depicts semantics of \((EVEN, ODD)\). If \(n\) lies above \(m\), then \(\text{Ind}_{\mu^G}(Z, Z') n\) is a \(F^G_{\mu^G}\)-algebra homomorphism over \textit{fold} \(m\), when iterating each objects in total category \((\mathcal{T}_a, \mathcal{T}_b)\) on many-sorted indexed fibration \((P_a, P_b)\), a semantic set describing properties of \((EVEN, ODD)\) is as follows:

\[
\{(Z(E), Z'(O)) | \forall E \in EVEN, \forall O \in ODD\}.
\]

Mutual recursive is a complex many-sorted inductive data type. Therefore, traditional methods, including algebras and category theory, are difficult to process effectively its semantic computing. Example 3 makes a semantic model of many-sorted indexed fibration by Fibrations theory, which is not dependent on some particular tools such as predicates logics and set theory strictly, which analyzes deeply semantic properties of mutual recursive, and depicts abstractly inductive rule with universality. The work of example 3 expands and deepens traditional methods in the level of category theory, deals with semantic computation of mutual recursive in the uniform Fibrations theory framework, and further develops the width and depth of traditional methods of inductive data types in math.

6. Related Works

Inductive data types is an import branch of types theory. Martin-Löf constructive types theory [17] achieved a series of significant research results in 1970s, which laid the foundations for constructive inductive data types. However, there exist some deficiencies in polymorphism type system, e.g., inability to give sound interpretation in classical set theory models [18], and inconsistent problems of classical reasoning in constructive logics [19]. Joint efforts of many scholars promote the development of inductive data types further, such as Pitt’s model in [19], effective topos in [20], Mod model in [21], and so on, which resolves the problems above to some extent. Therewith, generalized inductive types have become a key point of inductive data types, and it is introduced to researching of construction calculus [22-23]. Literature [24] presented categorical model \textit{Per} of generalized inductive types, and gave its interpretations of construction calculus in the subcategory \(\omega - \text{Set}\) of effective topos [25].

The existing research mainly focus on finite syntax construction of inductive data types by methods of algebras or category theory, for instance, [26] analyzed constructions of inductive data types based on algebraic functors, described semantic relationships and properties in uniform formal frameworks. However, for numerous inductive data types, including streams, lists, trees, stacks and heaps, there still exist plenty of unsolved problems in semantic computing and program logic, for example, analyzing semantic properties and depicting inductive rules, especially the latter, are mostly generated automatically. For example, construction calculus generates its inductive rules automatically only from inductive structures of inductive data types, which hardly satisfies logic modeling requirements of semantic computing at the level of functor syntax, and not the level of functor semantics, so their inductive rules are weakened to some appended axioms. In terms of inductive rules generated automatically, we take an example of Church Encoding in construction calculus, which is proved to be invalid [27].

Inductive rules generated automatically are scarce of solid math foundation and accurate formal description. Considering the status quo of the three usual inductive data types including fibered, single-sorted indexed and many-sorted indexed inductive data types, this paper presents semantic models of non-indexed, single-sorted indexed and many-sorted indexed fibration by math tools of Fibrations theory, analyzes their semantic properties deeply, and depicts abstractly their inductive rules with universality. Mathes had gained more prominent achievements in the field of inductive data
types [28], he researched systematically inductive rules of nested data types in intentional types theory, but his functor only manages to depict inductive rules within the second-order limitation. Nevertheless, the single-sorted indexed fibration $P/I$ presented by this paper depicts well inductive rule of Stream$(I)$ which is first-order inductive data types, and the many-sorted indexed fibration $(P_a, P_b)$ depicts well inductive rule of EVEN and ODD which are second-order inductive data types. Therefore, by combining with single-sorted indexed fibration and many-sorted indexed fibration, we can further describe inductive rules of any high-order inductive data types. At the same time, Matthes processed his recursive computation of inductive data types based on unassertive axiom methods, which was dependent on predicates logic strictly, whose inductive rules are not universal. However, this paper depicts inductive rules of inductive data types by initial algebras semantics, which further enhances the abilities of analyzing semantic properties and describing inductive rules of complex indexed fibration by the tool of Fibrations theory which is a highly abstract and flexible extensive.

Inductive data types in programming based on Fibrations theory present the expansion and deepening of traditional inductive data types methods at the level of category theory, especially after coalgebraic methods in [2] appear some categorical concepts including Cartesian arrow and opposite Cartesian arrow, fibration, and after opfibration has been integrated organically, which renders Fibrations theory to revitalize in the field of inductive data types in programming, and possesses promising application in the proceeding of theory researching, as well as in engineering practice in computer science. Meanwhile, inductive data types based on Fibrations theory do not represent research of pure math, but from the applicative standpoint of program languages, combining Fibrations theory with the latest research findings of object-oriented languages, algebraic specifications and semantic computation, it carries out fundamental research for some kernel problems, e.g., categorical properties and semantic interpretations of core concepts of inductive data types in programming, semantic computation, and specifications description of operation behaviors.

