

From Parametricity to Conservation Laws, via Noether’s Theorem

Robert Atkey

bob.atkey@gmail.com

Abstract

Invariance is of paramount importance in programming languages and in physics. In programming languages, John Reynolds’ theory of relational parametricity demonstrates that parametric polymorphic programs are invariant under change of data representation, a property that yields “free” theorems about programs just from their types. In physics, Emmy Noether showed that if the action of a physical system is invariant under change of coordinates, then the physical system has a conserved quantity: a quantity that remains constant for all time. Knowledge of conserved quantities can reveal deep properties of physical systems. For example, the conservation of energy is by Noether’s theorem a consequence of a system’s invariance under time-shifting.

In this paper, we link Reynolds’ relational parametricity with Noether’s theorem for deriving conserved quantities. We propose an extension of System $F\omega$ with new kinds, types and term constants for writing programs that describe classical mechanical systems in terms of their Lagrangians. We show, by constructing a relationally parametric model of our extension of $F\omega$, that relational parametricity is enough to satisfy the hypotheses of Noether’s theorem, and so to derive conserved quantities for free, directly from the polymorphic types of Lagrangians expressed in our system.

Categories and Subject Descriptors D.3.1 [Programming Languages]: Formal Definitions and Theory; J.2 [Physical Sciences and Engineering]: Physics

General Terms Languages, Theory, Types, Physics

Keywords classical mechanics; relational parametricity; higher-kinded types; invariance; conservation laws

1. Introduction

Reynolds’ theory of relational parametricity [18] tells us that parametrically polymorphic programs automatically satisfy invariance properties. Such invariance properties are often called *Free Theorems*, after Wadler [22], since they follow “for free” from the types of programs, rather than through detailed study of the program text. An illustrative example is the free theorem for programs f with the following type:

$$f : \forall \alpha. \text{List } \alpha \rightarrow \text{Nat}$$

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

POPL ’14, January 22–24, 2014, San Diego, CA, USA.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-2544-8/14/01...\$15.00.

<http://dx.doi.org/10.1145/2535838.2535867>

Such programs take lists of α s, for any type α , and return natural numbers. Using Reynolds’ theory of relational parametricity, Wadler showed that *any* f with this type satisfies the following property:

$$\forall \alpha, \beta, g : \alpha \rightarrow \beta, l : \text{List } \alpha. f [\alpha] l = f [\beta] (\text{map } g l)$$

Thus, any f with the type given above is invariant under mapping some arbitrary function g over its input. Thinking in terms of abstract data types and change of data representation, this free theorem states that f is invariant under change of data representation from an arbitrary type α to another arbitrary type β , via g .

The use of invariance under change to derive useful consequences is a technique much older than programming languages. In physics, *Noether’s theorem* [14] provides a general way to derive conservation laws for physical systems from invariance properties. Informally speaking, Noether’s theorem states that every continuous symmetry of a physical system implies the existence of a corresponding conserved quantity. A conserved quantity is a function of the state of the system that is constant for all time. We illustrate the use of Noether’s theorem with a simple example of a classical system consisting of two particles of equal mass m connected by a spring with spring constant k , such that the particles are constrained to move one-dimensionally.

Noether’s theorem applies to systems described in terms of *Lagrangians*. For classical mechanics, a Lagrangian is a function of time and the positions and velocities of all the particles in the system, to the difference between the kinetic energy and the potential energy of the system. For our system with two particles at positions x_1 and x_2 with velocities \dot{x}_1 and \dot{x}_2 the Lagrangian is:

$$L(t, x_1, x_2, \dot{x}_1, \dot{x}_2) = \frac{1}{2}m(\dot{x}_1^2 + \dot{x}_2^2) - \frac{1}{2}k(x_1 - x_2)^2 \quad (1)$$

The first summand represents the total kinetic energy of the system in terms of the mass and velocities, while the second summand represents the potential energy contained within the spring due to the distance between the particles.

From the Lagrangian (1), using the *principle of stationary action*, we can derive the following two equations of motion for this system. (We describe the principle of stationary action, and the process for deriving the equations of motion fully in Section 2.) For this system, the equations of motion are a pair of ordinary differential equations (ODEs) that describe how the positions and velocities of the particles evolve over time (the second derivative with respect to time, \ddot{x}_i , denotes the acceleration of the i th particle):

$$m\ddot{x}_1 = -k(x_1 - x_2) \quad m\ddot{x}_2 = -k(x_2 - x_1)$$

We could now proceed to solve these ODEs to further analyse the behaviour of this system. However, Noether’s theorem gives us a powerful way of gaining insight into properties of these ODEs “for free”, without necessarily having to find solutions to them. This is accomplished by finding invariance properties of the Lagrangian.

The Lagrangian (1) does not refer to any fixed point in space; only the relative distance between the two particles, along with their velocities, is relevant. Therefore, the Lagrangian (1) is invariant under translation in space by some arbitrary displacement y :

$$L(t, x_1, x_2, \dot{x}_1, \dot{x}_2) = L(t, x_1 + y, x_2 + y, \dot{x}_1, \dot{x}_2) \quad (2)$$

By Noether’s theorem, invariance under spatial translation implies that the linear momentum of the whole system is constant for all time. For this system, conservation of linear momentum is stated mathematically as the vanishing of the derivative with respect to time of the total linear momentum of the system:

$$\frac{d}{dt} m(\dot{x}_1 + \dot{x}_2) = 0 \quad (3)$$

In general, Noether’s theorem gives us a way of deriving conserved properties like (3) from invariance properties like (2). In this case, we have used invariance under translation in space to derive conservation of linear momentum. Other common examples include invariance under translation in time, yielding conservation of energy, and invariance under rotation, yielding conservation of angular momentum. We will see examples of each of these kinds of invariance, and their consequent conservation laws, in Section 5.

The invariance property stated in Equation (2) is highly reminiscent of the free theorem we stated for the polymorphic program at the start of this introduction. A change in representation in the input, whether a change in data representation from a type α to a type β , or a change in where time zero is counted from, results in no change in the output. Since Reynolds’ relational parametricity allowed us to derive the invariance properties for the program f just by looking at its type, we ask the following question:

Question: *Is it possible to use Reynolds’ theory of relational parametricity to derive the invariance properties of Lagrangians required for Noether’s theorem to derive conservation laws?*

In this paper, we answer this question positively. We show that it is indeed the case that we can use a generalised version of Reynolds’ theory of relational parametricity to prove the geometric invariance properties required for the hypotheses of Noether’s theorem. We construct an extension of System $F\omega$ suitable for writing invariant Lagrangians, with a relationally parametric model that allows us to prove invariance properties as free theorems. In our system, the Lagrangian (1) we gave above for describing a system of two particles coupled by a spring will have the type:

$$\forall y: T(1). C^\infty(\mathbb{R}\langle 1, 0 \rangle \times \mathbb{R}\langle 1, y \rangle \times \mathbb{R}\langle 1, y \rangle \times \mathbb{R}\langle 1, 0 \rangle \times \mathbb{R}\langle 1, 0 \rangle, \mathbb{R}\langle 1, 0 \rangle)$$

We explain this type by breaking it down from left to right. The quantifier $\forall y: T(1)$ indicates that we are quantifying over all translations y in one-dimensional space, just as we did implicitly in Equation (2). The notation $C^\infty(-, -)$ denotes the type of *smooth* functions between spaces. (A smooth function is a function that is infinitely differentiable.) All the functions that we wish to use as Lagrangians must be smooth (at least up to second order) for the theory to work. The type $\mathbb{R}\langle g, x \rangle$ denotes real numbers that vary with some linear transformation g and translation x . In this case, all the linear transformations are 1, indicating no transformation. Likewise, when the translation component is 0, there is no translation. Thus a value of type $\mathbb{R}\langle 1, 0 \rangle$ is invariant, but a value of type $\mathbb{R}\langle 1, y \rangle$ varies with the translation y . In the smooth function type above, the domain space consists of vectors of five real numbers: the time, the two particle positions, which vary with the translation y , and the two particle velocities.

Atkey, Johann and Kennedy [5], building on work by Kennedy [11], have already presented a polymorphic type system for expressing geometric invariance properties similar to the translation invariance property in Equation (2). In the present work, we extend

their type system by embedding it within System $F\omega$. The benefits of this embedding are twofold. Firstly, by using the richer type-level structure of System $F\omega$, we can easily add useful indexed types like length indexed vectors and smooth function types, both of which would have required special treatment in the special purpose type system presented by Atkey *et al.* Secondly, by using System $F\omega$ as our base, we are able to reuse the reflexive graph-based relationally parametric semantics presented by Atkey [3], extending it as appropriate for our geometric setting.

Contributions Our core contributions are threefold:

1. We reformulate the type system for geometric invariance of Atkey, Johann and Kennedy [5] as an extension of System $F\omega$. This shows that the special status of the group-indexed types in Atkey *et al.*’s work can be incorporated into a standard framework for indexed types.
2. We present a relationally parametric semantics of our extension of System $F\omega$. In this model, following Atkey’s model for basic System $F\omega$ [3], each kind is interpreted as a reflexive graph. Our key technical contribution is to note that the many geometric groups that we wish to incorporate are expressible as groupoids, and hence as reflexive graphs. We therefore show that Reynolds’ theory of relational parametricity, suitably generalised to indexed types by considering reflexive graphs, already accommodates geometric invariance properties.
3. Finally, we connect the free theorems derivable from our relationally parametric semantics with Noether’s theorem for deriving conservation laws. We present many examples of invariant Lagrangians expressible in our type system, along with the consequent conservation laws for physical systems. In each case, the necessary invariance property is derived from the type of the Lagrangian.

Outline

- In the next section, Section 2, we describe the necessary background to understand Noether’s theorem and the conservation laws it generates. We briefly introduce the Lagrangian formulation of classical mechanics, define precisely what it means for a physical system to be invariant under transformations, and state Noether’s theorem.
- Having shown how invariance generates conservation laws in Section 2, in Section 3 we show how to derive invariance properties from indexed types. We present a relationally parametric semantics for System $F\omega$, and show how geometric invariance properties can be accommodated within this semantics.
- In Section 4 we construct a type system for classical mechanics by extending System $F\omega$ with the necessary constants for smooth function types and combinators for constructing smooth functions. Programming directly with our combinators in raw System $F\omega$ is awkward, due to the point-free style required, so in Section 4.2 we define a surface syntax for defining smooth functions that is translated into our applied System $F\omega$.
- In Section 5 we present several examples of using our extension of System $F\omega$ for writing invariant Lagrangians that describe physical systems.
- Section 6 concludes, and offers directions for further work.

2. Conservation Laws from Invariance

Noether’s theorem applies to systems described using the formalism of Lagrangian mechanics, a reformulation of Newtonian classical mechanics. In this section we can only offer a very brief introduction to the concepts of Lagrangian mechanics. The reader

is referred to other sources, such as Landau and Lifschitz [12] or Arnol'd [2] for the necessary background in the mathematical theory of classical mechanics. Lagrangian mechanics, and Noether's theorem, make crucial use of the Calculus of Variations. Gelfand and Fomin [7] provide a good introduction to this topic.

In this section, we will endeavour to keep the presentation relatively elementary. However, some knowledge of differential and integral calculus will be required. Knowledge of basic Newtonian mechanics will also be helpful. We will use standard notation from physics and mathematics for integrals and (partial) derivatives. In particular, when discussing derivatives with respect to time, we use dot notation: \dot{x} for the first derivative of x with respect to time, and \ddot{x} for the second derivative.

2.1 Lagrangian Mechanics

Lagrangian mechanics is a branch of analytical mechanics, along with Hamiltonian mechanics [2, 12]. Analytical mechanics seeks to discover the underlying structure of Newtonian classical mechanics by reformulating it in different terms.

