

Inductive Data Types Based on Fibrations Theory in Programming

Decheng Miao¹, Jianqing Xi², Yubin Guo³ and Deyou Tang²

¹School of Information Science and Engineering, Shaoguan University, Shaoguan, China

²School of Software, South China University of Technology, Guangzhou, China

³School 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 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 transforma-

tion, 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 $\mathbf{Obj} \mathcal{C}$ be a set of objects for category \mathcal{C} , and $\mathbf{Mor} \mathcal{C}$ a set of morphisms for category \mathcal{C} . We introduce some basic concepts in this paper.

Definition 1. Let $P : \mathcal{T} \rightarrow \mathcal{B}$ be a functor between small categories \mathcal{T} and \mathcal{B} , $f : C \rightarrow D \in \mathbf{Mor} \mathcal{B}$, $u : X \rightarrow Y \in \mathbf{Mor} \mathcal{T}$. We call morphism u a Cartesian arrow of f and Y if $P(Y) = D$, $P(u) = f$, for $\forall Z \in \mathbf{Obj} \mathcal{T}$, $v : Z \rightarrow Y \in \mathbf{Mor} \mathcal{T}$ and $\forall h : P(Z) \rightarrow C \in \mathbf{Mor} \mathcal{B}$, it satisfies $f \circ h = P(v)$, and there exists a unique $w : Z \rightarrow X \in \mathbf{Mor} \mathcal{T}$ 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 $P : \mathcal{T} \rightarrow \mathcal{B}$ be a functor between small categories \mathcal{T} and \mathcal{B} ; then if for $\forall Y \in \mathbf{Obj} \mathcal{T}$ and $\forall f : C \rightarrow P(Y) \in \mathbf{Mor} \mathcal{B}$ 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 $P : \mathcal{T} \rightarrow \mathcal{B}$, we call \mathcal{B} the base category of P , and \mathcal{T} its total category. If for an object C in $\mathbf{Obj} \mathcal{B}$, $\exists X \in \mathbf{Obj} \mathcal{T}$, $k \in \mathbf{Mor} \mathcal{T}$ is satisfied $P(X) = C$ and $P(k) = id_C$, 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_Y^\downarrow for Cartesian arrow u of f and Y in definition 1 if $f^*(Y)$ is domain of f_Y^\downarrow , then $f^*(Y)$ lies above C , that is, $Y \in \mathbf{Obj} \mathcal{T}_D, f^*(Y) \in \mathbf{Obj} \mathcal{T}_C$.

Definition 3. A morphism $f : C \rightarrow D$ in base category \mathbb{B} is extended to be a functor $f^* : \mathbb{T}_D \rightarrow \mathbb{T}_C$ between fibers \mathbb{T}_D and \mathbb{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 $P : \mathbb{T} \rightarrow \mathbb{B}$ be a functor between small categories \mathbb{T} and \mathbb{B} , $f : C \rightarrow D \in \mathbf{Mor} \mathbb{B}$, $u : X \rightarrow Y \in \mathbf{Mor} \mathbb{T}$. We call morphism u an opposite Cartesian arrow of f and X if $P(X) = C$, $P(u) = f$, for $\forall Z \in \mathbf{Obj} \mathbb{T}$, $v : X \rightarrow Z \in \mathbf{Mor} \mathbb{T}$ and $\forall h : D \rightarrow P(Z) \in \mathbf{Mor} \mathbb{B}$, it satisfies $h \circ f = P(v)$, and there exists a unique $w : Y \rightarrow Z \in \mathbf{Mor} \mathbb{T}$ such that $w \circ u = v$ and $P(w) = h$.

Similarly to definition 1, if u is a co-cone [10] in category \mathbb{T} , the opposite Cartesian arrow u in definition 4 is a universal co-cone in \mathbb{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 $P : \mathbb{T} \rightarrow \mathbb{B}$ be a functor between small categories \mathbb{T} and \mathbb{B} ; if for $\forall X \in \mathbf{Obj} \mathbb{T}$ and $\forall f : P(X) \rightarrow D \in \mathbf{Mor} \mathbb{B}$, there exists an opposite Cartesian arrow of f and X , we call P an opfibration.

Definition 6. If the functor $P : \mathbb{T} \rightarrow \mathbb{B}$ between small categories \mathbb{T} and \mathbb{B} is simultaneously a fibration and an opfibration, then it is a bifibration.

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

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

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

Definition 8. If $F \dashv G : \mathbb{C} \rightarrow \mathbb{D}$ is a pair of ad joint functors, η, ε is the unit and co-unit of this adjunction respectively, and for $\forall X \in \mathbf{Obj} \mathbb{C}$, $\forall Y \in \mathbf{Obj} \mathbb{D}$, $\exists f : F(X) \rightarrow Y \in \mathbf{Mor} \mathbb{D}$, $\exists g : X \rightarrow G(Y) \in \mathbf{Mor} \mathbb{C}$, then the transpose of f and g is $G(f)\eta_X$ and $\varepsilon_Y F(g)$ respectively.