Compared with traditional methods including algebras and category theory, the advantages of this paper in analyzing semantic properties and describing inductive rules of inductive data types are mainly reflected in the following three aspects:

Firstly, succinct description and flexible extension of Fibrations theory can accurately analyze semantics of inductive data types, compute semantics automatically, reduce coupling factors between inductive data types in programming, and strengthen cohesion of inductive data types, thus improving dependence of program languages;

Secondly, a high abstract of Fibrations theory does not rely on particular constraints of traditional methods, gives semantic interpretations of inductive data types in the level of fibration semantics, and depicts abstractly inductive rules of inductive data types with universality, which provides solid math foundations and accurate formal descriptions for semantic computation of program languages modeling;

Thirdly, rigorousness and consistency of Fibrations theory in semantics computing are suitable for reasoning precisely, for programming, especially analyzing and constructing of inductive data types on preliminary stages in programming, which reduces extreme errors in early software developing, and provides a sound basis for post-works, e.g., confirmation testing and system maintenance.

7. Conclusion

Fibrations theory integrates traditional thinking of program languages, whose high abstraction, flexible extension and succinct description produces vigorous and profound impact on programming and its formal semantics, and promotes extreme applications of category theory in computer science. However, from the status of document retrieval performed nowadays, there are a few scholars carrying out research of Fibrations theory internationally, and relatively small amount of literature applies Fibrations theory to computer science. Notably, the literature systematically researching and deeply aiming at programming and its formal semantics are of even less quantity. Regarding China, we have not found other scholars carrying out
research Fibrations theory and their applications in computer science.

This paper researches syntax construction and semantics of inductive data types in programming by Fibrations theory to make semantic models of fibration, and depicts their inductive rules with universality by ad joint functors and their adjunction properties. This provides succinct and uniform descriptive ways for semantics computation and program logic in programming. At the same time, it also enhances processing and proving abilities of program languages for semantic behaviors of inductive data types. Fibrations theory has particular advantages over resolving abstract problems description. Meanwhile, it also has promising applications in theoretical computer science. We hope our work will stimulate other scholars in China to pay closer attention to Fibrations theory.

Our future work will discuss preliminarily soundness, completeness, and consistence of formal system involving inductive data types and their inductive rules. Furthermore, we will extend our works to co-inductive data types and their co-inductive rules, as well as analyze deeply integration and computation between inductive data types and co-inductive data types by dual principles of category theory and appropriate distributive laws. Moreover, expanding our work on inductive data types and co-inductive data types to 2-category by Fibrations theory, profoundly discussing math structures and categorical properties of semantic computing and program logic in high-order category still lies ahead of us.

Acknowledgement

This work is funded by Guangdong Provincial Natural Science Foundation, China (Grant No. S2013010015944), Guangdong Provincial University Outstanding Young Teacher Programming, China (Grant No. YQ2014155), Guangdong Provincial strategic emerging industries core technology research projects, China (Grant No. 2011A010801008, 2012A010701011, 2012A010701003), Guangzhou municipal science and technology plan, China (Grant No. 20120000034) and Shaoquan municipal science and technology plan, China (Grant No. 2013CX/K 61).

References

http://dx.doi.org/10.2168/lmcs-9(3:6)2013


http://dx.doi.org/10.1007/3-540-13346-17


http://dx.doi.org/10.1007/3-540-52335-9 47


http://dx.doi.org/10.1109/LICS.1997.614955

http://dx.doi.org/10.1007/978-3-642-15205-4 27

http://dx.doi.org/10.1017/S095679680900731X

Received: May, 2015
Revised: September, 2015
Accepted: September, 2015

Contact addresses:

Decheng Miao
School of Information Science and Engineering
Shaoguan University
288 Daxue Road
Shaoguan
China
e-mail: miaodecheng@squ.edu.cn

Jianqing Xi
School of Software
South China University of Technology
381 Wushan Road
Guangzhou
China
e-mail: xijianqingxj@163.com

Yubin Guo
School of Information
South China University of Agriculture
483 Wushan Road
Guangzhou
China
e-mail: lxmJy@126.com

Deyou Tang
School of Software
South China University of Technology
381 Wushan Road
Guangzhou
China
e-mail: tangdy@163.com

Yubin Guo is an associate professor at South China University of Agriculture, Guangzhou, China. He received his PhD degree in computer application technology from the School of Computer Science and Engineering, South China University of Technology in 2012. His research interests include formal languages theory, categorical methods and database system. He has more than 20 research papers presented in journals and at conferences.

Jianqing Xi is a professor and a doctoral supervisor at South China University of Technology, Guangzhou, China. He received his PhD degree in computer architecture from the department of Computer Science, National University of Defense Technology in 1992. His research interests include database and network computing. He has more than 100 research papers presented in journals and at conferences.

Decheng Miao is an associate professor at Shaoguan University, Shaoguan, China. He received his PhD degree in computer application technology from the School of Computer Science and Engineering, South China University of Technology in 2012. His research interests include formal languages theory, categorical methods and database system. He has more than 20 research papers presented in journals and at conferences.

Deyou Tang is an associate professor at South China University of Technology, Guangzhou, China. He received his PhD degree in computer application technology from the School of Computer Science and Engineering, South China University of Technology in 2008. His research interests include algorithm design, database and network computing. He has more than 60 research papers presented in journals and at conferences.