Lagrangian and Action For Lagrangian mechanics, classical mechanical systems are described in terms of *Lagrangians* $L = T - V$, where T is the total kinetic energy of the system, and V is the total potential energy. The Lagrangian is a function of time, the positions of all the particles in the system, and their velocities. This is usually written like so, with abuse of notation for \dot{q} which is a variable here, not the derivative of a function:

$$L(t, q, \dot{q}) = T - V$$

In this expression, q and \dot{q} are actually vectors containing the components of the position and velocity vectors for each particle. A key feature of Lagrangian mechanics is that the coordinate system used to describe the system need not necessarily be the usual cartesian coordinates. For example, one of the components of q might be the angle of swing of a pendulum. This use of generalised coordinates is one of the advantages of Lagrangian mechanics over the Newtonian presentation in terms of positions and forces.

The Lagrangian for a particular system is used to define the *action* of that system. For a particular path $q : \mathbb{R} \rightarrow \mathbb{R}^n$ and endpoints $a < b$ in time, the action of the Lagrangian L over this path is defined by the following integral:

$$\mathcal{S}[q; a; b] = \int_a^b L(t, q(t), \dot{q}(t)) dt$$

Below, we will use the principle of stationary action to select the physically realisable paths from all the possible paths $q : \mathbb{R} \rightarrow \mathbb{R}^n$.

Example 1. We have already seen an example of a Lagrangian, in the introduction, describing a system of two particles coupled by a spring. Another example of a Lagrangian describes a system consisting of a single particle of mass m acted upon by a constant downward gravitational field with gravitational potential g :

$$L(t, x, y, \dot{x}, \dot{y}) = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - mgy \quad (4)$$

Again, the left summand represents the kinetic energy of the system in terms of the mass and horizontal and vertical velocities of the particle. The right summand represents the potential energy of the system due to the gravitational field's strength determined by the particle's distance from a fixed baseline at $y = 0$.

The Principle of Stationary Action The *principle of stationary action* states that the physically realisable paths q are the paths that yield stationary points (i.e., either minima or maxima) of the action $\mathcal{S}[q; a; b]$, for all a and b . Defining what is exactly meant by a stationary point of the action is beyond the scope of this paper, and can be found in standard references on the calculus of variations,

such as Gelfand and Fomin [7]. The key point for our purposes is that a path q satisfies the principle of stationary action if and only if it satisfies a system of ODEs derived from the Lagrangian L , called the Euler-Lagrange equations:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0 \quad (5)$$

The ODEs derived from the Lagrangian for a classical mechanical system are called the *equations of motion* for the system.

The notation in the Euler-Lagrange equation, while standard, is not exactly clear, especially to a programming language theorist used to a more careful treatment of variables. The two partial derivatives $\frac{\partial L}{\partial \dot{q}_i}$ and $\frac{\partial L}{\partial q_i}$ take partial derivatives of L with respect to \dot{q}_i and q_i as variables. The derivative with respect to time, $\frac{d}{dt}$, then treats \dot{q}_i as a function of time, hence the appearance of the second derivatives \ddot{q}_i in the equations of motion. Sussman and Wisdom [20] criticise the standard presentation of Lagrangian mechanics for this kind of confusing notation, and define a clearer notation based on the Scheme programming language. We elect to stick with the standard notation as used in physics textbooks just for familiarity.

Example 2. Computing the Euler-Lagrange equations for the Lagrangian (4), above, yields a pair of ODEs, one each for the x and y coordinates of the single particle of the system:

$$m\ddot{x} = 0 \quad m\ddot{y} = -mg$$

Note that both these ODEs have the form $F = m\ddot{x}$, just as in Newton's second law. Lagrangian mechanics allows, from the principle of stationary action, the derivation of Newton's second law, instead of postulating it as an axiom.

2.2 Noether's Theorem

As we described in the introduction, Noether's theorem provides us with deep insights into the properties of the solutions of the Euler-Lagrange equations. Formally, Noether's theorem requires the invariance of the action \mathcal{S} , rather than invariance of the Lagrangian itself. We therefore first define what is meant by invariance of the action, and then state Noether's theorem.

Invariance of the Action Let

$$\mathcal{S}[q; a; b] = \int_a^b L(t, q(t), \dot{q}(t)) dt$$

be the action of some physical system described by the Lagrangian L . Assume a differentiable invertible function $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ that transforms time in some way and a function $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ that transforms the vector of generalised coordinates \vec{q} to another vector of generalised coordinates $\Psi(\vec{q})$.

The action \mathcal{S} is *invariant* between the endpoints a and b , under the transformations Φ and Ψ , if it is the case that for all paths q :

$$\int_a^b L(t, q(t), \dot{q}(t)) dt = \int_{\Phi(a)}^{\Phi(b)} L(s, q^*(s), \frac{dq^*}{ds}(s)) ds \quad (6)$$

where $q^*(s) = (\Psi \circ q \circ \Phi^{-1})(s)$ is the path q transformed by Φ and Ψ . By change of variables, the right hand integral in the above equation is equal to the following integral:

$$\int_a^b L(\Phi(t), q^*(\Phi(t)), \frac{dq^*}{ds}(\Phi(t))) \cdot |\dot{\Phi}(t)| dt$$

For our application to classical mechanics, the endpoints a and b are arbitrary, so we can rewrite Equation (6) to the following equation between applications of the Lagrangian:

$$L(t, q(t), \dot{q}(t)) = L(\Phi(t), q^*(\Phi(t)), \frac{dq^*}{ds}(\Phi(t))) \cdot |\dot{\Phi}(t)| \quad (7)$$

For most examples, it will be the case that Φ has the form $\Phi(t) = t + t'$ and Ψ is an affine transformation $\Psi(q) = Gq + x$, where G is an $n \times n$ real matrix and $x \in \mathbb{R}^n$. In this case, Equation (7) simplifies further, using the definition of q^* and the chain rule of differentiation, to:

$$L(t, q(t), \dot{q}(t)) = L(t + t', Gq(t) + x, G\dot{q}(t)) \quad (8)$$

Equation (8) is obviously implied by the following equation, where we treat q and \dot{q} as universally quantified variables, instead of as functions of time:

$$L(t, q, \dot{q}) = L(t + t', Gq + x, G\dot{q}) \quad (9)$$

Equation (9) is the form of invariance that the free theorems we derive from the types of our typed language for classical mechanics will usually take. We define our type system for classical mechanics in Sections 3 and 4 below.

Example 3. In the introduction, we stated that the Lagrangian (1) describing the spring-coupled system is invariant under translation in space. Our running example in this section, the Lagrangian (4) for the single particle under gravity, is not invariant under all spatial translations, due to the reference in the potential energy term to the distance from the fixed baseline at $y = 0$. It is invariant under translations along the x -axis, however. Both Lagrangians are also invariant under translations in time, a special case of Equation (9):

$$L(t, q, \dot{q}) = L(t + t', q, \dot{q})$$

This invariance property is simply a consequence of the fact that neither Lagrangian explicitly mentions the time t .

Noether's Theorem Noether's theorem applies to actions that are *continuously* invariant. Being continuously invariant means that, instead of having fixed transformations Φ and Ψ , we have two families of transformations, Φ_ϵ and Ψ_ϵ , differentially indexed by a real parameter ϵ , such that Φ_0 and Ψ_0 are the identity function.

Theorem 1 (Noether). *If the action*

$$\mathcal{S}[q; a; b] = \int_a^b L(t, q(t), \dot{q}(t)) dt$$

is invariant under ϵ -indexed families of transformations Φ_ϵ and Ψ_ϵ , then the equation

$$\frac{d}{dt} \left(\sum_{i=1}^n \frac{\partial L}{\partial \dot{q}_i} \psi_i + \left(L - \sum_{i=1}^n \dot{q}_i \frac{\partial L}{\partial \dot{q}_i} \right) \phi \right) = 0 \quad (10)$$

holds for all paths q satisfying the Euler-Lagrange equations (5), where $\psi = \left. \frac{\partial \Psi}{\partial \epsilon} \right|_{\epsilon=0}$ and $\phi = \left. \frac{\partial \Phi}{\partial \epsilon} \right|_{\epsilon=0}$.

Proof. E.g., Gelfand and Fomin [7], Section 20. \square

Example 4. In Example 3, we noted that the Lagrangians (1) and (4) are both invariant under arbitrary translations in time. In terms of continuous invariance, their corresponding actions are invariant under the families $\Phi_\epsilon(t) = t + \epsilon$ and $\Psi_\epsilon(q) = q$. Plugging these definitions into Equation (10), we learn that, for these systems:

$$\frac{d}{dt} \left(L - \sum_{i=1}^2 \dot{q}_i \frac{\partial L}{\partial \dot{q}_i} \right) = 0$$

For the spring-coupled particles Lagrangian, (1), we obtain:

$$\frac{d}{dt} \left(\frac{1}{2} m(\dot{x}_1^2 + \dot{x}_2^2) + \frac{1}{2} k(x_1 - x_2)^2 \right) = 0$$

and for the single particle under gravity Lagrangian (4), we obtain:

$$\frac{d}{dt} \left(\frac{1}{2} m(\dot{x}^2 + \dot{y}^2) + mgy \right) = 0$$

In both cases, we have discovered that, as a consequence of invariance under translation in time, the total energy of the system is conserved (note that the $-$ signs in the original Lagrangians have turned into $+$ s). In general, invariance under translation in time implies, via Noether's theorem, conservation of energy.

This finishes our short introduction to Lagrangian mechanics and Noether's theorem. We have shown how Noether's theorem can be used to derive conservation laws from invariance properties. We now go on to showing how suitable invariance properties can be derived from types, building towards Section 4, where we construct a type system for writing Lagrangians that expresses invariance properties directly in the types. In Section 5, we give examples of invariant Lagrangians and their consequent conservation laws.

3. Invariance from Types

We derive invariance properties from types by constructing a relationally parametric model of System $F\omega$, which we then extend with suitable kind-, type-, and term-level constants for writing invariant Lagrangians. Our relationally parametric model is based on the reflexive graph model presented by Atkey [3] (based on previous work on reflexive graph models of relational parametricity by Robinson and Rosolini [19] and Hasegawa [9]). In reflexive graph models of relational parametricity, the kinds of System $F\omega$ are interpreted as reflexive graphs, while types are interpreted as morphisms of reflexive graphs. As demonstrated by Atkey, uniformly interpreting kinds as reflexive graphs allows for a straightforward interpretation of higher kinds like $* \rightarrow *$. In the present setting, reflexive graphs will allow us to accommodate arbitrary groupoids as kinds (Section 3.3). Groupoid kinds are the key technical tool that we need to derive the kinds of geometric invariance properties we need for Noether's theorem.

3.1 System $F\omega$: Syntax and Relationally Parametric Model

We now present the syntax and relationally parametric semantics of System $F\omega$. Since we will incrementally extend the syntax of the system throughout this section and the next, we do not state a final triple of theorems stating that we have correctly interpreted kinds, types and terms. Rather, we state the necessary properties of the interpretation as three separate properties (Properties 1, 2, and 3, below), and maintain them as we extend the system.

We deviate from the standard presentation of System $F\omega$ in that we distinguish between large and small kinds, and only allow quantification over small kinds. In particular, the kind of types $*$ is *not* small. The reason for this choice is that we desire a simple set-theoretic model. However, requiring that we have a set of sets closed under large products, in order to interpret quantification over the kind $*$, would force us to use an intuitionistic metatheory [17]. The basic results of the Calculus of Variations that we are relying on, such as the derivation of the Euler-Lagrange equations, use excluded middle in their proofs, so an intuitionistic metatheory is not an option. We therefore just assume that we have a set of "small" sets, which includes the real numbers and is closed under subsets, products, function spaces and set comprehensions.