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

Proof. \Rightarrow . Let $*f \dashv f^* : \mathbb{T}_C \rightarrow \mathbb{T}_D$ be a pair of ad joint functors, the unit is η , the co-unit is ε , and $P : \mathbb{T} \rightarrow \mathbb{B}$ is a fibration between small categories \mathbb{T} and \mathbb{B} . For $\exists Y \in \mathbf{Obj} \mathbb{T}_D$, we can construct a Cartesian arrow $f_Y^{\downarrow} : f^*(Y) \rightarrow Y$ whose codomain is Y . $\exists X \in \mathbf{Obj} \mathbb{T}_C$, let $l : X \rightarrow *f(X)$ be a morphism above f , the proof that l is an opposite Cartesian arrow above f is as follows: it satisfies $l = f_{*f(X)}^{\downarrow} \circ \eta_X$ by the adjunction properties of $*f \dashv f^*$, seen from Figure 1. We write id to identify morphism, if $g : X \rightarrow Y$ is another morphism above f , let $\phi : X \rightarrow f^*(Y)$ be vertical morphism in \mathbb{T}_C , we get $P(\phi) = id_C$. By definition 1 we know $g = f_Y^{\downarrow} \circ \phi$, Cartesian arrow f_Y^{\downarrow} is an universal cone, whose universal property ensures ϕ is the unique morphism from g to f_Y^{\downarrow} . We write $\hat{\phi}$ for the transpose of ϕ under the adjunction $*f \dashv f^*$, then $\hat{\phi} = \varepsilon_Y \circ *f(\phi) : *f(X) \rightarrow Y$, $f^*(\hat{\phi}) \circ \eta_X = \phi$. Universal property of universal cone f_Y^{\downarrow} ensures the unique existence of $f^*(\hat{\phi})$, and it satisfies $\hat{\phi} \circ f_{*f(X)}^{\downarrow} = f_Y^{\downarrow} \circ f^*(\hat{\phi})$. Above all, there exist two equalities, that are, $\hat{\phi} \circ l = \hat{\phi} \circ f_{*f(X)}^{\downarrow} \circ \eta_X = f_Y^{\downarrow} \circ f^*(\hat{\phi}) \circ \eta_X = f_Y^{\downarrow} \circ \phi = g$, $g = \hat{\phi} \circ l$, then the transpose of ϕ is the unique morphism from l to g , and $P(\hat{\phi}) = id_D$. We thus prove that l is an opposite Cartesian arrow f_{\downarrow}^X above f by definition 4.

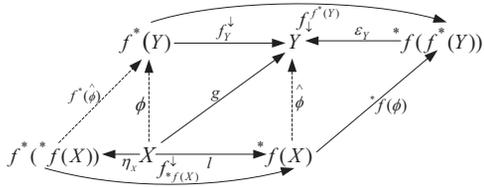


Figure 1. Proof of opposite Cartesian arrow.

\Leftarrow . We assume $g : X \rightarrow Y \in \mathbf{Mor} \mathbb{T}$ lies above f , write $\mathbb{T}_C(X, f^*(Y))$ for set composed of morphisms above C in fiber \mathbb{T}_C , $\mathbb{T}_D(*f(X), Y)$ for set composed of morphisms above D in fiber \mathbb{T}_D . For $\forall k : X' \rightarrow X \in \mathbf{Mor} \mathbb{T}_C$, $\forall h : Y \rightarrow Y' \in \mathbf{Mor} \mathbb{T}_D$, because $P : \mathbb{T} \rightarrow \mathbb{B}$ is a bifibration, this gives us one-to-one corresponding map $\varphi_{X,Y} : \mathbb{T}_D(*f(X), Y) \rightarrow \mathbb{T}_C(X, f^*(Y))$. We write $k^{op} : X \rightarrow X' \in \mathbf{Mor} \mathbb{T}_C$ for an opposite morphism of k , it satisfies that $k^{op} \circ f_{\downarrow}^{Xop} = f_{\downarrow}^{X'op} \circ *f(k^{op})$ and $id_{f^*(Y)} \circ f_Y^{\downarrow op} = f_Y^{\downarrow op} \circ id_Y$, so the left part of diagram in Figure 2 commutes. Similarly, it also satisfies $id_X \circ f_{\downarrow}^{Xop} = f_{\downarrow}^{Xop} \circ id_{*f(X)}$ and $f^*(h) \circ f_Y^{\downarrow op} = f_{Y'}^{\downarrow op}$, i.e., the right part of diagram in Figure 2 commutes, so $\varphi_{X,Y}$ is natural isomorphism. We thus prove $*f \dashv f^*$ by definition of adjoint functors in [13].