Syntax The kinds of System $F\omega$ include at least the base kind of types, $*$, and function and product kinds, as generated by the following grammar:

$$\kappa ::= * \mid \kappa_1 \rightarrow \kappa_2 \mid \kappa_1 \times \kappa_2 \mid \dots$$

We will extend the collection of kinds in Section 3.2 with type-level natural numbers, and in Section 3.3 with groupoid kinds.

We use the judgement κ small to denote when the kind κ will be interpreted by a small reflexive graph (to be defined below). The kind $*$ is not small, but the construction of function and product

$$\begin{array}{c}
\frac{\alpha : \kappa \in \Theta}{\Theta \vdash \alpha : \kappa} \quad \frac{\Theta, \alpha : \kappa_1 \vdash A : \kappa_2}{\Theta \vdash \lambda \alpha : \kappa_1. A : \kappa_1 \rightarrow \kappa_2} \\
\\
\frac{\Theta \vdash F : \kappa_1 \rightarrow \kappa_2 \quad \Theta \vdash A : \kappa_1}{\Theta \vdash FA : \kappa_2} \\
\\
\frac{\Theta \vdash A : \kappa_1 \quad \Theta \vdash B : \kappa_2}{\Theta \vdash \langle A, B \rangle : \kappa_1 \times \kappa_2} \quad \frac{\Theta \vdash A : \kappa_1 \times \kappa_2}{\Theta \vdash \pi_i A : \kappa_i} \quad i \in \{1, 2\} \\
\\
\frac{\Theta \vdash A : * \quad \Theta \vdash B : *}{\Theta \vdash A \times B : *} \quad \frac{\Theta \vdash A : * \quad \Theta \vdash B : *}{\Theta \vdash A \rightarrow B : *} \\
\\
\frac{\Theta, \alpha : \kappa \vdash A : * \quad \kappa \text{ small}}{\Theta \vdash \forall \alpha : \kappa. A : *}
\end{array}$$

Figure 1. Types and their Kinds

$$\begin{array}{c}
\frac{\Theta, \alpha : \kappa_1 \vdash A : \kappa_2 \quad \Theta \vdash B : \kappa_1}{\Theta \vdash (\lambda \alpha : \kappa_1. A) B \equiv A\{B/\alpha\} : \kappa_2} \beta \\
\\
\frac{\Theta \vdash A : \kappa_1 \rightarrow \kappa_2}{\Theta \vdash (\lambda \alpha : \kappa_1. A \alpha) \equiv A : \kappa_1 \rightarrow \kappa_2} \eta \\
\\
\frac{\Theta \vdash A_1 : \kappa_1 \quad \Theta \vdash A_2 : \kappa_2}{\Theta \vdash \pi_i \langle A_1, A_2 \rangle \equiv A_i : \kappa_i} \beta \\
\\
\frac{\Theta \vdash A : \kappa_1 \times \kappa_2}{\Theta \vdash \langle \pi_1 A, \pi_2 A \rangle \equiv A : \kappa_1 \times \kappa_2} \eta \\
\\
\text{plus: reflexivity, symmetry, transitivity and congruence}
\end{array}$$

Figure 2. Type equality

kinds preserves smallness:

$$\frac{\kappa_1 \text{ small} \quad \kappa_2 \text{ small}}{\kappa_1 \rightarrow \kappa_2 \text{ small}} \quad \frac{\kappa_1 \text{ small} \quad \kappa_2 \text{ small}}{\kappa_1 \times \kappa_2 \text{ small}}$$

The additional kinds we introduce below will all be small.

The well-kinded types, type equalities, and well-typed terms of System $F\omega$ are shown in Figures 1, 2 and 3, respectively. Kinding contexts $\alpha_1 : \kappa_1, \dots, \alpha_n : \kappa_n$ are denoted by Θ and typing contexts $x_1 : A_1, \dots, x_n : A_n$ are denoted by Γ . Note that typing contexts Γ only contain well-kinded types of kind $*$. Terms also have an equational theory, with $\beta\eta$ -laws for functions, products, and universal quantification, which we omit. Our presentation of System $F\omega$ is entirely standard (see, e.g., Pierce [16]), except for the restriction to small kinds in the formation of universal kinds $\forall \alpha : \kappa. A$. As is the case for kinds, we will extend the types, type equalities, and terms in Sections 3.2, 3.3 and 4, below.

Reflexive Graphs and the Interpretation of Kinds We will interpret every kind κ as a reflexive graph, which we now define. A *reflexive graph* is a triple (O, R, id) , where O is a large set of objects, $R : O \times O \rightarrow \text{Set}$ assigns a small set of directed ‘edges’ to each pair of objects, and $\text{id} : \Pi o \in O. R(o, o)$ assigns a distinguished ‘identity’ edge from every object to itself. We think of the edges of a reflexive graph as abstract “relations” between the objects. Indeed, in the interpretation of the kind of types, $*$, below, the edges will be exactly relations.

$$\begin{array}{c}
\frac{x : A \in \Gamma}{\Theta \mid \Gamma \vdash x : A} \quad \frac{\Theta \mid \Gamma \vdash e : A \quad \Theta \vdash A \equiv B : *}{\Theta \mid \Gamma \vdash e : B} \\
\\
\frac{\Theta \mid \Gamma \vdash e_1 : A_1 \quad \Theta \mid \Gamma \vdash e_2 : A_2}{\Theta \mid \Gamma \vdash (e_1, e_2) : A_1 \times A_2} \quad \frac{\Theta \mid \Gamma \vdash e : A_1 \times A_2}{\Theta \mid \Gamma \vdash \pi_i e : A_i} \quad i \in \{1, 2\} \\
\\
\frac{\Theta \mid \Gamma, x : A \vdash e : B}{\Theta \mid \Gamma \vdash \lambda x : A. e : A \rightarrow B} \\
\\
\frac{\Theta \mid \Gamma \vdash e_1 : A \rightarrow B \quad \Theta \mid \Gamma \vdash e_2 : A}{\Theta \mid \Gamma \vdash e_1 e_2 : B} \\
\\
\frac{\Theta, \alpha : \kappa \mid \Gamma \vdash e : A \quad \alpha \notin \text{fv}(\Gamma)}{\Theta \mid \Gamma \vdash \Lambda \alpha : \kappa. e : \forall \alpha : \kappa. A} \\
\\
\frac{\Theta \mid \Gamma \vdash e : \forall \alpha : \kappa. A \quad \Theta \vdash B : \kappa}{\Theta \mid \Gamma \vdash e [B] : A\{B/\alpha\}}
\end{array}$$

Figure 3. Terms and their Types

A *small reflexive graph* is a reflexive graph (O, R, id) where O is a small set of objects. We use small reflexive graphs as the semantic interpretation of small kinds.

The interpretation of kinds as reflexive graphs, and small kinds as small reflexive graphs is a key property of our semantics that we will maintain as we add additional kinds in Sections 3.2 and 3.3, below. We state this as Property 1 of our semantics:

Property 1. *Each kind κ is interpreted as a reflexive graph $\llbracket \kappa \rrbracket$. If κ small, then $\llbracket \kappa \rrbracket$ is a small reflexive graph.*

An appealing interpretation of reflexive graphs is as “categories without composition”. Following this intuition, we define morphisms of reflexive graphs as “functors”, without the preservation of composition condition. A *morphism of reflexive graphs* (O_1, R_1, id_1) and (O_2, R_2, id_2) is a pair of mappings $f : O_1 \rightarrow O_2$ and $r : \forall o, o' \in O_1. R_1(o, o') \rightarrow R_2(f o, f o')$ such that identities are preserved: $r o o (\text{id}_1 o) = \text{id}_2 (f o)$. We use morphisms of reflexive graphs below to interpret well-kinded types.

We will use the notation $-^O$, $-^R$ and $-^{\text{id}}$ for the first, second and third projections out of tuples representing reflexive graphs. Similarly, we use $-^f$ and $-^r$ for the first and second projections out of tuples representing reflexive graph morphisms.

We now define the interpretations of the basic kinds of System $F\omega$ we defined above, making sure that we maintain Property 1. At base kind, the collection of objects is simply the (large) set of all small sets; edges between A and B are binary relations on A and B (i.e., subsets of $A \times B$); and the distinguished identity edge is exactly the equality relation:

$$\llbracket * \rrbracket = (\text{Set}, \text{Rel}, \equiv)$$

The reflexive graph $\llbracket * \rrbracket$ is not small, due to the collection of all small sets Set not forming a small set.

For higher kinds $\kappa_1 \rightarrow \kappa_2$, the collection of objects consists of reflexive graph morphisms from the interpretation of κ_1 to the interpretation of κ_2 ; the edges between morphisms (f, r) and (f', r') are edge transformers; and the distinguished identity relation for (f, r) is just r :

$$\begin{aligned}
\llbracket \kappa_1 \rightarrow \kappa_2 \rrbracket = & \\
& (\{(f, r) \mid (f, r) : \llbracket \kappa_1 \rrbracket \rightarrow \llbracket \kappa_2 \rrbracket\}, \\
& ((f, r), (f', r')) \mapsto \Pi o, o'. \llbracket \kappa_1 \rrbracket^R(o, o') \rightarrow \llbracket \kappa_2 \rrbracket^R(f o, f' o'), \\
& (f, r) \mapsto r)
\end{aligned}$$

By the assumption that our collection of small sets is closed under the formation of function spaces and set comprehension, if $\llbracket \kappa_1 \rrbracket$ and $\llbracket \kappa_2 \rrbracket$ are small reflexive graphs, then so is $\llbracket \kappa_1 \rightarrow \kappa_2 \rrbracket$.

Product kinds are interpreted by taking the product of their interpretations as reflexive graphs:

$$\begin{aligned} \llbracket \kappa_1 \times \kappa_2 \rrbracket &= \\ &(\llbracket \kappa_1 \rrbracket^O \times \llbracket \kappa_2 \rrbracket^O, \\ &((o_1, o_2), (o'_1, o'_2)) \mapsto \llbracket \kappa_1 \rrbracket^R(o_1, o'_1) \times \llbracket \kappa_2 \rrbracket^R(o_2, o'_2), \\ &(o_1, o_2) \mapsto (\llbracket \kappa_1 \rrbracket^{\text{id}}(o_1), \llbracket \kappa_2 \rrbracket^{\text{id}}(o_2))) \end{aligned}$$

The collection of objects of the interpretation of a product kind is the product of the underlying collections of objects of the two parts, and the relational component is simply the product of the relational components. This naturally leads to the identity component being defined as the tuple of the identity components of the two parts. Again, by assumption that our collection of small sets is closed under products, if $\llbracket \kappa_1 \rrbracket$ and $\llbracket \kappa_2 \rrbracket$ are small reflexive graphs, then so is $\llbracket \kappa_1 \times \kappa_2 \rrbracket$. Product kinds generalise to the interpretation of kinding contexts $\Theta = \alpha_1 : \kappa_1, \dots, \alpha_n : \kappa_n$, which are interpreted as the product of the reflexive graph interpretations of $\kappa_1, \dots, \kappa_n$:

$$\begin{aligned} \llbracket \alpha_1 : \kappa_1, \dots, \alpha_n : \kappa_n \rrbracket &= \\ &(\llbracket \kappa_1 \rrbracket^O \times \dots \times \llbracket \kappa_n \rrbracket^O, \\ &(\theta, \theta') \mapsto \llbracket \kappa_1 \rrbracket^R(\pi_1 \theta, \pi_1 \theta') \times \dots \times \llbracket \kappa_n \rrbracket^R(\pi_n \theta, \pi_n \theta'), \\ &\theta \mapsto (\llbracket \kappa_1 \rrbracket^{\text{id}}(\pi_1 \theta), \dots, \llbracket \kappa_n \rrbracket^{\text{id}}(\pi_n \theta))) \end{aligned}$$

In Sections 3.2 and 3.3 below, we will extend System $F\omega$ with additional kinds, and assign them reflexive graph interpretations, making sure that we maintain Property 1.