$$\begin{array}{ccc}
 \mathbb{T}_c(X, f^*(Y)) & \xleftarrow{\varphi_{X,Y}} & \mathbb{T}_b(*f(X), Y) & \mathbb{T}_c(X, f^*(Y)) & \xleftarrow{\varphi_{X,Y}} & \mathbb{T}_b(*f(X), Y) \\
 \mathbb{T}_c(k^{op}, id_{f^*(Y)}) \downarrow & & \mathbb{T}_b(*f(k^{op}), id_Y) \downarrow & \mathbb{T}_c(id_X, f^*(h)) \downarrow & & \mathbb{T}_b(id_{f(X)}, h) \downarrow \\
 \mathbb{T}_c(X', f^*(Y)) & \xleftarrow{\varphi_{X',Y}} & \mathbb{T}_b(*f(X'), Y) & \mathbb{T}_c(X, f^*(Y)) & \xleftarrow{\varphi_{X,Y}} & \mathbb{T}_b(*f(X), Y)
 \end{array}$$

Figure 2. Proof of adjunction properties.

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^* and op-re-indexing functor $*f$ 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 fibration by Fibrations theory, which analyzes se-

mantic properties and describes inductive rule of fibered inductive data types.

3.1. Semantic Model of Non-Indexed Fibration

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

Definition 10. Let $F : \mathbb{T} \rightarrow \mathbb{T}'$ and $G : \mathbb{T}' \rightarrow \mathbb{T}$ be two fibered functors above base category \mathbb{B} , we call G a right fibered adjoint functor to F , and $F \dashv G$ is a pair of fibred adjunction above \mathbb{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 $P : \mathbb{T} \rightarrow \mathbb{B}$ be a functor between small categories \mathbb{T} and \mathbb{B} , $F : \mathbb{B} \rightarrow \mathbb{B}$ is an endo-functor in base category \mathbb{B} , the lifting of F with respect to P is an endo-functor $F^\perp : \mathbb{T} \rightarrow \mathbb{T}$ in total category \mathbb{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 $\forall D \in \mathbf{Obj} \mathbb{B}$, if $\exists 1_D \in \mathbf{Obj} \mathbb{T}_D$, which is the terminal object of fiber \mathbb{T}_D , and $\forall f : C \rightarrow D \in \mathbf{Mor} \mathbb{B}$, $f^*(1_D)$ is then the terminal object of fiber \mathbb{T}_C , namely, re-indexing functor f^* 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 $P : \mathbb{T} \rightarrow \mathbb{B}$ be a non-indexed fibration, functor $T : \mathbb{B} \rightarrow \mathbb{T}$ maps $\forall C \in \mathbf{Obj} \mathbb{B}$ to the terminal object in fiber \mathbb{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 $\mathbf{1}_{\mathbb{B}}$ and $\mathbf{1}_{\mathbb{T}}$ are terminal objects of base category \mathbb{B} and total category \mathbb{T} respectively, we thus get $P(\mathbf{1}_{\mathbb{T}}) = \mathbf{1}_{\mathbb{B}}$. For $\forall C \in \mathbf{Obj} \mathbb{B}$ there exists a unique morphism $u : C \rightarrow \mathbf{1}_{\mathbb{B}}$ in base category \mathbb{B} , so we have a isomorphism expression, that is, $T(C) \cong u^*(\mathbf{1}_{\mathbb{T}})$. For $\forall f : C \rightarrow D \in \mathbf{Mor} \mathbb{B}$, we have an isomorphism expression $f^*(T(D)) \cong T(C)$, and the truth functor T maps f to a Cartesian arrow $f_{T(D)}^\perp$ in total category \mathbb{T} . If the truth functor T is full and faithful [13], then it is a fibered right adjoint functor to non-indexed fibration P [8].

For each object C in category \mathbb{B} , a F -algebra $(C, i : F(C) \rightarrow C)$ is constructed by the endofunctor F , we call C the carrier of this F -algebra. The morphism between (C, i) and another F -algebra $(D, j : F(D) \rightarrow D)$ is a morphism $f : C \rightarrow D$ 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 Alg_F for it. If the initial F -algebra $(\mu F, in : F(\mu F) \rightarrow \mu F)$ 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 μF , as the carrier of initial F -algebra, is least fixed point of functor F . The functor F denotes syntax construction of μF , and its morphism in gives a kind of semantic interception of μF under this syntax construction.

A F -algebra $(C, i : F(C) \rightarrow C)$ is mapped to a F^\perp -algebra $(T(C), T(i) : T(F(C)) \cong F^\perp(T(C)) \rightarrow T(C))$ 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 $Alg(T)$ for the functor from F -algebra category Alg_F to F^\perp -algebra category Alg_{F^\perp} , and define $Alg(T) \stackrel{def}{=} T$. Objects and morphisms in base category \mathbb{B} with respect to non-indexed

fibration P are mapped to those responded in total category \mathbb{T} by the truth functor T , which further makes connections from Alg_F to Alg_{F^\perp} via $Alg(T)$ and isomorphism property of composed functors of definition 12. The initial F^\perp -algebra $(T(\mu F), in^\perp : F^\perp(T(\mu F)) \rightarrow T(\mu F))$ in total category \mathbb{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 $Alg(T)$, that is, $Alg(T)(in) = in^\perp$. 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 $P : \mathbb{T} \rightarrow \mathbb{B}$ be a non-indexed fibration between small categories \mathbb{T} and \mathbb{B} , $T : \mathbb{B} \rightarrow \mathbb{T}$ is a truth functor of P . If $\{-\}$ is a right adjoint functor to T , namely, $T \dashv \{-\}$, then $\{-\}$ is called a comprehension functor of P .

Let $\sigma : \{-\} \rightarrow P$ be a natural transformation, $F\sigma$ is also a natural transformation by composed theorem of natural transformations, as stated in [13], then for $\forall X \in \mathbf{Obj} \mathbb{T}$, $*(F\sigma_X) : \mathbb{T}_{F\{X\}} \rightarrow \mathbb{T}_{FP(X)}$ is an op-re-indexing functor induced by $F\sigma_X$, and $F^\perp(X) = *(F\sigma_X)(T(F\{X\})) \in \mathbf{Obj} \mathbb{T}_{FP(X)}$, i.e., the action $F^\perp(X)$ of each object X in total category by the lifting F^\perp of endofunctor F in base category, is determined completely by its semantic behaviors of F above $\{X\}$, and $\{X\}$ is the extension of X over the comprehension functor $\{-\}$. For $\forall k : X \rightarrow X' \in \mathbf{Mor} \mathbb{T}$, $F^\perp(k) = F^\perp(X) \rightarrow F^\perp(X') = *(F\sigma_X)(T(F\{X\})) \rightarrow *(F\sigma_{X'})(T(F\{X'\}))$, that is, $F^\perp(k) \in *(FP(k))$, then $F^\perp(k)$ is an element of re-indexing functor morphism $*(FP(k))$.

Similar to $Alg(T)$, we write $Alg\{-\}$ for the functor from F^\perp -algebra category Alg_{F^\perp} to F -algebra category Alg_F , define $Alg\{-\} \stackrel{def}{=} \{-\}$, it satisfies $Alg(T) \dashv Alg\{-\}$ in [9] by adjunction property of $T \dashv \{-\}$ from definition 13. For each F^\perp -algebra $(X, j : F^\perp(X) \rightarrow X)$, $Alg\{-\}(j) = F\{X\} \rightarrow \{X\}$, that is, $Alg\{-\}(j) = \{j\}$, seen from Figure 3. By previous analysis we also know that $\{j\}$ is an isomorphism image of j by the action of functor $Alg\{-\}$. If $g : X \rightarrow T(C)$ is a F^\perp -algebra morphism

from j to $Alg(T)(i)$, then a F -algebra morphism $h : \{X\} \rightarrow C$ from $Alg\{-\}(j)$ to i is a F -algebras homomorphism above g . Accordingly, g is a F^\perp -algebras homomorphism above h . The right ad joint functor $Alg\{-\}$ to $Alg(T)$ 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 μF .

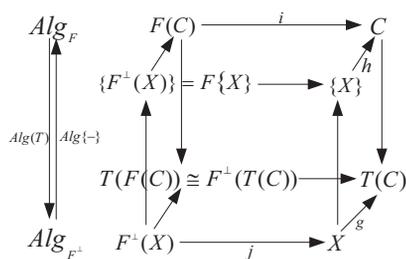


Figure 3. Adjunction properties of $Alg(T) \vdash Alg\{-\}$.

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 $P : \mathbb{T} \rightarrow \mathbb{B}$, its comprehension functor $\{-\}$ is right ad joint to its truth functor T , i.e., $P \dashv T \dashv \{-\}$. Let F be an endo-functor in base category \mathbb{B} , and μF is the carrier of initial F -algebra, then each preserving truth lifting F^\perp of F with respect to P has a 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 μF be the carrier of initial F -algebra, we utilize endo-functor F in base category \mathbb{B} to construct a recursive operation on fibered inductive data types $fold : (F(C) \rightarrow C) \rightarrow \mu F \rightarrow C$. For each F -algebra $(C, i : F(C) \rightarrow C)$, by the action of recursive operation $fold$, $fold i$ is the map sending i to the unique F -algebra morphism $fold i : \mu F \rightarrow C$ from initial F -algebra morphism in to i , seen from Figure 4. The essence of $fold$ stems from initial algebra semantics is a parameterized recursive operation on inductive data types, which possesses lots of good properties such as correct semantics, flexible extension and succinct expression.