Interpretation of Types Well-kinded types $\Theta \vdash A : \kappa$ from Figure 1 are interpreted as reflexive graph morphisms $\llbracket A \rrbracket : \llbracket \Theta \rrbracket \rightarrow \llbracket \kappa \rrbracket$. We sum this up as a property of our semantics:

Property 2. *Each well-kinded type $\Theta \vdash A : \kappa$ is interpreted as a reflexive graph morphism $\llbracket A \rrbracket : \llbracket \Theta \rrbracket \rightarrow \llbracket \kappa \rrbracket$, such that if $\Theta \vdash A \equiv B : \kappa$ then $\llbracket A \rrbracket = \llbracket B \rrbracket$.*

The interpretation of the λ -calculus fragment (i.e., variables, λ -abstraction and application, and products) of the language of well-kinded types is displayed in Figure 4. The interpretations are unsurprising given the reflexive graph interpretation of the kinds $\kappa_1 \rightarrow \kappa_2$ and $\kappa_1 \times \kappa_2$ we gave above.

Figure 5 shows the interpretations of the basic type constructors for function and product types, and of universal quantification. Each of these constructs builds an object of kind $*$, so the object-level interpretation is a small set in Set , and the relation-level interpretation is an actual relation. In the cases of the function and product types, the object-level interpretation is just as set-theoretic function and product respectively, and the relation-level interpretation uses the standard logical relations interpretations of these type constructors. Universal quantification, $\forall \alpha : \kappa. A$, is interpreted at the object level by taking the dependent product over the objects of the interpretation of κ (this product exists because we have stipulated that κ must be small), and then restricting to those elements of the dependent product that preserve relations. This restriction is required for this interpretation to preserve identity edges, and so be a reflexive graph morphism. The relation-level interpretation of universal quantification is the standard relational interpretation of such types, albeit here generalised to kinds interpreted as arbitrary reflexive graphs.

Well-kinded typing contexts $\Theta \vdash \Gamma$ are interpreted as reflexive graph morphisms $\llbracket \Gamma \rrbracket : \llbracket \Theta \rrbracket \rightarrow \llbracket * \rrbracket$ by taking the product of the interpretations of their constituent types, similar to the interpretation of the product types $A \times B$.

We will extend basic System $F\omega$ with additional types and type equalities in Sections 3.2, 3.3, and 4, below. These new types will

also be assigned interpretations as reflexive graph morphisms, and we will ensure that Property 2 is maintained.

Interpretation of Terms We omit the straightforward and relatively uninteresting interpretation of well-typed terms $\Theta \mid \Gamma \vdash e : A$, and just state that there is a well-defined function interpreting each well-typed term, with the property that it takes related environments to related results; this is the fundamental theorem of logical relations for System $F\omega$:

Property 3. *For all well-typed terms $\Theta \mid \Gamma \vdash e : A$ there is a function $\llbracket e \rrbracket \in (\forall \theta \in \llbracket \Theta \rrbracket^O. \llbracket \Gamma \rrbracket^f \theta \rightarrow \llbracket A \rrbracket^f \theta)$, such that, for all $\theta, \theta' \in \llbracket \Theta \rrbracket^O$, $\rho \in \llbracket \Theta \rrbracket^R(\theta, \theta')$, $\gamma \in \llbracket \Gamma \rrbracket^f \theta$ and $\gamma' \in \llbracket \Gamma \rrbracket^f \theta'$, if $(\gamma, \gamma') \in \llbracket \Gamma \rrbracket^r \theta \theta' \rho$ then $(\llbracket e \rrbracket \theta \gamma, \llbracket e \rrbracket \theta' \gamma') \in \llbracket A \rrbracket^r \theta \theta' \rho$. Moreover, this interpretation is sound for the $\beta\eta$ equational theory of terms.*

3.2 Discrete Kinds

In Section 3.1, we only had a single base kind: the kind $*$ of proper types, with a specific interpretation as the reflexive graph of sets and relations. We now describe two families of base kinds with interpretations that are particular sorts of reflexive graph. In this section, we look at discrete kinds; kinds whose reflexive graph interpretations are such that the reflexive edges are the only edges between objects. In the following section (Section 3.3), we look at groupoid kinds, where edges are composable and invertible.

Discrete kinds can be seen as the natural way of lifting types up to the kind level. For our purposes, we will only require a kind of natural numbers, which we will use for constructing n -ary vectors of positions and velocities when we use our type system for writing Lagrangians in Section 4. The kind of natural numbers is small:

$$\kappa ::= \dots \mid \text{nat} \quad \overline{\text{nat small}}$$

The interpretation of this kind as a reflexive graph goes as follows:

$$\llbracket \text{nat} \rrbracket = (\mathbb{N}, (n_1, n_2) \mapsto \{ * \mid n_1 = n_2 \}, n \mapsto *)$$

The collection of objects of this reflexive graph is simply the set of natural numbers. The only edges in the graph are unique edges between equal numbers. Since we have assumed that our collection of small sets contains the natural numbers, we have clearly maintained Property 1 of our semantics.

We add new type-level constructs for the kind Nat of natural numbers, representing zero and successor, and a kind-generic recursion operator:

$$\frac{}{\Theta \vdash \text{zero} : \text{nat}} \quad \frac{\Theta \vdash A : \text{nat}}{\Theta \vdash \text{succ } A : \text{nat}} \\ \frac{\Theta \vdash A : \text{nat} \quad \Theta \vdash B : \kappa \quad \Theta \vdash C : \kappa \rightarrow \kappa}{\Theta \vdash \text{natrec}_\kappa A B C : \kappa}$$

These three constructs all have the evident interpretations in terms of the inductive structure of the natural numbers, and satisfy the following β -laws, maintaining Property 2 of our semantics:

$$\begin{aligned} \text{natrec}_\kappa \text{ zero } B C &\equiv B \\ \text{natrec}_\kappa (\text{succ } A) B C &\equiv C (\text{natrec}_\kappa A B C) \end{aligned}$$

3.3 Groupoid Kinds

Discrete kinds are useful for lifting types up to the kind level, but they do not really exploit the flexibility of reflexive graphs. Moreover, they do not provide us with the invariance properties we require to apply Noether's theorem. By considering kinds whose interpretations are groupoids, we will be able to derive the invariance properties that we need.

Recall that a groupoid is a category in which all morphisms have inverses [15]. Functors between categories always preserve

$$\begin{aligned}
\llbracket \Theta \vdash \alpha_i : \kappa_i \rrbracket^f \theta &= \pi_i \theta \\
\llbracket \Theta \vdash \alpha_i : \kappa_i \rrbracket^r \theta \theta' \rho &= \pi_i \rho \\
\llbracket \Theta \vdash \lambda \alpha : \kappa_1. A : \kappa_1 \rightarrow \kappa_2 \rrbracket^f \theta &= (\lambda o \in \llbracket \kappa_1 \rrbracket^O. \llbracket A \rrbracket^f(\theta, o), \lambda o, o' \in \llbracket \kappa_1 \rrbracket^O, r \in \llbracket \kappa_1 \rrbracket^R(o, o'). \llbracket A \rrbracket^r(\theta, o)(\theta, o')(\llbracket \Theta \rrbracket^{\text{id}} \theta, r)) \\
\llbracket \Theta \vdash \lambda \alpha : \kappa_1. A : \kappa_1 \rightarrow \kappa_2 \rrbracket^r \theta \theta' \rho &= \lambda o, o' \in \llbracket \kappa_1 \rrbracket^O, r \in \llbracket \kappa_1 \rrbracket^R(o, o'). \llbracket A \rrbracket^r(\theta, o)(\theta', o')(\rho, r) \\
\llbracket \Theta \vdash F A : \kappa_2 \rrbracket^f \theta &= \pi_1(\llbracket F \rrbracket^f \theta) (\llbracket A \rrbracket^f \theta) \\
\llbracket \Theta \vdash F A : \kappa_2 \rrbracket^r \theta \theta' \rho &= \llbracket F \rrbracket^r \theta \theta' \rho (\llbracket A \rrbracket^f \theta) (\llbracket A \rrbracket^f \theta') (\llbracket A \rrbracket^r \theta \theta' \rho) \\
\llbracket \Theta \vdash \langle A, B \rangle : \kappa_1 \times \kappa_2 \rrbracket^f \theta &= (\llbracket A \rrbracket^f \theta, \llbracket B \rrbracket^f \theta) \\
\llbracket \Theta \vdash \langle A, B \rangle : \kappa_1 \times \kappa_2 \rrbracket^r \theta \theta' \rho &= (\llbracket A \rrbracket^r \theta \theta' \rho, \llbracket B \rrbracket^r \theta \theta' \rho) \\
\llbracket \Theta \vdash \pi_i A : \kappa_i \rrbracket^f \theta &= \pi_i(\llbracket A \rrbracket^f \theta) \\
\llbracket \Theta \vdash \pi_i A : \kappa_i \rrbracket^r \theta \theta' \rho &= \pi_i(\llbracket A \rrbracket^r \theta \theta' \rho)
\end{aligned}$$

Figure 4. Interpretation of type-level λ -calculus as reflexive graph morphisms

$$\begin{aligned}
\llbracket \Theta \vdash A \rightarrow B : * \rrbracket^f \theta &= \llbracket A \rrbracket^f \theta \rightarrow \llbracket B \rrbracket^f \theta \\
\llbracket \Theta \vdash A \rightarrow B : * \rrbracket^r \theta \theta' \rho &= \{(f, f') \mid \forall (a, a') \in \llbracket A \rrbracket^r \theta \theta' \rho. (f a, f' a') \in \llbracket B \rrbracket^r \theta \theta' \rho\} \\
\llbracket \Theta \vdash A \times B : * \rrbracket^f \theta &= \llbracket A \rrbracket^f \theta \times \llbracket B \rrbracket^f \theta \\
\llbracket \Theta \vdash A \times B : * \rrbracket^r \theta \theta' \rho &= \{(a, b), (a', b') \mid (a, a') \in \llbracket A \rrbracket^r \theta \theta' \rho, (b, b') \in \llbracket B \rrbracket^r \theta \theta' \rho\} \\
\llbracket \Theta \vdash \forall \alpha : \kappa. A : * \rrbracket^f \theta &= \{x \in (\forall o \in \llbracket \kappa \rrbracket^O. \llbracket A \rrbracket^f(\theta, o)) \mid \forall o, o', r \in \llbracket \kappa \rrbracket^R(o, o'). (x o, x o') \in \llbracket A \rrbracket^r(\theta, o)(\theta, o')(\llbracket \Theta \rrbracket^{\text{id}} \theta, r)\} \\
\llbracket \Theta \vdash \forall \alpha : \kappa. A : * \rrbracket^r \theta \theta' \rho &= \{(x, x') \mid \forall o, o', r \in \llbracket \kappa \rrbracket^R(o, o'). (x o, x' o') \in \llbracket A \rrbracket^r(\theta, o)(\theta', o')(\rho, r)\}
\end{aligned}$$

Figure 5. Interpretation of basic types as reflexive graph morphisms

isomorphisms, so morphisms of groupoids are just functors. Every group is a groupoid with one object, and a morphism from this object to itself for every element of the group: composition is the group operation, inverses are given by the group inverses, and the identity morphism is given by the group unit. Homomorphisms between groups are in one-to-one correspondence with functors between the corresponding groupoids. An example of a groupoid that is not a group is given by the collection of cartesian spaces (i.e., \mathbb{R}^n for some n), with all diffeomorphisms (smooth functions with smooth inverses) as the morphisms.

Trivially, every groupoid is a reflexive graph, simply by forgetting the composition and inverses. Likewise, every functor between groupoids is a morphism of reflexive graphs, by forgetting the preservation of composition. Therefore, given any groupoid, we can add a new kind to System $F\omega$ that is interpreted by that groupoid, and given any functor $f : \llbracket G_1 \rrbracket \rightarrow \llbracket G_2 \rrbracket$ between groupoids interpreting kinds G_1 and G_2 , we obtain a type constant of kind $G_1 \rightarrow G_2$, with interpretation f .

Integers Our first groupoid kind is Z , interpreted by additive group of integers. We will use this kind to state the types of the trigonometric functions \sin and \cos in Section 4.1, which are periodic with period 2π . The kind Z is small:

$$\kappa ::= \dots \mid Z \quad \overline{Z \text{ small}}$$

The kind Z has the following interpretation as a reflexive graph:

$$\llbracket Z \rrbracket = (\{\ast\}, \lambda(\ast, \ast). Z, \lambda\ast. 0)$$

Note the difference between this interpretation and the interpretation of the kind Nat in the previous section. In the interpretation of Nat , the objects were complex and the relations were trivial. The reflexive graph interpreting Z has a trivial collection of objects, but a rich structure at the relation level, given by the group of integers.

This structure means that we can add the following type-level constants to our system:

$$\overline{\Theta \vdash 0 : Z} \quad \frac{\Theta \vdash A : Z \quad \Theta \vdash B : Z}{\Theta \vdash A + B : Z} \quad \frac{\Theta \vdash A : Z}{\Theta \vdash -A : Z}$$

Each of these constants has a trivial object-level interpretation, due to the trivial collection of objects in the reflexive graph $\llbracket Z \rrbracket$. At the relation-level, the interpretations are simply given using the group structure of the integers. These interpretations satisfy the abelian group axioms, so we add these axioms to the type-level equations.

Translation and Linear Transformation Groups The group of integers is not suitable for stating the invariance properties we require to apply Noether's theorem. To do so, we require groups of translations and invertible linear transformations. We extend our type system with small kinds representing n -dimensional ($n \geq 0$) translations, invertible linear transformations, and orthogonal transformations:

$$\kappa ::= \dots \mid T(n) \mid GL(n) \mid O(n) \quad \overline{T(n), GL(n), O(n) \text{ small}}$$

The interpretations of these kinds follow the same pattern as for the kind Z above. Each interpretation is a reflexive graph with a trivial collection of objects, and relations taken from the appropriate group:

$$\begin{aligned}
\llbracket T(n) \rrbracket &= (\{\ast\}, \lambda(\ast, \ast). \mathbb{R}^n, \lambda\ast. \vec{0}) \\
\llbracket GL(n) \rrbracket &= (\{\ast\}, \lambda(\ast, \ast). GL(n), \lambda\ast. I) \\
\llbracket O(n) \rrbracket &= (\{\ast\}, \lambda(\ast, \ast). O(n), \lambda\ast. I)
\end{aligned}$$

Here, we take $GL(n)$ to be exactly the group of invertible real $n \times n$ matrices, and $O(n)$ to be the group of $n \times n$ orthogonal matrices (recall that an orthogonal matrix is one whose transpose is equal to its inverse). We add the group operations and group axioms for each of these groupoid kinds in the same way as we did for the

kind Z above, except that we use multiplicative notation for the kinds $\text{GL}(n)$ and $\text{O}(n)$. Note that, except for $\text{GL}(0)$, $\text{GL}(1)$, $\text{O}(0)$ and $\text{O}(1)$, the groups $\text{GL}(n)$ and $\text{O}(n)$ are not abelian.

We also extend our type system with the following type-level constants. Each of these is interpreted by the correspondingly named homomorphism between the groups involved. For example, scale_n is interpreted by the homomorphism $\text{GL}(1) \rightarrow \text{GL}(n)$ that takes a non-zero real number s to the matrix with ss along the diagonal and zeros elsewhere.

$$\begin{aligned} \text{exp} & : \mathbb{T}(1) \rightarrow \text{GL}(1) \\ \text{ortho}_n & : \text{O}(n) \rightarrow \text{GL}(n) \\ \text{scale}_n & : \text{GL}(1) \rightarrow \text{GL}(n) \\ \underline{c} * - & : Z \rightarrow \mathbb{T}(1) \end{aligned}$$

In the last of these \underline{c} stands for an arbitrary real constant. Since each of these is interpreted by a group homomorphism, we are justified in adding the group homomorphism laws as axioms to our type-level equational theory, maintaining Property 2.

The Groupoid of Cartesian Spaces Finally in this section, we introduce the groupoid kind of cartesian spaces, which we will use for the configuration spaces of the classical mechanical systems we describe in our system. As we noted above, the collection of cartesian spaces is a groupoid with a non-trivial collection of objects. We extend our system with a kind of cartesian spaces:

$$\kappa ::= \dots \mid \text{CartSp}$$

with the following interpretation:

$$\llbracket \text{CartSp} \rrbracket = (\mathbb{N}, \lambda m, n. \{f : \mathbb{R}^m \rightarrow \mathbb{R}^n \mid f \text{ a diffeomorphism}\}, \lambda n. \text{id}_{\mathbb{R}^n})$$

We have chosen to represent cartesian spaces of dimension n just as the natural number n . This gives us a canonical representation of n -dimensional space for each n , avoiding tricky problems with isomorphic constructions of the same space.

Cartesian spaces are closed under products:

$$\frac{\Theta \vdash X : \text{CartSp} \quad \Theta \vdash Y : \text{CartSp}}{\Theta \vdash X \times Y : \text{CartSp}}$$

with the following interpretation, where the cartesian product of cartesian spaces adds their dimensions, and composes diffeomorphisms “in parallel”:

$$\begin{aligned} \llbracket X \times Y \rrbracket^f \theta & = \llbracket X \rrbracket^f \theta + \llbracket Y \rrbracket^f \theta \\ \llbracket X \times Y \rrbracket^r \theta \theta' \rho & = \lambda(\vec{x}, \vec{y}). (\llbracket X \rrbracket^r \theta \theta' \rho \vec{x}, \llbracket Y \rrbracket^r \theta \theta' \rho \vec{y}) \end{aligned}$$

Cartesian spaces are generated by the following construct that yields the n -dimensional cartesian space ($n \geq 0$) that varies with the given invertible linear transformation and translation:

$$\frac{\Theta \vdash G : \text{GL}(n) \quad \Theta \vdash T : \mathbb{T}(n)}{\Theta \vdash \mathbb{R}^n \langle G, T \rangle : \text{CartSp}}$$

As a notational convenience, we write $\mathbb{R}^1 \langle G, T \rangle$ as just $\mathbb{R} \langle G, T \rangle$ and $\mathbb{R}^0 \langle G, T \rangle$ as just \mathbb{R}^0 .

The type $\mathbb{R}^n \langle G, T \rangle$ has the following interpretation as a reflexive graph morphism:

$$\begin{aligned} \llbracket \mathbb{R}^n \langle G, T \rangle \rrbracket^f \theta & = n \\ \llbracket \mathbb{R}^n \langle G, T \rangle \rrbracket^r \theta \theta' \rho & = \lambda \vec{x}. (\llbracket G \rrbracket^r \theta \theta' \rho \vec{x} + \llbracket T \rrbracket^r \theta \theta' \rho) \end{aligned}$$

Note that $\llbracket \mathbb{R}^n \langle G, T \rangle \rrbracket^r \theta \theta' (\llbracket \Theta \rrbracket^{\text{id}} \theta)$ is the identity diffeomorphism, so this interpretation preserves identity edges, exactly as required in the definition of reflexive graph morphisms. Consequently, the interpretation of $\mathbb{R}^n \langle G, T \rangle$, and also the interpretation for products of cartesian spaces, preserves Property 2.

Using the type-level natural numbers and their recursion operator, we can define a type of n -ary products of a cartesian space:

$$\text{vec} = \lambda n : \text{Nat}, X : \text{CartSp}. \text{natrec } n \ \mathbb{R}^0 \ (\lambda x. X \times x)$$

n -ary products of cartesian spaces will be useful in Section 5 when we define Lagrangians describing systems that are generic in the number of particles involved.

Finally, we include an operator that takes any cartesian space to its corresponding proper type:

$$\frac{\Theta \vdash X : \text{CartSp}}{\Theta \vdash \llbracket X \rrbracket : *}$$

with the following straightforward interpretation:

$$\begin{aligned} \llbracket \llbracket X \rrbracket \rrbracket^f \theta & = \mathbb{R}^{\llbracket X \rrbracket^f \theta} \\ \llbracket \llbracket X \rrbracket \rrbracket^r \theta \theta' \rho & = \{(\vec{x}, \vec{x}') \mid (\llbracket X \rrbracket^r \theta \theta' \rho) \vec{x} = \vec{x}'\} \end{aligned}$$

4. A Type System for Classical Mechanics

In the previous section we constructed a way to derive invariance properties from types. In Section 2, we saw that Noether’s theorem is a method for deriving conservation laws for classical mechanical systems from their invariance properties. In this section, we construct a type system for constructing invariant Lagrangians. In the following section, Section 5, we present several uses of our type system for defining invariant Lagrangians that describe many different kinds of classical mechanical systems.

4.1 Extending System $F\omega$ with Smooth Functions

The Type of Smooth Functions Given a pair of cartesian spaces X and Y we provide a way to construct the type of smooth invariant smooth functions from X to Y :

$$\frac{\Theta \vdash X : \text{CartSp} \quad \Theta \vdash Y : \text{CartSp}}{\Theta \vdash C^\infty(X, Y) : *}$$

with the following interpretation. Since $C^\infty(X, Y)$ is a proper type, the relational interpretation in this case is a genuine relation. We relate functions that are invariant under the diffeomorphism interpretations of X and Y in the current relational context ρ :

$$\begin{aligned} \llbracket C^\infty(X, Y) \rrbracket^f \theta & = \{f : \mathbb{R}^{\llbracket X \rrbracket^f \theta} \rightarrow \mathbb{R}^{\llbracket Y \rrbracket^f \theta} \mid f \text{ smooth}\} \\ \llbracket C^\infty(X, Y) \rrbracket^r \theta \theta' \rho & = \{(f, f') \mid \forall \vec{x} \in \mathbb{R}^{\llbracket X \rrbracket^f \theta}. \\ & \quad (\llbracket Y \rrbracket^r \theta \theta' \rho)(f \vec{x}) = f' (\llbracket X \rrbracket^r \theta \theta' \rho \vec{x})\} \end{aligned}$$

An Example Free Theorem We are now in a position to formally show how we can use the relationally parametric model we have defined to derive invariance properties. Consider a closed term with the following type:

$$e : \forall o : \text{O}(n). C^\infty(\mathbb{R}^n \langle \text{ortho}_n(o), 0 \rangle, \mathbb{R} \langle 1, 0 \rangle)$$

We can deduce, from Property 3 and the relational interpretations of all the types involved, that the denotation of e satisfies the following free theorem:

$$\forall O \in \text{O}(n). \forall \vec{x} \in \mathbb{R}^n. \llbracket e \rrbracket(O \vec{x}) = \llbracket e \rrbracket(\vec{x})$$

This free theorem is exactly the kind of invariance property we require to apply Noether’s theorem. Compare the form of the statement here, with the mildly more general statement of Equation 9, back in Section 2. Moreover, $\llbracket e \rrbracket$ is a smooth function from \mathbb{R}^n to \mathbb{R} , just as we require for Lagrangian mechanics.