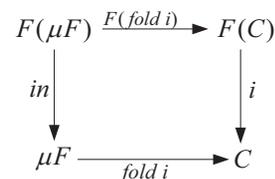


Figure 4. F -algebra morphism.

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 $\mu F^\perp = T(\mu F)$, $X = T(C) \in \mathbf{Obj} \mathbb{T}$. Similarly, we can construct a recursive operation $fold : (F^\perp(X) \rightarrow X) \rightarrow \mu F^\perp \rightarrow X$, which depicts semantics of fibered inductive data types in total category \mathbb{T} by the tool of preserving truth lifting F^\perp of F , see Figure 5. So for $\forall C \in \mathbf{Obj} \mathbb{B}$, $X \in \mathbf{Obj} \mathbb{T}_C$, we have the inductive rule of fibered inductive data types with universality, that is,

$$Ind_{Gen} : (F^\perp(X) \rightarrow X) \rightarrow T(\mu F) \rightarrow X.$$

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

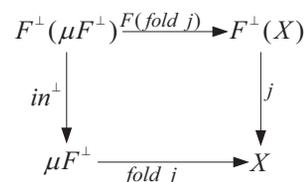


Figure 5. F^\perp -algebra morphism.

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 \mathbb{B} , we write $\mathbf{1}$ for the terminal object. For $\forall N \in \mathbf{Obj} \mathbb{B}$, it is to be said $F : N \rightarrow \mathbf{1} + N$, then its inject $in_1(\mathbf{1}) = 0$ is the minimum natural number, its inject $in_2(N) = N + 1$ is a successful function. For any natural number property $X \in \mathbf{Obj} \mathbb{T}_N$ in total category \mathbb{T} with respect to the non-indexed fibration P , e.g., transitivity, compatibility and completeness, an induction $(X(in_1(\mathbf{1})) \rightarrow (X(N) \rightarrow X(in_2(N)))) \rightarrow X(N)$ holds for property X of Nat . For each F -algebra $(N, h : F(N) \rightarrow N)$, it is lifted to be a F^\perp -algebra $(X, k : F^\perp(X) \rightarrow X)$ via the non-indexed fibration P , which satisfies diagram commutes, that is, $FP(X) = PF^\perp(X)$. 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^\perp -algebra describes semantics of Nat . If k lies over h , then $Ind_{Gen} X k$ is a F^\perp -algebra homomorphism over $fold h$, and when it iterates each object in total category \mathbb{T} , we can get the semantics set describing properties of Nat , that is $\{X(N) | \forall N \in \mathbf{Obj} \mathbb{B}\}$.

Compared with traditional methods including algebras and category theory, inductive rule Ind_{Gen} 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 Ind_{Gen} 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 $P : \mathbb{T} \rightarrow \mathbb{B}$ be a fibration or bifibration between small categories \mathbb{T} and \mathbb{B} , $T : \mathbb{B} \rightarrow \mathbb{T}$ is the truth functor of P . $\exists I \in \mathbf{Obj} \mathbb{B}$ is a discrete indexed object in base category \mathbb{B} , let single-sorted indexing functor $P/I : \mathbb{T}/T(I) \rightarrow \mathbb{B}/I$ be $P/I(u) = P(u) : P(Y) \rightarrow I \in \mathbf{Obj} \mathbb{B}/I$ for $\forall u : Y \rightarrow T(I) \in \mathbf{Obj} \mathbb{T}/T(I)$, then single-sorted indexing functor P/I is also a fibration or bifibration.

Proofs. For $\forall f : C \rightarrow D \in \mathbf{Mor} \mathbb{B}$, there exists a Cartesian arrow $f_X^\perp : f^*(X) \rightarrow X$ above f with respect to fibration P such that $P(X) = D$, and exists an unique morphism $w : T_P(I) \rightarrow f^*(X)$ such that $v = f_X^\perp \circ w$ and $P(v) = f \circ h$, seen from Figure 6. Let $\alpha : D \rightarrow I \in \mathbf{Obj} \mathbb{B}/I, \beta : C \rightarrow I \in \mathbf{Obj} \mathbb{B}/I$, we thus have two morphisms in the slice category \mathbb{B}/I , i.e., $\gamma : P(u) \rightarrow \alpha = P(Y) \rightarrow D \in \mathbf{Mor} \mathbb{B}/I$ and $\delta : P(u) \rightarrow \beta = P(Y) \rightarrow C \in \mathbf{Mor} \mathbb{B}/I$, which satisfy diagram commutes, that is $\gamma = f \circ \delta$. In total category $\mathbb{T}/T(I)$ of functor $P/I, s : X \rightarrow T(I) \in \mathbf{Obj} \mathbb{T}/T(I), t : f^*(X) \rightarrow T(I) \in \mathbf{Obj} \mathbb{T}/T(I)$, so $g : u \rightarrow s = Y \rightarrow X \in \mathbf{Mor} \mathbb{T}/T(I)$, there exists a unique morphism $k : u \rightarrow t = Y \rightarrow f^*(X)$ such that $g = f_X^\perp \circ k$, then f_X^\perp is a Cartesian arrow of f with respect to functor P/I by definition 1, namely, if P is a fibration, single-sorted indexed functor P/I is also a fibration.

homomorphism over m . We define the functor $Alg\{-\}_{P/I} \stackrel{def}{=} \{-\}_{P/I}$ from Alg_{F^\top} to Alg_F , 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 μF 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.

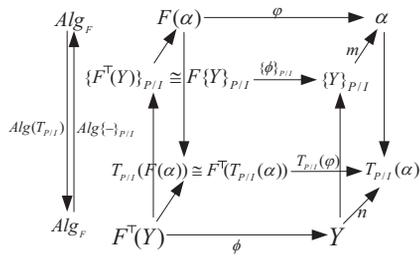


Figure 8. Adjunction properties of $Alg(T_{P/I})$ and $Alg\{-\}_{P/I}$.

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

Let $P : \mathbb{T} \rightarrow \mathbb{B}$ be a bifibration with truth functor T and comprehension functor $\{-\}$, $\forall I \in \mathbf{Obj} \mathbb{B}$ is a discrete indexed object in base category \mathbb{B} . F is an endo-functor in base category \mathbb{B}/I , and the carrier of initial F -algebra is μF , 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.

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

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 $\mathbb{T}/T(I)$, 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 $(\mu F^\top, in^\top : F^\top(\mu F^\top) \rightarrow \mu F^\top)$ to (Y, q) . For $\forall \alpha \in \mathbf{Obj} \mathbb{B}/I$, $Y \in \mathbf{Obj} \mathbb{T}/T(I)$, inductive rule of single-sorted indexed inductive data types with universality is as follows:

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

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

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

Example 2. Element types of single-sorted indexed 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, $\forall I \in \mathbf{Obj} \mathbb{B}$. For any stream $\alpha : S \rightarrow I \in \mathbf{Obj} \mathbb{B}/I$, $F : \alpha \rightarrow I \times \alpha$ is an endo-functor in \mathbb{B}/I , $head : \alpha \rightarrow I$ is head function of this stream, and $tail : \alpha \rightarrow \alpha$ is tail function after cut head element in the stream. Let stream type $Stream(I)$ be the carrier μF of initial F -algebra in base category \mathbb{B}/I , for each stream property $Y \in \mathbf{Obj} \mathbb{T}/T(I)$ in total category $\mathbb{T}/T(I)$ on single-sorted indexed fibration P/I , such as merging and inverting, 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 $Stream(I)$; a recursive operation by initial property of initial F^\top -algebra describes semantics of $Stream(I)$. If q lies above r , then $Ind'_{Gen} Yq$ is an F^\top -algebra homomorphism above $fold\ r$, 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 $\{Y(\alpha) | \forall \alpha \in \mathbf{Obj}\ \mathbb{B}/I\}$ depicting properties of $Stream(I)$.

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 \mathbb{C} , present semantic model of many-sorted indexed fibration, describe many-sorted indexed inductive data types in base category \mathbb{B} indexed by $\mathbf{Obj}\ \mathbb{C}$, make semantic logic model of many-sorted indexed inductive data types in indexed category \mathbb{C} by the fibration $G : \mathbb{B} \rightarrow \mathbb{C}$, 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 $G : \mathbb{B} \rightarrow \mathbb{C}$ 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 $\forall a \in \mathbf{Obj}\ \mathbb{C}$, T_a is a fiber in total category \mathbb{T} on fibration GP over a . The restriction $P_a : T_a \rightarrow \mathbb{B}_a$ of P at a is a pullback of P along including functor $Inc : \mathbb{B}_a \rightarrow \mathbb{B}$, and \mathbb{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 \mathbb{C} the set of T_a constructs the truth functor T of fibration P , namely, $T = \{T_a | \forall a \in \mathbf{Obj}\ \mathbb{C}\}$. But $F : \mathbb{B}_a \rightarrow \mathbb{B}_a$ is indeed an endo-functor in fiber \mathbb{B}_a not in base category \mathbb{B} , so whether its lifting F_G^\perp is also an endo-functor in total category \mathbb{T} is not decidable. Similar to undesirability of F_G^\perp , 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 $G : \mathbb{B} \rightarrow \mathbb{C}$ 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 $P : \mathbb{T} \rightarrow \mathbb{B}$ and $G : \mathbb{B} \rightarrow \mathbb{C}$ be two fibrations between small categories, P is a fibered fibration over G , then for $\forall a \in \mathbf{Obj} \mathbb{C}$, a restriction $P_a : \mathbb{T}_a \rightarrow \mathbb{B}_a$ of P at a is also a fibered fibration.