Combinators for Smooth Functions Of course, such free theorems are pointless without ways to build elements of the smooth function spaces. We now extend our type system, for the last time, with a set of term-level constants for constructing smooth functions, invariant under linear transformations and translations.

Our first set of combinators constructs constant and identity smooth functions, and composes smooth functions between cartesian spaces. We also have combinators for pairing and projection for the product of cartesian spaces.

$$\begin{aligned}
\text{const} &: \{Y\} \rightarrow C^\infty(X, Y) \\
\text{id} &: C^\infty(X, X) \\
(\gg) &: C^\infty(X, Y) \rightarrow C^\infty(Y, Z) \rightarrow C^\infty(X, Z) \\
\text{pair} &: C^\infty(X, Y) \rightarrow C^\infty(X, Z) \rightarrow C^\infty(X, Y \times Z) \\
\text{proj}_1 &: C^\infty(X \times Y, X) \\
\text{proj}_2 &: C^\infty(X \times Y, Y)
\end{aligned}$$

Next, we include combinators for constant, invariant, vectors, the zero vector, vector addition, subtraction and scaling. We also include the Euclidean dot product of vectors. The type dot product operation is where the orthogonal group kind $\mathbf{O}(n)$ is used: the dot product of vectors is invariant under orthogonal transformations.

$$\begin{aligned}
\vec{c} &: \{\mathbb{R}^n \langle 1, 0 \rangle\} \\
0 &: \forall g: \mathbf{GL}(n). \{\mathbb{R}^n \langle g, 0 \rangle\} \\
(+) &: \forall g: \mathbf{GL}(n), t_1, t_2: \mathbf{T}(n). \\
&\quad C^\infty(\mathbb{R}^n \langle g, t_1 \rangle \times \mathbb{R}^n \langle g, t_2 \rangle, \mathbb{R}^n \langle g, t_1 + t_2 \rangle) \\
(-) &: \forall g: \mathbf{GL}(n), t_1, t_2: \mathbf{T}(n). \\
&\quad C^\infty(\mathbb{R}^n \langle g, t_1 \rangle \times \mathbb{R}^n \langle g, t_2 \rangle, \mathbb{R}^n \langle g, t_1 - t_2 \rangle) \\
(*) &: \forall g_1: \mathbf{GL}(1), g_2: \mathbf{GL}(n). \\
&\quad C^\infty(\mathbb{R} \langle g_1, 0 \rangle \times \mathbb{R}^n \langle g_2, 0 \rangle, \mathbb{R}^n \langle \text{scale}_n(g_1)g_2, 0 \rangle) \\
(\cdot) &: \forall g: \mathbf{GL}(1), o: \mathbf{O}(n). \\
&\quad C^\infty(\mathbb{R}^n \langle (\text{scale}_n g)(\text{ortho}_n o), 0 \rangle \times \\
&\quad \mathbb{R}^n \langle (\text{scale}_n g)(\text{ortho}_n o), 0 \rangle, \mathbb{R} \langle (\text{scale}_n g)^2, 0 \rangle)
\end{aligned}$$

Each of these combinators has a straightforward interpretation as a smooth function, and the proof that the fundamental theorem of logical relations is maintained for our system (i.e., that Property 3 is maintained) follows directly from elementary facts about linear algebra.

It will be useful for our examples to have the following trigonometric functions, exponential function and division operation. The types of each of these functions describes some of their invariance properties, such as the periodicity of the sin and cos functions. A technical problem with our current system is that we do not account for the non-definedness of division at 0, nor do we account for the square root of negative numbers. However, division and square root will be useful in our examples in Section 5, so we take a pragmatic approach, in common with most physics textbooks, and informally treat them as total functions.

$$\begin{aligned}
\text{sin} &: \forall z: \mathbf{Z}. C^\infty(\mathbb{R} \langle 1, 2\pi * z \rangle, \mathbb{R} \langle 1, 0 \rangle) \\
\text{cos} &: \forall z: \mathbf{Z}. C^\infty(\mathbb{R} \langle 1, 2\pi * z \rangle, \mathbb{R} \langle 1, 0 \rangle) \\
\text{exp} &: \forall t: \mathbf{T}(1). C^\infty(\mathbb{R} \langle 1, t \rangle, \mathbb{R} \langle \text{exp } t, 0 \rangle) \\
(/) &: \forall g_1, g_2: \mathbf{GL}(1). C^\infty(\mathbb{R} \langle g_1, 0 \rangle \times \mathbb{R} \langle g_2, 0 \rangle, \mathbb{R} \langle g_1 g_2^{-1}, 0 \rangle) \\
\text{sqrt} &: \forall g: \mathbf{GL}(1). C^\infty(\mathbb{R} \langle g \cdot g, 0 \rangle, \mathbb{R} \langle g, 0 \rangle)
\end{aligned}$$

Finally, we include three primitive combinators for dealing with n -ary products of vectors. These combinators will be useful when defining Lagrangians that are generic in the number of particles.

$$\begin{aligned}
\text{sum} &: \forall n: \mathbf{Nat}, g: \mathbf{GL}(1). C^\infty(\text{vec } n \ (\mathbb{R} \langle g, 0 \rangle), \mathbb{R} \langle g, 0 \rangle) \\
\text{map} &: \forall n: \mathbf{Nat}. C^\infty(Z \times X, Y) \rightarrow C^\infty(Z \times \text{vec } n \ X, \text{vec } n \ Y) \\
\text{cross} &: \forall m, n: \mathbf{Nat}. \\
&\quad C^\infty(\text{vec } m \ X \times \text{vec } n \ Y, \text{vec } (m * n) \ (X \times Y))
\end{aligned}$$

The combinator sum sums the list of real numbers it is given, while cross takes the cartesian product of two vectors of elements of cartesian spaces, and map maps the given smooth function over a vector. The multiplication of natural numbers $m * n$ is straightforwardly defined in terms of the *natrec* recursion operator. We will use cross in Section 5 to define Lagrangians in terms of the interactions between systems of particles.

$$\begin{aligned}
[z]_\Delta &= \text{const } z \\
[x]_\Delta &= \text{project}_\Delta(x) \\
[\text{let } \vec{x} = e_1 \text{ in } e_2]_\Delta &= \text{pair id } [e_1]_\Delta \gg \gg [e_2]_{\Delta, \vec{x}} \\
[(e_1, \dots, e_n)]_\Delta &= \text{pair}_n([e_1]_\Delta, \dots, [e_n]_\Delta) \\
[e_1(e_2)]_\Delta &= [e_2]_\Delta \gg \gg e_1 \\
[\vec{c}]_\Delta &= \text{const } \vec{c} \\
[0]_\Delta &= \text{const } 0 \\
[e_1 + e_2]_\Delta &= \text{pair } [e_1]_\Delta [e_2]_\Delta \gg \gg (+) \\
[e_1 - e_2]_\Delta &= \text{pair } [e_1]_\Delta [e_2]_\Delta \gg \gg (-) \\
[e_1 e_2]_\Delta &= \text{pair } [e_1]_\Delta [e_2]_\Delta \gg \gg (*) \\
[e_1 \cdot e_2]_\Delta &= \text{pair } [e_1]_\Delta [e_2]_\Delta \gg \gg (\cdot) \\
[\sin e]_\Delta &= [e]_\Delta \gg \gg \text{sin} \\
[\cos e]_\Delta &= [e]_\Delta \gg \gg \text{cos} \\
[\exp e]_\Delta &= [e]_\Delta \gg \gg \text{exp} \\
[e_1/e_2]_\Delta &= \text{pair } [e_1]_\Delta [e_2]_\Delta \gg \gg (/) \\
[\text{sqrt } e]_\Delta &= [e]_\Delta \gg \gg \text{sqrt} \\
[\text{sum } e]_\Delta &= [e]_\Delta \gg \gg \text{sum} \\
[\text{map } (x. e_1) e_2]_\Delta &= \text{pair id } [e_2]_\Delta \gg \gg (\text{map } [e_1]_{\Delta, x}) \\
[\text{cross } e_1 e_2]_\Delta &= \text{pair } [e_1]_\Delta [e_2]_\Delta \gg \gg \text{cross}
\end{aligned}$$

Figure 7. Desugaring of the surface syntax

4.2 A Surface Syntax for Smooth Functions

The combinators we have presented allow the construction of smooth invariant functions, but doing so is painful due to the point-free style that they force. In order to be able to actually define readable Lagrangians in our calculus, we define a surface syntax for writing smooth functions that is desugared into our extended version of System $F\omega$.

We define a typing judgement of smooth terms $\Theta \mid \Gamma; \Delta \vdash e : X$ by the rules in Figure 6, where Θ and Γ are the kinding and typing contexts of our extension of $F\omega$, and $\Delta = x_1 : X_1, \dots, x_n : X_n$ is a context of cartesian spaces. That is, for each $x : X$ in Δ , we have $\Theta \vdash X : \text{CartSp}$.

The desugaring of smooth function terms into our extension of System $F\omega$ is via the operation $[-]_\Delta$, defined in Figure 7. In this definition, we make use of two derived combinators for projection and pairing:

$$\begin{aligned}
\underline{\text{project}}_{x: X, \Delta}(x) &= \text{proj}_1 \\
\underline{\text{project}}_{y: Y, \Delta}(x) &= \text{proj}_2 \gg \gg \text{project}_\Delta(x) \quad (x \neq y)
\end{aligned}$$

and

$$\begin{aligned}
\underline{\text{pair}}_1(e) &= e \\
\underline{\text{pair}}_{n+1}(e, \vec{e}) &= \text{pair } e \ (\underline{\text{pair}}_n(\vec{e}))
\end{aligned}$$

The following theorem follows easily by induction on the typing derivations of smooth terms, where $[\Delta] = [x_1 : X_1, \dots, x_n : X_n] = X_1 \times \dots \times X_n$:

Theorem 2. *If $\Theta \mid \Gamma; \Delta \vdash e : X$ then $\Theta \mid \Gamma \vdash [e]_\Delta : C^\infty([\Delta], X)$.*

5. Examples of Conservation Laws from Types

We now present a number of examples of classical mechanical systems expressible in the type system we constructed in the previous section, and derive the free theorems and consequent conserved properties for each one. These examples are all standard examples demonstrating the application of Lagrangian mechanics (see, for example, Landau and Lifschitz [12]). Our contribution here is to express them in a type system that makes clear their invariance properties.

Administrative rules		
$\frac{z : \{X\} \in \Gamma}{\Theta \mid \Gamma; \Delta \vdash z : X}$	$\frac{x : X \in \Delta}{\Theta \mid \Gamma; \Delta \vdash x : X}$	$\frac{\Theta \mid \Gamma; \Delta \vdash e_1 : X_1 \times \dots \times X_n \quad \Theta \mid \Gamma; \Delta, x_1 : X_1, \dots, x_n : X_n \vdash e_2 : Y}{\Theta \mid \Gamma; \Delta \vdash \text{let } x_1, \dots, x_n = e_1 \text{ in } e_2 : Y}$
$\frac{\Theta \mid \Gamma \vdash e_1 : C^\infty(X, Y) \quad \Theta \mid \Gamma; \Delta \vdash e_2 : X}{\Theta \mid \Gamma; \Delta \vdash e_1(e_2) : Y}$	$\frac{\Theta \mid \Gamma; \Delta \vdash e_1 : X \quad \dots \quad \Theta \mid \Gamma; \Delta \vdash e_n : X_n}{\Theta \mid \Gamma; \Delta \vdash (e_1, \dots, e_n) : X_1 \times \dots \times X_n}$	
Vector space operations, and dot product		
$\frac{\vec{c} \in \mathbb{R}^n}{\Theta \mid \Gamma; \Delta \vdash \vec{c} : \mathbb{R}^n \langle 1, 0 \rangle}$	$\frac{\Theta \vdash G : \text{GL}(n)}{\Theta \mid \Gamma; \Delta \vdash 0 : \mathbb{R}^n \langle G, 0 \rangle}$	$\frac{\Theta \mid \Gamma; \Delta \vdash e_1 : \mathbb{R}^n \langle G, T_1 \rangle \quad \Theta \mid \Gamma; \Delta \vdash e_2 : \mathbb{R}^n \langle G, T_2 \rangle}{\Theta \mid \Gamma; \Delta \vdash e_1 + e_2 : \mathbb{R}^n \langle G, T_1 + T_2 \rangle}$
$\frac{\Theta \mid \Gamma; \Delta \vdash e_1 : \mathbb{R}^n \langle G, T_1 \rangle \quad \Theta \mid \Gamma; \Delta \vdash e_2 : \mathbb{R}^n \langle G, T_2 \rangle}{\Theta \mid \Gamma; \Delta \vdash e_1 - e_2 : \mathbb{R}^n \langle G, T_1 - T_2 \rangle}$	$\frac{\Theta \mid \Gamma; \Delta \vdash e_1 : \mathbb{R} \langle G_1, 0 \rangle \quad \Theta \mid \Gamma; \Delta \vdash e_2 : \mathbb{R}^n \langle G_2, 0 \rangle}{\Theta \mid \Gamma; \Delta \vdash e_1 e_2 : \mathbb{R}^n \langle \text{scale}_n(G_1) G_2, 0 \rangle}$	
$\frac{\Theta \mid \Gamma; \Delta \vdash e_1 : \mathbb{R}^n \langle \text{scale}_n(G) \text{ortho}_n(O), 0 \rangle \quad \Theta \mid \Gamma; \Delta \vdash e_2 : \mathbb{R}^n \langle \text{scale}_n(G) \text{ortho}_n(O), 0 \rangle}{\Theta \mid \Gamma; \Delta \vdash e_1 \cdot e_2 : \mathbb{R} \langle \text{scale}_n(G)^2, 0 \rangle}$		
Transcendental functions		
$\frac{\Theta \mid \Gamma; \Delta \vdash e : \mathbb{R} \langle 1, T \rangle}{\Theta \mid \Gamma; \Delta \vdash \exp e : \mathbb{R} \langle \exp T, 0 \rangle}$	$\frac{\Theta \mid \Gamma; \Delta \vdash e : \mathbb{R} \langle 1, 2\pi * Z \rangle}{\Theta \mid \Gamma; \Delta \vdash \sin e : \mathbb{R} \langle 1, 0 \rangle}$	$\frac{\Theta \mid \Gamma; \Delta \vdash e : \mathbb{R} \langle 1, 2\pi * Z \rangle}{\Theta \mid \Gamma; \Delta \vdash \cos e : \mathbb{R} \langle 1, 0 \rangle}$
Division, square root, sum, and cross		
$\frac{\Theta \mid \Gamma; \Delta \vdash e_1 : \mathbb{R} \langle G_1, 0 \rangle \quad \Theta \mid \Gamma; \Delta \vdash e_2 : \mathbb{R} \langle G_2, 0 \rangle}{\Theta \mid \Gamma; \Delta \vdash \frac{e_1}{e_2} : \mathbb{R} \langle G_1 G_2^{-1}, 0 \rangle}$	$\frac{\Theta \mid \Gamma; \Delta \vdash e : \mathbb{R} \langle G^2, 0 \rangle}{\Theta \mid \Gamma; \Delta \vdash \text{sqrt } e : \mathbb{R} \langle G, 0 \rangle}$	$\frac{\Theta \mid \Gamma; \Delta \vdash e : \text{vec } N \langle \mathbb{R} \langle G, 0 \rangle \rangle}{\Theta \mid \Gamma; \Delta \vdash \text{sum } e : \mathbb{R} \langle G, 0 \rangle}$
$\frac{\Theta \mid \Gamma; \Delta, x : X \vdash e_1 : Y \quad \Theta \mid \Gamma; \Delta \vdash e_2 : \text{vec } N X}{\Theta \mid \Gamma; \Delta \vdash \text{map } (x. e_1) e_2 : \text{vec } N Y}$	$\frac{\Theta \mid \Gamma; \Delta \vdash e_1 : \text{vec } M X \quad \Theta \mid \Gamma; \Delta \vdash e_2 : \text{vec } N Y}{\Theta \mid \Gamma; \Delta \vdash \text{cross } e_1 e_2 : \text{vec } (M * N) (X \times Y)}$	

Figure 6. A surface syntax for smooth functions

5.1 Single Particles

Free Particle Our first example is of a simple Lagrangian for a single free particle in 3-dimensional space, with no external forces. We define the Lagrangian using the syntax for smooth terms we defined in the previous section, in the following kinding, typing and cartesian space contexts:

$$\begin{aligned} \Theta &= t_t : \mathbb{T}(1), t_x : \mathbb{T}(3), o : \mathbb{O}(3) \\ \Gamma &= m : \{ \mathbb{R} \langle 1, 0 \rangle \} \\ \Delta &= t : \mathbb{R} \langle 1, t_t \rangle, x : \mathbb{R}^3 \langle \text{ortho}_3(o), t_x \rangle, \dot{x} : \mathbb{R}^3 \langle \text{ortho}_3(o), 0 \rangle \end{aligned}$$

where o represents an arbitrary orthogonal transformation of the space, t_t represents a translation in time, t_x represents a 3-dimensional translation in space, m is the constant mass of the free particle, t is the current time, x is the current position of the particle, and \dot{x} is the current velocity.

We write the Lagrangian like so, which in the absence of external forces just consists of the kinetic energy term:

$$L = \frac{1}{2} m (\dot{x} \cdot \dot{x}) : \mathbb{R} \langle 1, 0 \rangle$$

By the fundamental theorem of logical relations for our calculus (Property 3), we can derive the following free theorems, one each for the three group parameters t_t , t_x and o . Quantifying over translations in time, t_t , gives us invariance under translation in time:

$$\forall t_t \in \mathbb{R}. \llbracket L \rrbracket(t + t_t, \vec{x}, \dot{\vec{x}}) = \llbracket L \rrbracket(t, \vec{x}, \dot{\vec{x}})$$

As we saw in Example 4, Noether's theorem tells us that invariance under translation in time yields conservation of energy. Likewise,

quantifying over translations in space, t_x , gives us invariance under translation in space:

$$\forall \vec{t}_x \in \mathbb{R}^3. \llbracket L \rrbracket(t, \vec{x} + \vec{t}_x, \dot{\vec{x}}) = \llbracket L \rrbracket(t, \vec{x}, \dot{\vec{x}})$$

Similar to the example of the spring-coupled particles in the introduction, invariance under translations in space yields conservation of linear momentum.

Quantification over orthogonal transformations $o : \mathbb{O}(3)$ yields the following free theorem, which states that the Lagrangian is invariant under modification of the positions and velocities by arbitrary orthogonal transformations O :

$$\forall O \in \mathbb{O}(3). \llbracket L \rrbracket(t, O\vec{x}, O\dot{\vec{x}}) = \llbracket L \rrbracket(t, \vec{x}, \dot{\vec{x}})$$

We are particularly interested in the orthogonal transformations derived from rotations, since these will give us the continuous transformations we need to apply Noether's theorem. For example, rotation by an angle ϵ around the x_3 -axis is given by the following ϵ -indexed family of orthogonal matrices:

$$O_\epsilon = \begin{pmatrix} \cos \epsilon & \sin \epsilon & 0 \\ -\sin \epsilon & \cos \epsilon & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

This leads to the following ϵ -indexed family of transformations, suitable for Noether's theorem:

$$\Psi_\epsilon \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = O_\epsilon \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 \cos \epsilon + x_2 \sin \epsilon \\ -x_1 \sin \epsilon + x_2 \cos \epsilon \\ x_3 \end{pmatrix}$$

Plugging this into Equation 10, and using the definition of L above, we obtain the following conservation law:

$$\frac{d}{dt}(m\dot{x}x_2 - m\dot{x}x_1) = 0$$

This is exactly conservation of angular momentum around the x_3 -axis [12]. By considering the families of orthogonal matrices for rotation around other axes, we can also derive conservation laws for angular momentum around arbitrary axes.

Particle in a Potential Field If we now extend the typing context of our example with an arbitrary potential energy function V , depending on the position of the particle, we can define the Lagrangian for a single particle acted upon by a potential field that depends on the position of the particle. The invariance properties of the whole system will depend upon the invariance properties of the potential field. If we assume the following contexts:

$$\begin{aligned} \Theta &= t_t : \mathbb{T}(1), o : \mathbb{O}(3) \\ \Gamma &= m : \mathbb{R}\langle 1, 0 \rangle \int, \\ &V : \forall o : \mathbb{O}(3). C^\infty(\mathbb{R}^3\langle \text{ortho}_3(o), 0 \rangle, \mathbb{R}\langle 1, 0 \rangle) \\ \Delta &= t : \mathbb{R}\langle 1, t_t \rangle, x : \mathbb{R}^3\langle \text{ortho}_3(o), 0 \rangle, \dot{x} : \mathbb{R}^3\langle \text{ortho}_3(o), 0 \rangle \end{aligned}$$

where everything is as in the previous example, except for the addition of the polymorphic potential energy function V . We incorporate this into our Lagrangian like so:

$$L = \frac{1}{2}m(\dot{x} \cdot \dot{x}) - V(x) : \mathbb{R}\langle 1, 0 \rangle$$

Even though V is left abstract, the types of t , x , \dot{x} and L tell us that this Lagrangian is invariant under translations in time and also under all orthogonal transformations. Hence, by Noether's theorem, the system this Lagrangian describes has energy and angular momentum as conserved quantities.

5.2 The n -Body Problem

Using the constructs of our calculus for dealing with n -ary vectors of positions and velocities, we can compactly write Lagrangians that describe systems of n particles interacting through Newtonian gravitational attraction. We use the following kinding, typing and cartesian space contexts, where the type-level parameter n represents the number of particles that we are considering. For simplicity, we assume that all the bodies involved in the system have equal mass m .

$$\begin{aligned} \Theta &= n : \mathbb{N}\text{at}, t_t : \mathbb{T}(1), t_x : \mathbb{T}(3), o : \mathbb{O}(3) \\ \Gamma &= m : \mathbb{R}\langle 1, 0 \rangle \int \\ \Delta &= t : \mathbb{R}\langle 1, t_t \rangle, \\ &x : \text{vec } n(\mathbb{R}^3\langle \text{ortho}_3(o), t_x \rangle), \\ &\dot{x} : \text{vec } n(\mathbb{R}^3\langle \text{ortho}_3(o), 0 \rangle) \end{aligned}$$

The Lagrangian for this system is defined as follows in our surface syntax for smooth terms. The kinetic energy component is the sum of the kinetic energies of all the particles in the system. The potential energy components sums up all the gravitational potential energies due to the interactions between each pair of bodies, where G is the gravitational constant.

$$L = \frac{1}{2}m(\text{sum}(\text{map}(\dot{x}_i. \dot{x}_i \cdot \dot{x}_i)) \dot{x}) - \text{sum}(\text{map}((x_i, x_j). Gm^2/|x_i - x_j|)(\text{cross } x x)) : \mathbb{R}\langle 1, 0 \rangle$$

We have used the notation $|e|$ as shorthand for $\text{sqrt}(e \cdot e)$, i.e., the norm of the vector e .