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

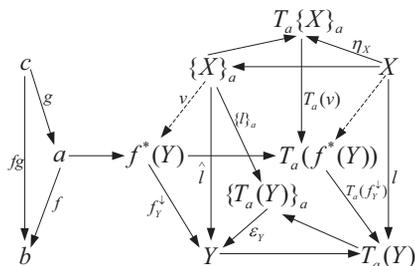


Figure 9. Truth functor T_a preserves Cartesian arrows.

Let $\eta : \mathbf{1}_{\mathbb{T}_a} \rightarrow T_a\{-\}_a$ and $\varepsilon : \{-\}_a T_a \rightarrow \mathbf{1}_{\mathbb{B}_a}$ be two natural transformations, the transpose $\hat{l} = \varepsilon_Y\{l\}_a$ of l lies above $f g$, then in fiber \mathbb{B}_a there exists a unique morphism $v : \{X\}_a \rightarrow f^*(Y) \in \mathbf{Mor} \mathbb{B}_a$ over g such that diagram commutes $f_Y^\downarrow v = \hat{l}$. Henceforth, in fiber \mathbb{T}_a we obtain a unique morphism $(T_a(v))\eta_X : X \rightarrow T_a(f^*(Y)) \in \mathbf{Mor} \mathbb{T}_a$ over g such that $T_a(f_Y^\downarrow)(T_a(v))\eta_X = l$, so $T_a(f_Y^\downarrow)$ 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 $T_a \dashv \{-\}_a$, η and ε is unit and co-unit of this adjunction, and η is vertical morphism, that is, the restriction $P_a : \mathbb{T}_a \rightarrow \mathbb{B}_a$ of P at a is also a fibered fibration.

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

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

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

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

For any F_G^\perp -algebra $(Z, \tau : F_G^\perp(Z) \rightarrow Z)$, the comprehension functor $\{-\}_a$ of many-sorted indexed fibration P_a maps the F -algebra (Z, τ)

Let $(EVEN, ODD)$ be carrier $\mu F = (\mu F_E, \mu F_O)$ of initial F -algebra over binary production in base category $(\mathbb{B}_a, \mathbb{B}_b)$ with respect to many-sorted indexed fibration (P_a, P_b) , for any F -algebra $m : F(E, O) \rightarrow (E, O)$, it is lifted to be a F_G^\perp -algebra $n : F_G^\perp(Z, Z') \rightarrow (Z, Z')$ by many-sorted indexed fibration (P_a, P_b) , which satisfies diagram commutes $(F(P_a, P_b))(Z, Z') = ((P_a, P_b)F_G^\perp)(Z, Z')$. A recursive operation *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^\perp -algebra depicts semantics of $(EVEN, ODD)$. If n lies above m , then $Ind''_{Gen}(Z, Z')n$ is a F_G^\perp -algebra homomorphism over *fold* m , when iterating each objects in total category $(\mathbb{T}_a, \mathbb{T}_b)$ on many-sorted indexed fibration (P_a, P_b) , a semantic set describing properties of $(EVEN, ODD)$ is as follows:

$$\{(Z(E), Z'(O)) \mid \forall E \in EVEN, \forall O \in ODD\}.$$

Mutual recursive is a complex many-sorted indexed 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 *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. Matthes 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 inductive data types 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 adjoint 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. 201200000034) and Shaoguan municipal science and technology plan, China (Grant No. 2013CX/K61).