Again, just by looking at the types of t , x , \dot{x} , and L , we can determine via free theorems that this Lagrangian is invariant under translation in time and space, and under all orthogonal transformations. Hence, by Noether's theorem, this system has energy and linear and angular momentum as conserved quantities.

5.3 Pendulum

All our examples above have used rectangular coordinate systems. Part of the power of the Lagrangian formulation of classical mechanics is the ability to use appropriate generalised coordinates to describe systems in simple terms. A standard example is that of a pendulum, where we take the angle θ of swing from the vertical as the coordinate.

We use the following kinding, typing and cartesian space contexts:

$$\begin{aligned} \Theta &= t_t : \mathbb{T}(1), z : \mathbb{Z} \\ \Gamma &= m : \mathbb{R}\langle 1, 0 \rangle \int, l : \mathbb{R}\langle 1, 0 \rangle \int \\ \Delta &= t : \mathbb{R}\langle 1, t_t \rangle, \theta : \mathbb{R}\langle 1, 2\pi * z \rangle, \dot{\theta} : \mathbb{R}\langle 1, 0 \rangle \end{aligned}$$

where m is the mass at the end of the pendulum (we assume the rod of the pendulum to be massless), l is the length of the pendulum, θ is the angle of swing and $\dot{\theta}$ is the current rate of change of the angle of swing.

Defining the Lagrangian for this system is a matter of simple trigonometry and differential calculus:

$$\begin{aligned} L &= \text{let } y = l \sin \theta \text{ in} \\ &\text{let } \dot{x} = l\dot{\theta} \cos \theta \text{ in} \\ &\text{let } \dot{y} = -l\dot{\theta} \sin \theta \text{ in} \\ &\frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - mgy : \mathbb{R}\langle 1, 0 \rangle \end{aligned}$$

From the type of t , we can easily determine that this Lagrangian is invariant under translation in time, and hence the total energy is a conserved quantity of this system. We can also derive another free theorem due to the quantification over the group of integers via the variable z . However, this does not yield a continuous invariance of the Lagrangian, and hence no conserved quantity via Noether's theorem.

5.4 Oscillators

Coupled Particles Our first example of a Lagrangian in the introduction was of a pair of particles of equal mass coupled by a spring, for which we derived conservation of linear momentum and energy. In terms of our type system, we have the following kinding, typing and cartesian space contexts, where m is the individual mass of the two particles, and k is the spring constant:

$$\begin{aligned} \Theta &= t_t : \mathbb{T}(1), t_x : \mathbb{T}(1) \\ \Gamma &= m : \mathbb{R}\langle 1, 0 \rangle \int, k : \mathbb{R}\langle 1, 0 \rangle \int \\ \Delta &= t : \mathbb{R}\langle 1, t_t \rangle, x_1 : \mathbb{R}\langle 1, t_x \rangle, x_2 : \mathbb{R}\langle 1, t_x \rangle, \\ &x_1 : \mathbb{R}\langle 1, 0 \rangle, x_2 : \mathbb{R}\langle 1, 0 \rangle \end{aligned}$$

In our surface syntax for smooth terms, the Lagrangian is written just as it was in the introduction:

$$L = \frac{1}{2}m(\dot{x}_1^2 + \dot{x}_2^2) - \frac{1}{2}k^2(x_1 - x_2)^2 : \mathbb{R}\langle 1, 0 \rangle$$

The free theorems for this term state the invariance in time and space translation:

$$\begin{aligned} \forall t_t : \mathbb{R}. \llbracket L \rrbracket(t + t_t, x_1, x_2, \dot{x}_1, \dot{x}_2) &= \llbracket L \rrbracket(t, x_1, x_2, \dot{x}_1, \dot{x}_2) \\ \forall t_x : \mathbb{R}. \llbracket L \rrbracket(t, x_1 + t_x, x_2 + t_x, \dot{x}_1, \dot{x}_2) &= \llbracket L \rrbracket(t, x_1, x_2, \dot{x}_1, \dot{x}_2) \end{aligned}$$

As we saw in Section 2, these invariance properties allow us to deduce conservation of energy and linear momentum for this system.

Damped Oscillator All of the example Lagrangians we have looked at so far have had no dependence on time, and so they have all described systems with total energy as a conserved quantity. A system with a single particle attached to a damped spring (whose other endpoint is fixed at position 0) provides an example of system that does not have energy as a conserved quantity. This example is taken from Neuenschwander's book [13]. We use the

following kind, typing and cartesian space contexts:

$$\begin{aligned}\Theta &= t_t : \mathbb{T}(1) \\ \Gamma &= k : \mathbb{R}\langle 1, 0 \rangle \\ \Delta &= t : \mathbb{R}\langle 1, t_t + t_t \rangle, x : \mathbb{R}\langle \exp(-t_t), 0 \rangle, \dot{x} : \mathbb{R}\langle \exp(-t_t), 0 \rangle\end{aligned}$$

Here, the position and velocity of the single particle *scale* with the exponential of the translation in time. This will account for the exponential damping we apply to the system. For simplicity, we assume that the mass of the particle and the spring constant are both 1. The Lagrangian for this system looks similar to the coupled pair above, except for an additional damping factor:

$$L = \left(\frac{1}{2} \dot{x}^2 - \frac{1}{2} x^2 \right) \exp(t) : \mathbb{R}\langle 1, 0 \rangle$$

From the types of t , x and \dot{x} , we learn that this system is invariant under the families of transformations $\Phi_\epsilon(t) = t + 2\epsilon$ and $\Psi_\epsilon(x) = e^\epsilon x$. This leads to the following conservation law for this system, linking energy and linear momentum:

$$\frac{d}{dt} \left[\left(\frac{1}{2} x \dot{x} + \frac{1}{2} \dot{x}^2 + \frac{1}{2} x^2 \right) e^t \right] = 0$$

6. Conclusions

We have presented a type system for writing invariant Langrangians, with a relationally parametric semantics that allows the derivation of free theorems that can be used with Noether's theorem to discover conservation laws for classical mechanical systems. Our key technical contribution has been the observation that relationally parametric models of System $F\omega$ admit kinds that are interpreted as groupoids, allowing geometric invariance properties to be integrated directly into the model.

Related Work We are not aware of any other work linking type systems or relational parametricity with invariance properties for classical mechanics, or with Noether's theorem. We have already mentioned the work of Atkey, Johann and Kennedy [5] that uses relational parametricity with a specialised type system to derive geometric invariance properties, albeit in a setting without smooth functions, and without an application to Noether's theorem. In this paper, we have placed Atkey *et al.*'s work in a more general setting by framing it as an extension of System $F\omega$, which allows us to (a) incorporate type-level computation, as we used to define the n -ary vectors of coordinates; and (b) to reuse previous work on relationally parametric models of System $F\omega$ by Atkey [3].

Sussman and Wisdom [20] reformulate Lagrangian and Hamiltonian mechanics in a more programming language style so that it can be implemented within the programming language Scheme. In doing so, they clear up some of the ambiguous syntax common in the standard presentations of classical mechanics. However, they do not attempt to incorporate invariance properties into types, as we have done here.

There has been prior work on programming languages inspired by theoretical physics. Quantum programming languages, which hope to exploit the properties of quantum computation in an understandable way, have been the subject of study for several years now. We cite Abramsky [1] as an introduction to this field.

Future Work We have barely scratched the surface of the applicability of Noether's theorem to theoretical physics. Conservation laws for classical and quantum field theories are derivable from the much more general theorem that Noether originally proved [14]. Neuenschwander's book [13] describes in an easy to read way how Noether's theorem applies to field theories, allowing for relativistic and quantum systems to be studied, and how it naturally leads to the notion of Gauge invariance, an important principle in modern theories of particle physics.

From the type system point of view, it seems clear that greater precision in the types will be required in order to accommodate more detailed physical theories. Study of relational parametricity for dependent types, as done for example by Bernardy *et al.* [6] and Atkey *et al.* [4] may be useful here. Also, recent work on homotopy type theory and, in particular ∞ -groupoid models of type theory is almost certainly relevant [21].

For each of the examples in Section 5, we computed the necessary typing derivations by hand. Type inference for the system was have presented here, or some variant of it, is a key item of future work. Ideally, the user would enter the description of a Lagrangian, and the system would be able to tell them the free theorems, and consequent conservation laws that hold. Kennedy [10] and Gundry [8] have studied type inference for the related, but simpler, setting of dimension types.

References

- [1] S. Abramsky. High-Level Methods for Quantum Computation and Information. Proceedings, LICS, 2004.
- [2] V. I. Arnol'd. *Mathematical Methods of Classical Mechanics*. Springer, 1989.
- [3] R. Atkey. Relational Parametricity for Higher Kinds. Proceedings, CSL, 2012.
- [4] R. Atkey, N. Ghani, and P. Johann. A Relationally Parametric Model of Dependent Type Theory. Proceedings, POPL, 2014.
- [5] R. Atkey, P. Johann, and A. J. Kennedy. Abstraction and Invariance for Algebraically Indexed Types. Proceedings, POPL, 2013.
- [6] J.-P. Bernardy, P. Jansson, R. Paterson. Proofs for Free: Parametricity for Dependent Types. *Journal of Functional Programming* 22(2), pp. 107-152, 2012.
- [7] I. M. Gelfand and S. V. Fomin, R. A. Silverman (ed.). *Calculus of Variations*. Dover Publications, 2000.
- [8] A. Gundry. Type Inference for Units of Measure. Technical Report, University of Strathclyde, 2011.
- [9] R. Hasegawa. Relational Limits in General Polymorphism. *Publications of the Research Institute for Mathematical Sciences* 30, pp. 535-576, 1994.
- [10] A. J. Kennedy. Dimension Types. Proceedings, ESOP, 1994.
- [11] A. J. Kennedy. Relational Parametricity and Units of Measure. Proceedings, POPL, pp. 442-455, 1997.
- [12] L. D. Landau and E. M. Lifschitz. *Mechanics*. Pergamon Press. 1967.
- [13] D. E. Neuenschwander. *Emmy Noether's Wonderful Theorem*. The John Hopkins University Press, 2011.
- [14] E. Noether, M. Tavel (translator). Invariant Variation Problems. *Transport Theory and Statistical Physics* 1(3), pp. 186-207, 1971. Original in *Gott. Nachr.*, 1918:235-257, 1918.
- [15] S. Mac Lane. *Categories for the Working Mathematician*, 2nd edition. Springer, 1998.
- [16] B. Pierce. *Types and Programming Languages*. MIT Press, 2002.
- [17] A. M. Pitts. Polymorphism is Set Theoretic, Constructively. *Proc., Category Theory and Computer Science*, pp. 12-39, 1987.
- [18] J. C. Reynolds. Types, Abstraction and Parametric Polymorphism. *Information Processing* 83, pp. 513-523, 1983.
- [19] E. Robinson and G. Rosolini. Reflexive Graphs and Parametric Polymorphism. *Proc., Logic in Computer Science*, pp. 364-371, 1994.
- [20] G. J. Sussman and J. Wisdom. *Structure and Interpretation of Classical Mechanics*. MIT Press, 2001
- [21] The Univalent Foundations Program. *Homotopy Type Theory*. Institute for Advanced Study, 2013.
- [22] P. Wadler. Theorems for Free!. Proceedings, FPCA, pp. 347-359, 1989.