References

- [1] J. Rutten and D. Turi, "Initial algebra and final coalgebra semantics for concurrency", *Lecture Notes in Computer Science* vol. 666, pp. 477–530, 1993. <http://dx.doi.org/10.1007/3-540-56596-545>
- [2] J. Rutten, "Universal coalgebra: a theory of systems", *Theoretical Computer Science*, vol. 249, no. 1, pp. 3–80, 2000. [http://dx.doi.org/10.1016/S0304-3975\(00\)00056-6](http://dx.doi.org/10.1016/S0304-3975(00)00056-6)
- [3] D. C. Miao *et al.*, "Formal language algebraic model", *Journal of South China University of Technology (Natural Science Edition)*, vol. 39, no. 10, pp. 74–78, October, 2011.
- [4] M. Johnson and R. Rosebrugh, "Fibrations and universal view updatability", *Theoretical Computer Science*, vol. 388, pp. 109–129, 2007. <http://dx.doi.org/10.1016/j.tcs.2007.06.004>
- [5] M. Johnson *et al.*, "Lenses, fibrations and universal translations", *Mathematics Structure in Computer Science*, vol. 22, pp. 25–42, 2012. <http://dx.doi.org/10.1017/S0960129511000442>
- [6] D. C. Miao, "An opfibrations method of database view updates", *Journal of Shaoguan University (Natural Science Edition)*, vol. 34, no. 4, pp. 5–10, 2013.
- [7] H. Tews, "Coalgebra method for object-oriented Specification", Dresden University of Technology, Dresden, Tech. Rep. TUD FI02 08 Aug. 2002.
- [8] N. Ghani *et al.*, "Generic fibrational Induction", *Logical Methods in Computer Science*, vol. 8, no. 2, pp. 1–27, 2012. [http://dx.doi.org/10.2168/LMCS-8\(2:12\)2012](http://dx.doi.org/10.2168/LMCS-8(2:12)2012)
- [9] C. Hermida and B. Jacobs, "Structural induction and coinduction in a fibrational setting", *Information and Computation*, vol. 145, no. 2, pp. 107–152, 1998. <http://dx.doi.org/10.1006/inco.1998.2725>
- [10] M. Barr and C. Wells, "Category Theory for Computing Science" *International Series in Computer Science*, Prentice-Hall, 1990. Second edition, 1995.
- [11] C. Hermida, "Fibrations, Logical predicates and indeterminates," Ph.D. dissertation, University Of Edinburgh, Edinburgh 1993.
- [12] B. Jacobs, "Categorical logic and type theory", *Computing science Institute*, Nijmegen, University of Nijmegen, 1991.
- [13] W. He, *Category theory*. Science Press, Beijing, 2006.
- [14] P. Dybjer, "Inductive families", *Formal Aspects of Computing*, vol. 6, no. 4, pp. 440–465, 1994. <http://dx.doi.org/10.1007/BF01211308>
- [15] P. Morris and T. Altenkirch, "Indexed containers", in *Proceedings of the 24th Symposium on Logic in Computer Science*, pp. 277–285, Los Angeles, California, 2009.

- [16] N. Ghani *et al.*, “Indexed induction and coinduction, fibrationally”, *Logical Methods in Computer Science*, vol. 9 (3–6), pp. 1–31, 2013.
[http://dx.doi.org/10.2168/lmcs-9\(3:6\)2013](http://dx.doi.org/10.2168/lmcs-9(3:6)2013)
- [17] N. Bengt *et al.*, “Programming in Martin-L of type theory: an Introduction”, Nanjing University Press, Nanjing, 2002.
- [18] J. Reynolds, “Polymorphism is not set theoretic”, *Lecture Notes in Computer Science*, vol. 173, pp. 145–156, 1984.
<http://dx.doi.org/10.1007/3-540-13346-17>
- [19] A. Pitts, “Polymorphism is set theoretic, constructively”, *Lecture Notes in Computer Science*, vol. 283, pp. 12–39, 1989.
- [20] M. Hyland, “The effective topos”, *The L. E. J. Browser Centenary Symposium*, North Holland, 1982.
- [21] G. Longo and E. Moggi, “Constructive natural deduction and its ‘Modest’ interpretation”, M. I. T. Press, Massachusetts, 1990.
- [22] T. Coquand and C. Pualin, “Inductively defined Types”, *Lecture Notes in Computer Science*, vol. 417, pp. 50–66, 1990.
http://dx.doi.org/10.1007/3-540-52335-9_47
- [23] P. Dybier, “Inductive sets and families in Martin-Löf’s type theory and their set theoretical semantics”, Cambridge University Press, London, 1991.
- [24] Y. X. Fu, “Recursive models of general inductive Types”, *Fundamenta Informatics*, vol. 26, pp. 115–131, 1996.
- [25] Y. X. Fu, “Constructive semantics of inductive types”, *Journal of Software*, vol. 9, no. 3, pp. 236–240, 1998.
- [26] D. Turi and G. Plotkin, “Towards a Mathematical Operational Semantics”, in *Proceedings of the 12th Symposium on Logic in Computer*, pp. 280–291, Los Angeles, California, 1997.
<http://dx.doi.org/10.1109/LICS.1997.614955>
- [27] N. Ghani *et al.*, “Fibrational induction rules for initial algebras”, *Lecture Notes in Computer Science*, vol. 6247, pp. 336–350, 2010.
http://dx.doi.org/10.1007/978-3-642-15205-4_27
- [28] R. Matthes, “An induction principle for nested data types in intentional type theory”, *Journal of Functional Programming*, vol. 19, no. (3–4), pp. 439–468, 2009.
<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@sgu.edu.cn

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

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

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

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.

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.

YUBIN GUO is an associate professor at South China University of Agriculture, Guangzhou, China. She received her PhD degree in computer application technology from the School of Computer Science and Engineering, South China University of Technology in 2008. Her research interests include database and network computing. She has more than 60 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.
